

www.mientayvn.com

Khi đọc qua tài liệu này, nếu phát hiện sai sót hoặc nội dung kém chất lượng xin hãy thông báo để chúng tôi sửa chữa hoặc thay thế bằng một tài liệu cùng chủ đề của tác giả khác. Tài liệu này bao gồm nhiều tài liệu nhỏ có cùng chủ đề bên trong nó. Phần nội dung bạn cần có thể nằm ở giữa hoặc ở cuối tài liệu này, hãy sử dụng chức năng Search để tìm chúng.

Bạn có thể tham khảo nguồn tài liệu được dịch từ tiếng Anh tại đây:

http://mientayvn.com/Tai_lieu_da_dich.html

Thông tin liên hệ:

Yahoo mail: thanhlam1910_2006@yahoo.com

Gmail: frbwrthes@gmail.com

Theo yêu cầu của khách hàng, trong một năm qua, chúng tôi đã dịch qua 16 môn học, 34 cuốn sách, 43 bài báo, 5 sổ tay (chưa tính các tài liệu từ năm 2010 trở về trước) Xem ở đây

**DỊCH VỤ
DỊCH
TIẾNG
ANH
CHUYÊN
NGÀNH
NHANH
NHẤT VÀ
CHÍNH
XÁC
NHẤT**

Chỉ sau một lần liên lạc, việc dịch được tiến hành

Giá cả: có thể giảm đến 10 nghìn/1 trang

Chất lượng: Tạo dựng niềm tin cho khách hàng bằng công nghệ 1. Bạn thấy được toàn bộ bản dịch; 2. Bạn đánh giá chất lượng. 3. Bạn quyết định thanh toán.

**BỘ GIAO THÔNG VẬN TẢI
TRƯỜNG ĐẠI HỌC HÀNG HẢI
BỘ MÔN: TIN HỌC ĐẠI CƯƠNG
KHOA: CÔNG NGHỆ THÔNG TIN**

BÀI GIẢNG TIN HỌC ĐẠI CƯƠNG

**TÊN HỌC PHẦN : TIN HỌC ĐẠI CƯƠNG
MÃ HỌC PHẦN : 17101
TRÌNH ĐỘ ĐÀO TẠO : ĐẠI HỌC CHÍNH QUY
DÙNG CHO SV NGÀNH : TẤT CẢ CÁC NGÀNH**

MỤC LỤC

STT	NỘI DUNG	TRANG
CHƯƠNG I	NHẬP MÔN TIN HỌC ĐẠI CƯƠNG	
1.1	Một vài nét về tin học và máy tính	4
1.2	Thông tin và xử lý thông tin	4
1.3	Thành phần của máy tính	4
1.4	Các hệ đếm	5
1.5	Phân loại máy tính	9
1.6	Các bộ phận chính của máy tính PC	10
CHƯƠNG II	TỔNG QUAN VỀ LẬP TRÌNH	
2.1	Vấn đề giải quyết bài toán bằng máy tính	12
2.2	Một số ví dụ	13
CHƯƠNG III	NGÔN NGỮ LẬP TRÌNH VISUAL BASIC	
3.1	Khởi động Visual Basic	20
3.2	Thoát khỏi Visual Basic	20
3.3	Tạo, mở và lưu một dự án (Project)	20
3.4	Các bước xây dựng chương trình	21
3.5	Một số đối tượng cơ bản trong Visual Basic	23
3.6	Một số ví dụ về thiết kế giao diện và lập bảng thuộc tính	29
3.7	Các kiểu dữ liệu cơ bản	30
CHƯƠNG IV	KHAI BÁO HẰNG BIẾN BIỂU THỨC CÂU LỆNH	
4.1	Các hằng trong Visual Basic	36
4.2	Biến	36
4.3	Biểu thức	37
4.4	Quy định về viết dòng lệnh trong Visual Basic	37
4.5	Cách viết dòng chú thích trong chương trình	37
4.6	Câu lệnh gán	37
4.7	Câu lệnh End	38
4.8	Lệnh in dữ liệu	38

4.9	Lệnh nhập dữ liệu	39
4.10	Chuyển đổi kiểu dữ liệu trong quá trình tính giá trị biểu thức	40
4.11	Cấu trúc điều kiện	41
4.12	Câu lệnh lựa chọn Select	43
4.13	Cấu trúc lặp	45
4.14	Một số ví dụ	48
CHƯƠNG V	HÀM VÀ THỦ TỤC	
5.1	Khái niệm chương trình con	51
5.2	Hàm và thủ tục	51
5.3	Truyền tham số cho chương trình con	53
5.4	Biến toàn cục, biến địa phương, khái niệm tầm tác dụng	56
CHƯƠNG VI	DỮ LIỆU CÓ CẤU TRÚC, KIỂU MẢNG	
6.1	Khái niệm về mảng	59
6.2	Khai báo mảng tĩnh	59
6.3	Khai báo mảng động (Dynamic Array)	61
6.4	Một số thuật toán về mảng	62

CHƯƠNG 1

NHẬP MÔN TIN HỌC ĐẠI CƯƠNG

1.1. Một vài nét về tin học và máy tính

Có rất nhiều khái niệm về Tin học, song có thể hiểu Tin học là ngành khoa học nghiên cứu các quá trình có tính chất thuật toán nhằm mô tả và biến đổi thông tin. Các quá trình này được nghiên cứu một cách hệ thống về mọi phương diện: Lý thuyết phân tích, thiết kế, tính hiệu quả, việc cài đặt và các ứng dụng.

1.2. Thông tin và xử lý thông tin

1.2.1. Thông tin và dữ liệu

Thông tin (Information): Những hiện tượng sự vật phản ánh về một sự kiện, một vấn đề nào đó trong thế giới khách quan. Dựa vào đó con người có thể hiểu biết và nhận thức được thế giới khách quan. Thông tin có thể ghi lại và truyền đi.

Dữ liệu (Data): Là dạng thông tin khi được lưu trữ.

1.2.2. Đơn vị đo thông tin

Đơn vị đo thông tin là bit. Đây chính là tin về hệ thống chỉ có hai trạng thái đồng khả năng: bằng 0 hoặc bằng 1 (Điều này rất phù hợp với các máy tính điện tử bởi trong một thời điểm, mạch điện chỉ có một trong hai trạng thái đóng hoặc mở tương ứng với hai giá trị 1 hoặc 0).

Đơn vị đo thông tin bao gồm: Bit, Byte, KiloByte, MegaByte, GigaByte.

Qui đổi giữa các đơn vị như sau:

Byte (B): 1 Byte = 8 bit (b)

KiloByte (KB): 1 KB = 1024 Byte

MegaByte (MB): 1 MB = 1024 KB

GigaByte (GB): 1 GB = 1024 MB

Trong đó: **b** là viết tắt của bit

B là viết tắt của Byte

1.3. Thành phần của máy tính

1.3.1. Phần cứng (Hardware)

Là các thành phần vật lý cấu tạo nên máy tính. Các thành phần vật lý ở đây bao gồm các thiết bị điện tử và cơ khí.

Ví dụ: về các phần cứng máy tính như màn hình, bàn phím, chuột, bộ vi xử lý...

1.3.2. Phần mềm (SoftWare)

Là tập hợp các chỉ thị cho máy tính làm việc. Nói cách khác, toàn bộ các chương trình chạy trên máy tính gọi là phần mềm máy tính. Sự ra đời của phần mềm khiến cho hiệu quả sử dụng phần cứng được nâng cao, rất nhiều công việc của con người được tự động hoá, vận hành nhanh chóng.

Ví dụ: Phần mềm soạn thảo văn bản, bảng tính, trình diễn, đồ hoạ...

1.4. Các hệ đếm

1.4.1. Hệ đếm thập phân (Decimal)

Hệ thập phân là hệ đếm dựa vào vị trí với cơ số 10. Hệ này dùng các số từ 0 đến 9 để biểu diễn. Giá trị của hệ thập phân được đánh giá bằng vị trí các con số.

Cách viết: 127 hoặc $(127)_{10}$

Khai triển một số hệ 10:

$$(123.78)_{10} = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0 + 7 \times 10^{-1} + 8 \times 10^{-2}$$

1.4.2. Hệ đếm nhị phân (Binary)

Hệ nhị phân hay hệ đếm cơ số 2 chỉ có hai con số 0 và 1. Đó là hệ đếm theo vị trí. Giá trị của một số bất kỳ nào đó phụ thuộc vào vị trí của nó. Các vị trí có trọng số. Các vị trí của trọng số bằng bậc lũy thừa của cơ số 2.

Chấm cơ số được gọi là chấm nhị phân trong hệ đếm cơ số 2. Mỗi một con số nhị phân được gọi là một bit (Binary digit). Bit ngoài cùng bên trái là bit có trọng số lớn nhất (MSB) và bit ngoài cùng bên phải là bit có trọng số nhỏ nhất (LSB) như dưới đây:

$$\begin{array}{ccccccc} 2^3 & 2^2 & 2^1 & 2^0 & & 2^{-1} & 2^{-2} \\ \text{MSB} & 1 & 0 & 1 & 0 & \blacksquare & 1 & 1 & \text{LSB} \end{array}$$

Cách viết: $(1011)_2$

1.4.3. Hệ thập lục phân (HEXADECIMAL)

Các máy tính hiện đại thường dùng hệ đếm khác là hệ thập lục phân. Hệ thập lục phân là hệ đếm dựa vào vị trí với cơ số 16.

Hệ này dùng 10 chữ số từ 0 đến 9 và 6 ký tự từ A đến F để biểu diễn.

Hệ này thường dùng để viết gọn các số hệ nhị phân

Hệ này được biểu diễn như trong bảng sau:

Thập lục phân	Thập phân	Nhị phân
0	0	0000
1	1	0001
2	2	0010

3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

Cách viết : $(2BC)_H$

1.4.4. Chuyển đổi giữa các hệ đếm

* Chuyển từ hệ nhị phân sang hệ mười

Muốn đổi từ hệ nhị phân sang hệ mười chỉ cần tính các giá trị 2^i tương ứng với các chữ số khác không thứ i của hệ số nhị phân rồi cộng lại : (vị trí i tính từ phải qua trái)

Ví dụ 1: Ta muốn đổi một số hệ nhị phân $(101001)_2$ sang hệ mười, tiến hành như sau:

$$\begin{aligned} (101001)_2 &= 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &= 32 + 0 + 8 + 0 + 0 + 1 = (41)_{10} \end{aligned}$$

Kết quả ta được số 41 ở hệ 10

Ví dụ 2: Ta muốn đổi một số hệ nhị phân $(101.001)_2$ sang hệ mười, tiến hành như sau:

$$\begin{aligned} (101.001)_2 &= 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} \\ &= 4 + 0 + 2 + 0 + 0 + 0.125 = (6.125)_{10} \end{aligned}$$

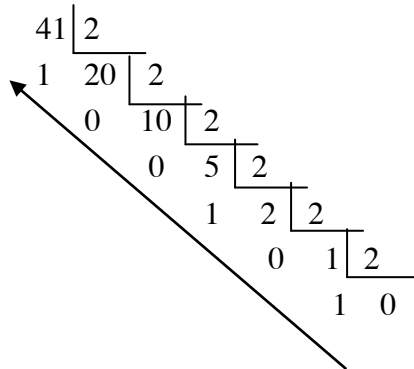
Chú ý: Trong các ví dụ trên quá trình biến đổi có thể đơn giản hoá bằng cách loại bỏ các con số không.

* Chuyển từ hệ thập phân sang hệ nhị phân:

Lấy phần nguyên của số cần đổi chia cho 2 và ghi nhớ phần dư, tiếp theo lấy thương của phép chia trước đó chia cho 2 và ghi nhớ phần dư. Cứ làm như vậy cho đến khi thương bằng 0 thì dừng lại và viết phần dư ngược từ dưới lên trên. Lấy phần lẻ thập phân của số cần đổi nhân với 2, tích nhận được gồm phần nguyên và phần lẻ nhị phân, lấy phần lẻ nhị phân của tích thu được nhân với 2. Cứ làm như vậy cho đến khi không còn

phần lẻ thập phân hoặc đến một độ chính xác nhất định nào đó thì dừng lại. Viết phần nguyên từ trên xuống dưới. Sau đó ghép kết quả của chúng lại với nhau.

Ví dụ 1: Đổi số $(41)_{10}$ từ hệ 10 sang hệ nhị phân:



Sau khi chia đến thương bằng 0 và viết ngược phần dư từ dưới lên trên ta được số $(101001)_2$ ở nhị phân.

Ví dụ 2: Biến đổi số thập phân $(41.625)_{10}$ thành số nhị phân:

Trước hết biến đổi phần nguyên thành số nhị phân theo ví dụ 1 được số $(101001)_2$

Sau đó biến đổi phần lẻ thành số nhị phân bằng cách: Lấy số cần đổi nhân với 2, tích nhận được gồm phần nguyên và phần lẻ nhị phân, lấy phần lẻ nhị phân của tích thu được nhân với 2. Cứ làm như vậy cho đến khi được tích chẵn bằng 1. Chọn riêng các phần nguyên (phần trước dấu phẩy) của các tích thu được và sắp xếp lại sẽ được các chữ số sau dấu phẩy cần tìm.

$$\begin{array}{r} 0.625 \times 2 = 1.250 \\ 1.250 \times 2 = 0.500 \\ 0.5 \times 2 = 1.000 \end{array}$$

phần lẻ $(0.625)_{10}$ được chuyển thành $(101)_2$ hệ 2

Vậy số $(41.625)_{10}$ ở hệ 10 sang hệ nhị phân là $(101001.101)_2$

***Chuyển từ hệ thập lục phân sang hệ thập phân:**

Các số thập lục phân có thể được biến đổi thành thập phân bằng cách tính tổng của các con số nhân với giá trị vị trí của nó.

Ví dụ1: Đổi số $(5B)_{16}$ thành số thập phân

$$\begin{aligned} (5B)_{16} &= 5 \times 16^1 + B \times 16^0 \\ &= 5 \times 16^1 + 11 \times 16^0 \\ &= 80 + 11 = (91)_{10} \end{aligned}$$

Kết quả là số $(91)_{10}$ hệ 10

Ví dụ 2: Đổi số $(2AF)_{16}$ thành số thập phân

$$\begin{aligned} (2AF)_{16} &= 2 \times 16^2 + A \times 16^1 + F \times 16^0 \\ &= 2 \times 16^2 + 10 \times 16^1 + 15 \times 16^0 \\ &= 2 \times 256 + 10 \times 16 + 15 \times 1 \\ &= 512 + 160 + 15 = (687)_{10} \end{aligned}$$

Kết quả là số $(687)_{10}$ hệ 10

***Chuyển từ hệ thập phân sang hệ thập lục phân:**

Lấy số thập phân cần đổi chia cho 16 và ghi nhớ phần dư, tiếp theo lấy thương của phép chia trước đó chia cho 16 và ghi nhớ phần dư. Cứ làm như vậy cho đến khi thương bằng 0 thì dừng lại và viết phần dư ngược từ dưới lên trên.

Ví dụ: Đổi số $(93)_{10}$ thành hệ thập lục phân:

	Thương số	+	số dư	
$\frac{93}{16}$	= 5	+	13	tương ứng là D
$\frac{5}{16}$	= 0	+	5	

Vậy kết quả là số $(93)_{10}$ đổi thành số $(5D)_{16}$ hệ thập lục phân

***Chuyển từ hệ thập lục phân sang hệ nhị phân:**

Thực chất các số thập lục phân cũng chỉ là cách biểu diễn các số nhị phân thuận lợi hơn. Muốn đổi từ số thập lục phân sang số hệ nhị phân ta chỉ cần thay thế từ con số thập lục phân bằng 4 bit nhị phân tương ứng:

Ví dụ: Đổi số $(C5)_{16}$ thành hệ nhị phân



Kết quả số $(C5)_{16}$ thành số $(11000101)_2$

***Chuyển từ hệ nhị phân sang hệ thập lục phân:**

Để biến đổi một số nhị phân thành số thập lục phân tương đương chỉ cần nhóm 4 bit tính từ phải qua trái sau đó chuyển từng nhóm 4 bit thành hệ thập lục phân và ghép lại với nhau sẽ được số hệ thập lục phân.

Ví dụ: Đổi số $(01011111)_2$ thành số thập lục phân



Kết quả số $(01011111)_2$ thành số $(5F)_{16}$

1.5. Phân loại máy tính

Theo góc độ người sử dụng thì máy tính được phân làm bốn loại sau:

* Máy tính lớn (Mainframe)

Máy tính lớn là máy có công suất cao, tốc độ xử lý nhanh và rất lớn thường đóng vai trò trong các hệ tin học phân tán có qui mô lớn.

Các máy tính lớn là các cỗ máy kích thước lớn, mạnh và rất đắt tiền, được sử dụng trong hầu hết các cơ quan, tổ chức lớn (ví dụ như các ngân hàng, các cơ quan trọng yếu của Chính phủ, các công ty đa quốc gia...).

Cần chú ý phân biệt máy tính lớn với máy tính cá nhân có kích thước lớn.

* Máy tính PC (Personal Computer-PC)

Máy tính điện tử có nguồn gốc ra đời từ khá sớm và có rất nhiều chủng loại. Tuy nhiên chiếc máy tính PC theo mô hình của IBM đầu tiên được ra đời vào năm 1981. Từ đó trở đi, tất cả các máy tính PC được sản xuất, qua nhiều lần mở rộng đều tương thích với thiết kế ban đầu. Thuật ngữ máy tính tương thích liên quan đến các máy tính được sản xuất bởi các công ty khác không phải IBM nhưng vẫn tương thích với các đặc điểm kỹ thuật truyền thống.



**IBM – PC
(1981)**



**IBM – PC
Tương thích**

* Máy Mac(Apple MAC)

Máy Mac là một máy tính, nhưng không phải là một máy PC theo mô hình IBM. Máy Mac là dòng máy tính do hãng Apple sản xuất và chạy Hệ điều hành Macintosh của Apple có giao diện đồ họa GUI và tương thích với nó là các phiên bản đặc biệt của các chương trình ứng dụng (như bộ xử lý văn bản và các bảng tính). Thậm chí để có thể kết nối với một Mac, người ta cần phải lựa chọn thêm các thiết bị phần cứng mở rộng.

Thời kỳ đầu, điều giúp cho việc phân biệt máy Mac và PC chính là việc MAC sử dụng giao diện đồ họa cho người dùng (GUI – Graphic User Interface), hay nói cách khác, chính là cách mà ta có thể dùng chuột để điều khiển máy tính. Khi PC mới xuất hiện, người dùng phải là một nhà chuyên môn mới có thể sử dụng được chiếc PC đó. Tuy nhiên, gần đây sự khác biệt



cơ bản giữa máy MAC và máy PC đã không còn rõ rệt như trước, nhất là khi Microsoft đã sử dụng Graphic User Interface như là công cụ trợ giúp người dùng máy tính PC.

(Chú ý: Ý tưởng sử dụng các giao diện đồ họa và tương tác người – máy thông qua thao tác chuột là do Apple đề xuất, không phải do Microsoft)

*** Máy tính xách tay (Laptop)**

Laptop, tiếng Anh có nghĩa là đặt trong lòng, phân biệt với máy tính đặt trên mặt bàn là desktop. Ở Việt Nam, ta hay gọi Laptop là máy tính xách tay bởi đây là một loại máy tính nhỏ có thể mang đi theo, có thể chạy bằng pin. Ngoài ra, người ta còn sử dụng tên gọi “Notebooks” (quyển vở/quyển sổ) để chỉ các máy tính xách tay có kích thước nhỏ hơn.



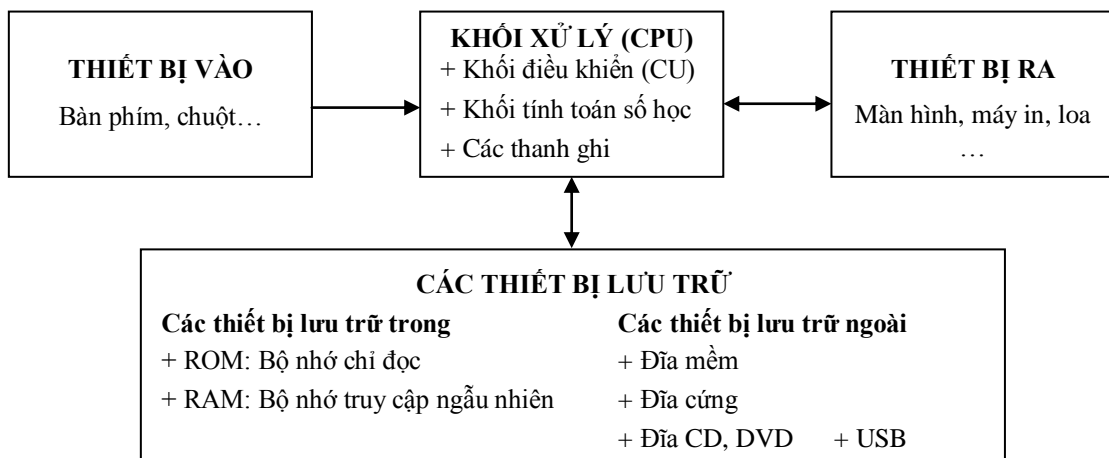
Cả hai dòng máy tính PC và MAC đều có những sản phẩm máy tính xách tay cho người dùng của mình. Do đặc điểm nhỏ gọn và năng lực xử lý khá tốt nên máy tính xách tay rất được ưa chuộng, đặc biệt là những người hay phải di chuyển hoặc phải thuyết trình trước đám đông. Những người thường xuyên sử dụng máy tính xách tay có thể kể đến những người làm công tác kinh doanh (chuyên đi thuyết trình sản phẩm), những người làm trong ngành giáo dục (giảng bài bằng máy tính xách tay kết nối máy chiếu)...

Chú ý: Thực chất máy tính xách tay cũng là dòng máy PC hay máy Mac là dòng máy PC nhưng do hãng Apple sản xuất.

1.6. Các bộ phận chính của máy tính PC

1.6.1. Sơ đồ các khối chức năng và bộ phận chính trong máy tính

Về cơ bản, thành phần chính của một máy tính bao gồm các khối như: Khối thiết bị vào/ Khối xử lý/ Khối thiết bị ra/ Khối thiết bị lưu trữ.



1.6.2. Khối xử lý trung tâm (Central Processing Unit – CPU)

Khối xử lý trung tâm, hay còn gọi là một bộ vi xử lý hoặc con chip, là bộ não của máy tính. Công việc chính của khối xử lý trung tâm là tính toán và điều khiển mọi hoạt động trong máy tính.



1.6.3. Bộ nhớ trong (Internal Memory)

Bộ nhớ trong máy tính dùng để chứa các lệnh và dữ liệu phục vụ cho quá trình thực hiện các chương trình.

Bộ nhớ ROM (Read Only Memory): Dữ liệu được đưa vào từ nhà sản xuất, thông tin trong ROM sẽ không bị mất khi tắt máy. Bộ nhớ ROM không thể nâng cấp hoặc thay thế.

Bộ nhớ RAM (Random Access Memory): Là nơi lưu trữ tạm thời dữ liệu của người sử dụng trong quá trình xử lý. Dữ liệu trên RAM sẽ bị mất khi tắt máy. Bộ nhớ Ram có thể được nâng cấp hoặc thay thế.

1.6.4. Bộ nhớ ngoài (External Memory)

Bộ nhớ ngoài hay các thiết bị lưu trữ ngoài bao gồm đĩa cứng, đĩa mềm, đĩa CD, DVD, đĩa Zip...

1.6.5. Các thiết bị vào (Input device)

Các thiết bị vào cho phép thông tin hay dữ liệu được nhập vào máy tính, ví dụ như bàn phím, chuột, máy quét...

1.6.6. Các thiết bị ra (Output device)

Các thiết bị ra cho phép thông tin có thể được xuất ra từ máy tính, ví dụ như máy in, màn hình, loa...

1.6.7. Các thiết bị ngoại vi (Peripheral device)

Thiết bị ngoại vi là thiết bị mà nó có thể gắn/cắm vào máy tính. Như vậy toàn bộ các thiết bị như máy quét, máy in, bàn phím, chuột, loa... đều là các thiết bị ngoại vi.

CHƯƠNG 2

TỔNG QUAN VỀ LẬP TRÌNH

2.1. Vấn đề giải quyết bài toán bằng máy tính

2.1.1. Lập trình (Programming)

Lập trình dùng để chỉ thao tác của con người nhằm kiến tạo, xây dựng một kịch bản bằng các thuật toán để giải quyết một bài toán Tin học trên máy tính thông qua các ngôn ngữ lập trình.

2.1.2. Ngôn ngữ lập trình (Programming language)

Ngôn ngữ lập trình là một tập con của ngôn ngữ máy tính. Đây là một dạng ngôn ngữ được chuẩn hóa (đối lập với ngôn ngữ tự nhiên). Nó là một hệ thống được ký hiệu hóa để miêu tả những tính toán (qua máy tính) trong một dạng mà cả con người và máy đều có thể đọc và hiểu. Ngôn ngữ lập trình cũng chính là một chương trình, nhưng có thể được dùng để tạo nên các chương trình khác. Một chương trình máy tính được viết bằng một ngôn ngữ lập trình thì những chỉ thị (của riêng ngôn ngữ ấy) góp phần tạo nên chương trình được gọi là mã nguồn của chương trình ấy.

2.1.3. Bài toán Tin học

Bài toán tin học không dùng để chỉ một bài toán cụ thể, mà dùng để chỉ một lớp các bài toán cụ thể thuộc cùng một loại. Một bài toán tin học được cấu tạo bởi hai yếu tố sau:

- + Thông tin đầu vào (Input): Là những thông tin bài toán đã cho
- + Thông tin đầu ra (Output): Là các thông tin cần tìm hoặc câu trả lời cần thiết.

Ví dụ: Giải bài toán tính diện tích tam giác bằng công thức $s=a*h/2$

- + Thông tin đầu vào: Là cạnh đáy a và đường cao h
- + Thông tin đầu ra: Là diện tích tam giác S hoặc dòng thông báo dữ liệu không hợp lệ.

2.1.4. Thuật toán

Thuật toán là tập hợp hữu hạn các bước, các phép toán cơ bản nhất được sắp xếp theo một trình tự nhất định để từ thông tin đầu vào của bài toán sau một tập hữu hạn các bước đó sẽ đạt được kết quả ở đầu ra như mong muốn.

Ví dụ: Trờ lại bài toán tính diện tích tam giác, thuật toán như sau:

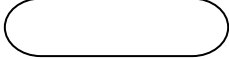
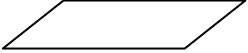
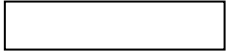
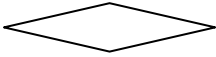


- + **Bước 1:** Cho giá trị của cạnh đáy a và đường cao h
- + **Bước 2:** Kiểm tra
 - Nếu $a>0$ và $h>0$ thì tính diện tích theo công thức $s=a*h/2$ sau đó xuống bước 3.
 - Ngược lại thông báo dữ liệu không hợp lệ và quay về bước 1
- + **Bước 3:** In diện tích S

2.1.5. Biểu diễn thuật toán qua sơ đồ khối (A.Flow Chart)

Biểu diễn thuật toán: Là sơ đồ dùng các ký hiệu và các đường nối với nhau để diễn tả các quan hệ logic và trình tự thực hiện các phép toán của một chương trình máy tính.

Để thể hiện trực quan về thuật toán người ta dùng sơ đồ khối. Qua sơ đồ khối người lập trình có thể quan sát các bước của thuật toán cũng như dòng thông tin hình thành và biến đổi trong quá trình thuật toán làm việc.

Việc thể hiện thuật toán người ta quy ước dùng các khối sau để thể hiện.

-  - Thể hiện sự bắt đầu và kết thúc chương trình
-  - Thể hiện việc nhập, xuất dữ liệu
-  - Chứa các công thức tính toán
-  - Chứa các biểu thức kiểm tra điều kiện và rẽ nhánh chương trình
-  - Gọi chương trình con
-  - Chỉ ra hướng đi của thuật toán và kết nối giữa các hình

2.2. Một số ví dụ

Ví dụ 1:

Lập thuật toán và vẽ sơ đồ thuật toán để giải và biện luận phương trình $ax + b = 0$

Bài giải:

Xây dựng thuật toán:

- Dữ liệu đầu vào là giá trị hai số a và b (trong lập trình thường gọi tắt là nhập giá trị a và b)

- Kiểm tra điều kiện của a:

Nếu $a = 0$ thì

+ Nếu $b=0$ thì

Phương trình vô số nghiệm

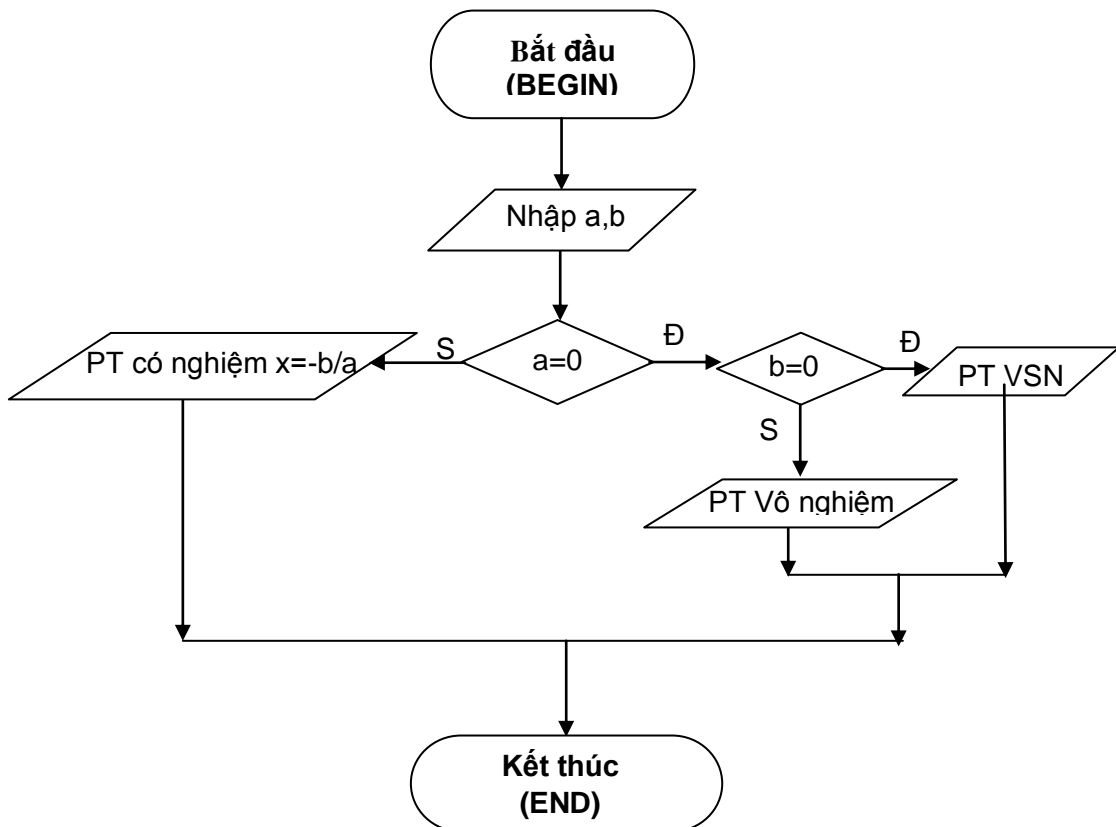
Ngược lại

Phương trình vô nghiệm

Ngược lại

Phương trình có nghiệm $x=-b/a$

Sơ đồ thuật toán:



Ví dụ 2:

Cho ba số bất kỳ a,b,c hãy tìm giá trị lớn nhất trong ba số này

Bài giải:

Xây dựng thuật toán:

- Nhập giá trị cho ba biến a,b,c
- Coi một trong ba số là số lớn nhất (giả sử số a) đưa giá trị của a vào một biến max

Mô tả trên được viết như sau $max=a$

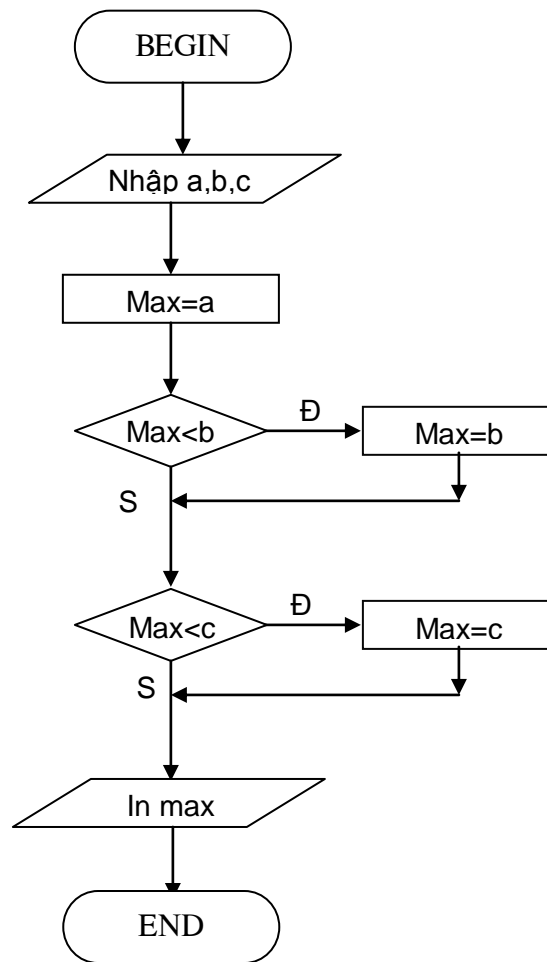
- Kiểm tra giá trị của biến max với từng số còn lại

Nếu $max < b$ thì $max=b$

Nếu $max < c$ thì $max=c$

- In giá trị lớn nhất trong max ra màn hình

Sơ đồ thuật toán:



Ví dụ 3: Hãy tính Tổng $S=1+1/2+1/3+\dots+1/(n-1)+1/n$ với n là một số tự nhiên.

Bài giải:

Xây dựng thuật toán:

Bước 1: Nhập giá trị cho n

Bước 2: Kiểm tra giá trị của $n > 0$

+ Nếu sai: Trở về bước 1

+ Nếu đúng: Chuyển sang bước 3

Bước 3: Cho $S=0$

Khởi tạo biến chạy i và cho $i=1$

Bước 4: Kiểm tra điều kiện của i

- Nếu $i \leq n$ thì:

+ Tính $S=S+1/i$

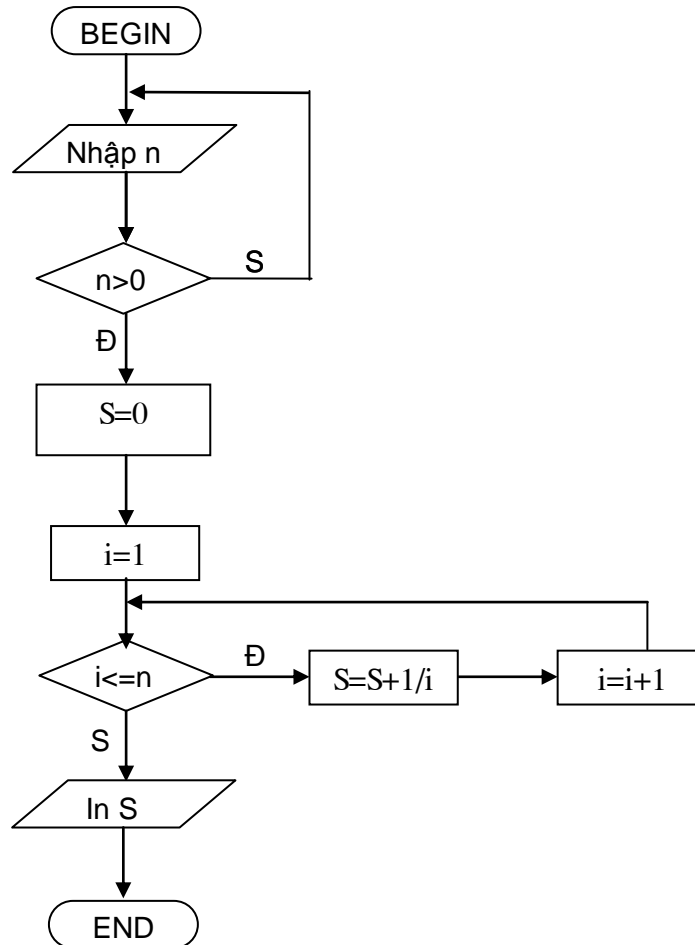
+ Tăng i : $i=i+1$

+ Quay trở về đầu bước 4

- Nếu $i > n$: Chuyển sang bước 5

Bước 5: In giá trị của S

Sơ đồ thuật toán:



Ví dụ 4:

Nhập vào một dãy số nguyên gồm n phần tử. Sau đó tính trung bình cộng các số chia hết cho 3.

Phân tích bài toán:

Dữ liệu đầu vào:

- Số phần tử của dãy n (n là số nguyên dương)
- Dãy số có n phần tử bất kỳ

Dữ liệu đầu ra: - Trung bình cộng của các số chia hết cho 3

Để tính được trung bình cộng ta cần phải tính Tổng và số lượng các số chia hết cho 3. Theo dữ liệu đầu vào của bài toán ta có thể giải quyết theo hai hướng sau:

Hướng 1: Lưu trữ dãy số bằng một ô nhớ	Hướng 2: Lưu trữ dãy số bằng n ô nhớ
<p>Bước 1: - Xác định số phần tử của dãy, nhập giá trị cho n.</p> <p>Bước 2:</p> <ul style="list-style-type: none"> - Kiểm tra điều kiện n <ul style="list-style-type: none"> + Nếu $n \leq 0$: Quay về bước 1 + Nếu $n > 0$: Chuyển sang bước 3 <p>Bước 3:</p> <p>Để tính TBC trước hết ta phải tính:</p> <ul style="list-style-type: none"> + Tổng các số thoả mãn, gọi tổng đó là T, khởi tạo giá trị ban đầu $T=0$ + Số lượng các phần tử thoả mãn, gọi là d, khởi tạo giá trị ban đầu $d=0$ - Khởi tạo biến chạy $i=1$ (coi như khi $i=1$ ta tiến hành nhập giá trị cho phần tử thứ nhất) <p>Bước 4:</p> <p>Kiểm tra điều kiện của i.</p> <ul style="list-style-type: none"> - Nếu $i \leq n$ <ul style="list-style-type: none"> + Nhập giá trị cho a + Kiểm tra: Nếu a chia hết cho 3 thì $T=T+a$ $d=d+1$ + Tăng biến i lên một đơn vị: $i=i+1$ + Quay về bước 4 - Nếu $i > n$: Chuyển sang bước 5 <p>Bước 5:</p> <ul style="list-style-type: none"> - Sau khi tính được T và d ta cần kiểm tra lại d vì có thể xảy ra trường hợp trong n phần tử không có phần tử nào chia hết cho 3 (tương ứng với $d=0$) thì không thể tính được TBC. - Nếu $d=0$ thì <ul style="list-style-type: none"> Dãy số không có phần tử thoả mãn 	<p>Bước 1: - Xác định số phần tử của dãy, nhập giá trị cho n.</p> <p>Bước 2:</p> <ul style="list-style-type: none"> - Kiểm tra điều kiện n <ul style="list-style-type: none"> + Nếu $n \leq 0$: Quay về bước 1 + Nếu $n > 0$: Chuyển sang bước 3 <p>Bước 3:</p> <p>Để tính TBC trước hết ta phải tính:</p> <ul style="list-style-type: none"> + Tổng các số thoả mãn, gọi tổng đó là T, khởi tạo giá trị ban đầu $T=0$ + Số lượng các phần tử thoả mãn, gọi là d, khởi tạo giá trị ban đầu $d=0$ - Khởi tạo biến chạy $i=1$ (coi như khi $i=1$ ta tiến hành nhập giá trị cho phần tử thứ nhất) <p>Bước 4:</p> <p>‘Nhập giá trị liên một lúc cho n phần tử chứa vào n ô nhớ a_1, a_2, \dots, a_n</p> <p>Kiểm tra điều kiện của i.</p> <ul style="list-style-type: none"> - Nếu $i \leq n$ <ul style="list-style-type: none"> + Nhập giá trị cho phần tử a_i + Tăng biến i lên một đơn vị: $i=i+1$ + Quay về bước 4 - Nếu $i > n$: Chuyển sang bước 5 <p>Bước 5:</p> <p>‘ Tính toán TBC</p> <ul style="list-style-type: none"> - Khởi tạo biến chạy $i=1$ <p>Bước 6:</p> <p>Kiểm tra điều kiện của i.</p> <ul style="list-style-type: none"> - Nếu $i \leq n$ <ul style="list-style-type: none"> + Kiểm tra: Nếu a chia hết cho 3 thì $T=T+a$ $d=d+1$ + Tăng biến i lên một đơn vị: $i=i+1$

điều kiện chia hết cho 3
ngược lại

Tính trung bình cộng: $TBC = T/d$

In giá trị của TBC ra

Bước 6: Kết thúc

+ Quay về bước 6

- Nếu $i > n$: Chuyển sang bước 7

Bước 7:

- Sau khi tính được T và d ta cần kiểm tra lại d vì có thể xảy ra trường hợp trong n phần tử không có phần tử nào chia hết cho 3 (tương ứng với $d=0$) thì không thể tính được TBC.

- Nếu $d=0$ thì

Dãy số không có phần tử thoả mãn điều kiện chia hết cho 3

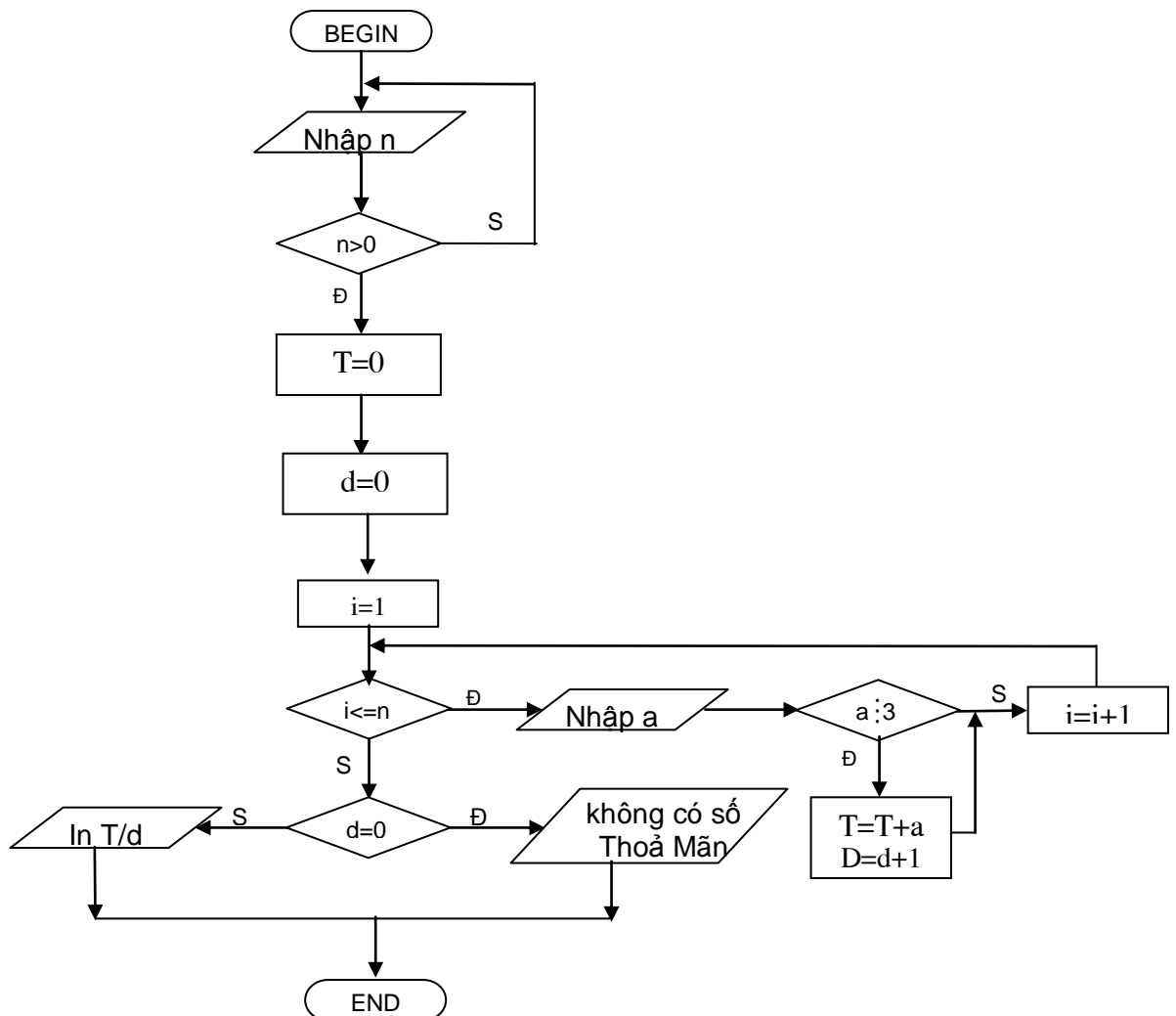
ngược lại

Tính trung bình cộng: $TBC = T/d$

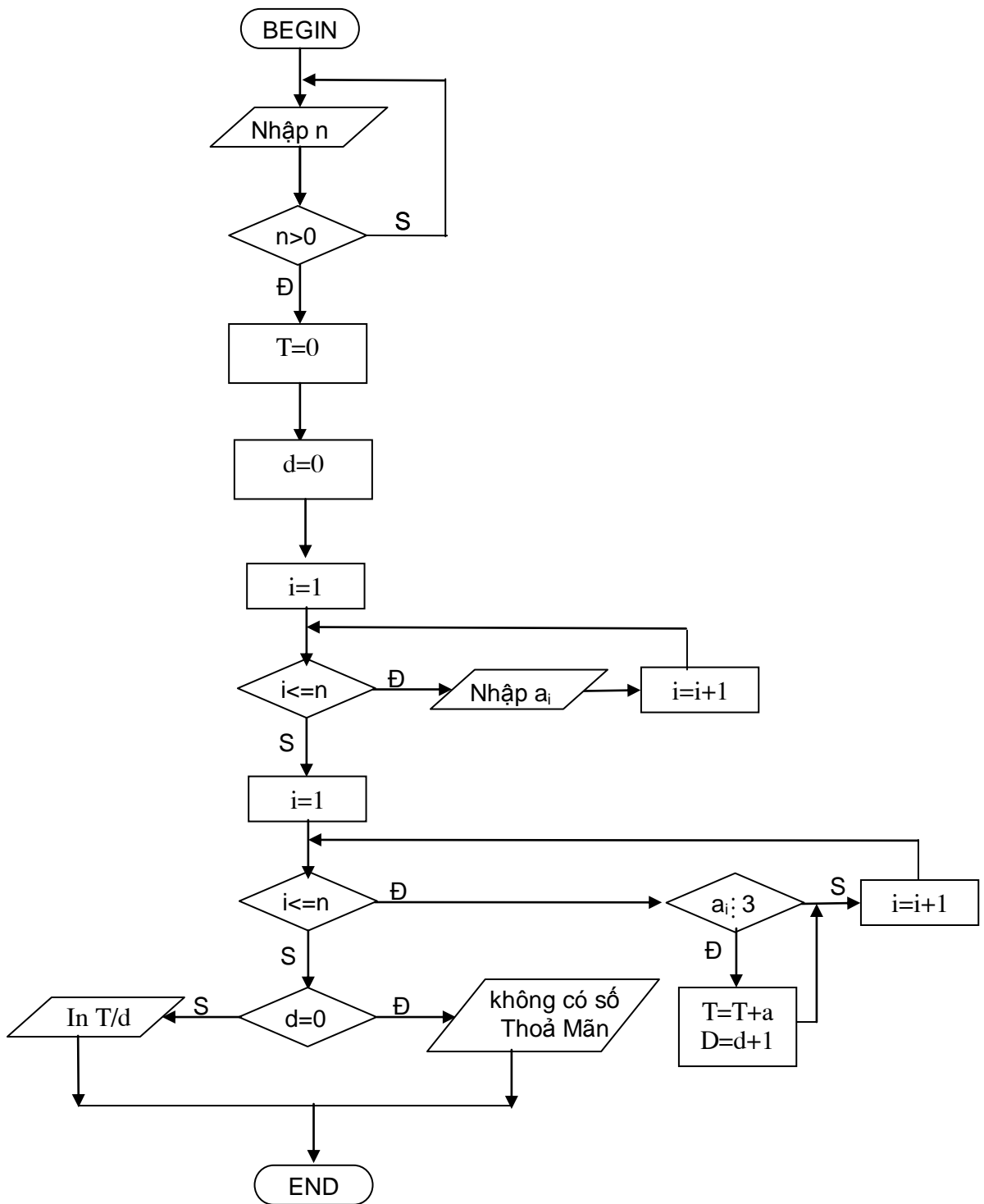
In giá trị của TBC ra

Bước 8: Kết thúc

Sơ đồ thuật toán theo hướng 1:



Sơ đồ thuật toán theo hướng 2:



CHƯƠNG 3

NGÔN NGỮ LẬP TRÌNH VISUAL BASIC

3.1. Khởi động Visual Basic

Kích chuột vào nút **Start\Programs\MiscoSoft Visual Studio 6.0\MicroSoft Visual Basic 6.0**

3.2. Thoát khỏi Visual Basic

Để thoát khỏi Visual Basic ta làm theo cách sau:

Vào menu **File\Exit**. Khi thoát khỏi Visual Basic nếu ta chưa ghi lại chương trình và các thành phần liên quan đến chương trình Visual Basic sẽ nhắc nhở chúng ta ghi lại.

3.3. Tạo, mở và lưu một dự án (Project)

Bước 1: Trên thư mục gốc của ổ đĩa C tạo thư mục có tên là VIDU

Lý do cần phải tạo riêng ra một thư mục cho chương trình vì một chương trình viết bằng Visual Basic bao gồm nhiều thành phần như: Các màn hình giao diện (Form), các đơn thể chương trình (Modul), các báo cáo (Report)... mỗi một thành phần sẽ được tạo các file tương ứng trên ổ đĩa; chính vì vậy để quản lý chương trình cho tốt phong cách lập trình tốt là hãy tạo riêng cho mỗi chương trình một thư mục riêng.

Bước 2: Khởi động VB khi đó hộp hội thoại **New Project** xuất hiện:



Trong hộp hội thoại này chọn tab New sau đó chọn loại Project muốn tạo, ở đây ta sẽ chọn StandardEXE. Hai tab còn lại có ý nghĩa sau:

- Existing: Mở Project đã có
- Recent: Mở Project đã làm việc trong thời gian gần đây nhất.

Một cách khác để tạo dự án mới đó là trong menu File chọn mục New Project.

Sau khi tạo Project mới, Visual Basic sẽ hiện một cửa sổ trên màn hình. Đây là Form trống với tiêu đề là Form1

Bước 3: Lưu Project

Một trong những vấn đề quan trọng khi viết chương trình là lưu lại những gì mình đã làm. Mặc dù chưa có thao tác gì thay đổi trên Form trống, ta cũng nên lưu Form lên đĩa. Khi ta lưu Project, có 2 file được lưu:

1. File chứa các thông tin Visual Basic cần trong việc xây dựng Project. Đây là file quản lý toàn bộ các thành phần của dự án, file này có phần mở rộng là VBP
2. File chứa các thông tin về Form. File này có phần mở rộng là FRM.

Các bước thực hiện lưu Project như sau:

- Vào menu File chọn mục Save Project As
- Visual Basic hiện hộp hội thoại Save File As. Chọn thư mục C:\VIDU1 đổi tên mặc định của Form trống từ Form1.Frm thành VIDU1.FRM.

Lưu ý: Không nên dùng tên mặc định mà Visual Basic đặt cho Form, ta nên đặt tên form gần gũi với các chức năng của Form ta đang muốn xây dựng.

- Sau khi ghi Form, Visual Basic hiện tiếp hộp hội thoại Save Project As, file Project có tên mặc định là Project1.Vbp. Ta chọn thư mục C:\VIDU1 và đổi tên cho file Project là VIDU1.VBP.

Lưu ý: Không nên dùng tên mặc định mà Visual Basic đặt cho Project, ta nên đặt tên Project gần gũi với chương trình đang muốn xây dựng.

Như vậy hai file VIDU1.FRM và VIDU1.VBP đã được lưu vào thư mục VIDU1 (lúc này Visual Basic sẽ phát sinh một file phụ là VIDU1.VBW)

3.4. Đối tượng và thuộc tính

3.4.1. Đối tượng

Trong Visual Basic đối tượng là tất cả những thành phần mà ta sử dụng để thiết kế lên giao diện người sử dụng cho ứng dụng. đối tượng có thể là một biểu mẫu, một nút lệnh, một hộp văn bản, một nhãn....

Thuộc tính (Properties) :Thuộc tính của đối tượng là những đặc điểm của đối tượng để mô tả đối tượng và phân biệt đối tượng này với đối tượng khác.Ví dụ đối tượng nút lệnh có các thuộc tính như tên, kích thước, vị trí, tiêu đềMỗi đối tượng trong Visual Basic đều có một tập hợp thuộc tính đã quy định trước.

Mỗi đối tượng đều có thể thực hiện một số hành động hoặc cho phép thực hiện một số hành động trên nó. Phương thức của đối tượng là hành động mà đối tượng có thể thực hiện được hoặc hành động được phép thực hiện trên nó. Mỗi đối tượng trong Visual

Basic đều có một tập hợp các phương thức đã quy định trước. Các phương thức của đối tượng được thực hiện nhờ các thủ tục biến cố, các hàm hoặc các modul độc lập.

Chú ý: Trong Visual Basic hai thuật ngữ **đối tượng** (object) và **điều khiển** (control) thường được sử dụng và có thể thay thế cho nhau nhưng điều khiển chỉ là những đối tượng đặt trong biểu mẫu.

3.4.2. Một số thuộc tính thường dùng của các đối tượng

■ Thuộc tính Name (Tên)

Thuộc tính này xác định tên cho đối tượng. Tên này được dùng để truy nhập đối tượng trong phần mã lệnh. Mỗi đối tượng đều có tên ngầm định do Visual Basic đặt cho. Ví dụ: Biểu mẫu có tên ngầm định là Form1, nút lệnh có tên ngầm định là command1....

Tên đối tượng được đặt theo qui tắc sau đây:

+ Tên có độ dài cực đại là 40 kí tự bao gồm chữ cái, chữ số và dấu gạch nối

+ Tên không được trùng với từ khoá

+ Tên phải bắt đầu bằng chữ cái và không có dấu cách. Tên có thể bao gồm cả chữ hoa lẫn chữ thường. Tên đối tượng phải gọi nhớ, ba kí tự đầu tiên của đối tượng được gọi là tiền tố. Visual Basic qui ước cho mỗi đối tượng một tiền tố xác định. Sử dụng tiền tố của tên cho phép dễ dàng nhận ra loại đối tượng khi biết tên đối tượng. Dưới đây là bảng tiền tố của tên các loại đối tượng:

Tiền tố	Loại đối tượng
Frm	Biểu mẫu (Form)
Cmd	Nút lệnh (Command Button)
Lbl	Nhãn (Label)
Txt	Hộp văn bản (Textbox)

Chú ý: Quy định về tiền tố là không bắt buộc nhưng chúng ta nên tuân thủ theo nguyên tắc này.

■ Thuộc tính Caption (Tiêu đề)

Thuộc tính Caption xác định tiêu đề cho đối tượng. Tiêu đề là một dãy kí tự bất kỳ do người sử dụng xác định. Tiêu đề của đối tượng xuất hiện cùng với đối tượng trong biểu mẫu. Tiêu đề của đối tượng cho phép người sử dụng nhận ra đối tượng trong màn hình giao diện của chương trình.

■ Thuộc tính Enabled (Khả ứng)

Thuộc tính này xác định khả năng đáp ứng các biến cố tác động trong thời gian thực hiện chương trình của đối tượng. Thuộc tính này có hai giá trị

+ Nếu là True thì đối tượng có thể đáp ứng các biến cố tác động trong thời gian thực hiện chương trình.

+ Nếu là False thì đối tượng không có khả năng đó.

■ Thuộc tính Font (Font chữ)

Thuộc tính này dùng để xác định Font cho đối tượng. Giá trị của thuộc tính này là tên các Font chữ đang được sử dụng. Ta có thể gán cho thuộc tính này tên của Font chữ bất kỳ hiện có trong máy tính với một kích thước xác định.

■ Thuộc tính BackColor (màu nền)

Xác định màu nền của đối tượng. Thuộc tính này có thể nhận màu bất kỳ trong hệ thống màu đang được sử dụng.

■ Thuộc tính ForeColor (Màu chữ)

Xác định màu chữ của đối tượng. Thuộc tính này có thể nhận màu bất kỳ trong hệ thống màu đang được sử dụng.

■ Thuộc tính Visible (Thấy được)

Thuộc tính này xác định tính thấy của đối tượng. Thuộc tính này có hai giá trị:

+ Nếu là giá trị True thì đối tượng có thể nhìn thấy được trong thời gian thực hiện chương trình

+ Nếu là giá trị False thì đối tượng bị ẩn trong thời gian thực hiện chương trình

■ Truy cập đến giá trị của điều khiển

Cú pháp chung để truy nhập đến giá trị của các điều khiển từ dòng lệnh như sau:

<tên điều khiển>.<tên thuộc tính>

Ví dụ: txtHovaten.Text = “Nguyễn Văn A”

optNam.Value = True

st = txtHovaten.Text

Tuy nhiên đối với các thuộc tính giá trị được sử dụng thường xuyên nên Visual Basic cho phép chúng ta dùng tên của điều khiển để truy nhập đến thuộc tính giá trị.

Ví dụ: Thay vì viết txtHovaten.Text = “Nguyễn Văn A” ta có thể viết txtHovaten = “Nguyễn Văn A”

Ngoài các thuộc tính chung của các điều khiển như đã phân nhóm ở trên một thuộc tính cũng rất hay được sử dụng đó là thuộc tính **TabIndex**. Thuộc tính này thiết lập thứ tự **quan tâm** của các điều khiển khi người sử dụng ấn phím TAB trên bàn phím.

3.5. Một số đối tượng cơ bản trong Visual Basic

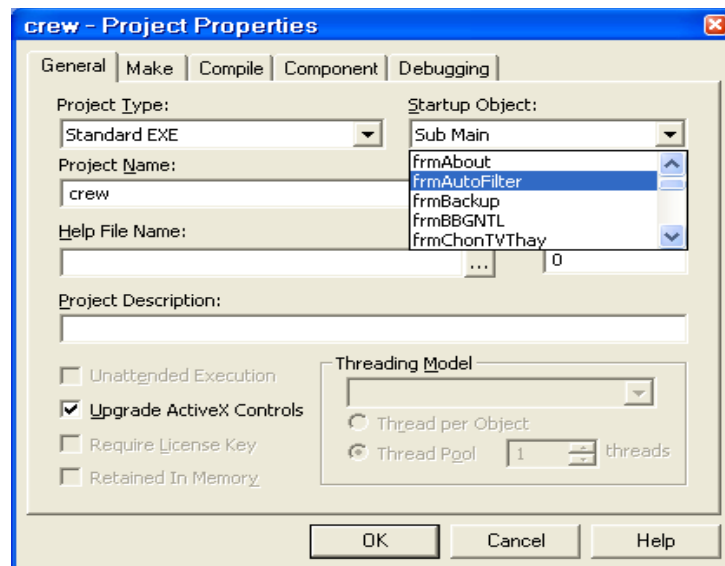
3.5.1. Biểu mẫu (Form)

Form còn gọi là biểu mẫu, là thành phần quan trọng nhất trong số tất cả các thành phần giao diện. Tất cả các thành phần giao diện người dùng ứng dụng đều được đặt trong Form. Khi ứng dụng chạy, nó mở ra một form. Sau khi Form được nạp vào bộ nhớ, tất cả các thành phần hàm chứa trong nó sẽ hiển thị và có thể truy nhập. Chúng ta có thể xem Form như một điều khiển đặc biệt.

Form trong Visual Basic được chia thành 02 loại: Form thường và Form MDI (Multi Document InterFace).

■ Xác định biểu mẫu khởi đầu cho dự án

Trong một chương trình thông thường sẽ có nhiều Form, câu hỏi đặt ra là chương trình khi chạy sẽ bắt đầu từ form nào? Tức là Form nào sẽ xuất hiện ra màn hình trước tiên khi chương trình chạy. Để thực hiện việc xác định form khởi đầu của chương trình chúng ta sẽ thực hiện như sau: Trong menu Project chọn mục <tên Project> Properties (tên Project chính là thuộc tính name của Project mà chúng ta đã đặt), khi đó màn hình (hình 16) sẽ xuất hiện. Trong màn hình này chọn thẻ General trong danh sách thả xuống (combo box) Start up Object chọn Form chúng ta muốn nó xuất hiện đầu tiên khi chương trình chạy.



■ Xóa một biểu mẫu

Trong quá trình thiết kế chương trình vì một lý do nào đấy mà ta muốn xóa một biểu mẫu, ta tiến hành như sau:

- Chọn Project
- Chọn Remove tên biểu mẫu

■ Các thuộc tính của biểu mẫu

Biểu mẫu có đến 50 thuộc tính. Những thuộc tính cơ bản nhất và hay sử dụng được cho trong bảng dưới đây:

Thuộc tính	Ý nghĩa, tác dụng
Appearance	Xác định khi nào biểu mẫu có dạng ba chiều
BackColor	Xác định màu nền của biểu mẫu
BorderStyle	Xác định kiểu đường viền xung quanh biểu mẫu
Caption	Xác định tiêu đề cho biểu mẫu
Enabled	Xác định tính khả ứng của đối tượng cho trong biểu mẫu
Font	Xác định Font chữ cho biểu mẫu
ForeColor	Xác định màu chữ của biểu mẫu
Height	Xác định chiều cao của biểu mẫu
Left	Xác định khoảng cách từ biên trái của màn hình đến biên trái của biểu mẫu
MDIChild	Xác định biểu mẫu này có thuộc kiểu cha con không. Nếu thuộc tính này là True thì biểu mẫu này là biểu mẫu MDI, tức là biểu mẫu con trong biểu mẫu cha. Nếu thuộc tính này là False (ngầm định) thì biểu mẫu này không phải là biểu mẫu MDI
Name	Xác định tên của biểu mẫu. Tên ngầm định của biểu mẫu là Form1
Top	Xác định khoảng cách từ biên trên của màn hình đến biên trên của biểu mẫu
Visible	Xác định tính thấy được của biểu mẫu
Width	Xác định chiều rộng của biểu mẫu

3.5.2. Nhãn (Label **A**)

■ Khái niệm nhãn

Mục đích của điều khiển này là hiển thị một đoạn văn bản lên form thông qua thuộc tính Caption. Label khác textbox ở chỗ nó không cho phép người sử dụng nhập một đoạn văn bản vào như là textbox. Thông thường Label bao giờ cũng được sử dụng kết hợp với textbox với mục đích chú thích cho từng mục dữ liệu nằm trên Form.

■ Các thuộc tính cơ bản của nhãn

Nhãn có đến 30 thuộc tính. Những thuộc tính cơ bản nhất và hay sử dụng được cho trong bảng dưới đây:

Thuộc tính	Ý nghĩa, tác dụng
------------	-------------------

Alignment	Xác định kiểu canh lề cho văn bản có trong nhãn. Thuộc tính này có ba giá trị là: 0 – canh trái, 1 – canh phải, 2 – canh giữa
Appearance	Xác định khi nào nhãn có dạng ba chiều
BackColor	Xác định màu nền của nhãn
BorderStyle	Xác định kiểu đường viền xung quanh nhãn
Caption	Xác định tiêu đề cho nhãn
Enabled	Xác định tính khả ứng của nhãn
Font	Xác định Font chữ cho nhãn
ForeColor	Xác định màu chữ của nhãn
Height	Xác định chiều cao của nhãn
Left	Xác định khoảng cách trái của nhãn
Name	Xác định tên nhãn. Tên ngầm định của biểu mẫu là Label1
Top	Xác định khoảng cách trên nhãn
Visible	Xác định tính thấy được của nhãn
Width	Xác định chiều rộng của nhãn
AutoSize	Thuộc tính này có hai giá trị là TRUE và FALSE, nếu thuộc tính này bằng TRUE thì Visual Basic sẽ tự điều chỉnh kích thước của nhãn sao cho vừa đủ để hiển thị giá trị của thuộc tính Caption

3.5.3. Hộp văn bản (Textb^{labl})

■ Khái niệm hộp văn bản

TextBox được sử dụng để nhập một đoạn văn bản hay hiển thị một đoạn văn bản. Đây là một điều khiển được sử dụng nhiều nhất trong các chương trình viết bằng Visual Basic.

■ Các thuộc tính cơ bản của hộp văn

Nhãn có đến 30 thuộc tính. Những thuộc tính cơ bản nhất và hay sử dụng được cho trong bảng dưới đây:

Thuộc tính	Ý nghĩa, tác dụng
Alignment	Xác định kiểu canh lề cho văn bản có trong hộp văn bản. Thuộc tính này có ba giá trị là: 0 – canh trái, 1 – canh phải, 2 – canh giữa
Appearance	Xác định khi nào hộp văn bản có dạng ba chiều
BackColor	Xác định màu nền của hộp văn bản
BorderStyle	Xác định kiểu đường viền xung quanh hộp văn bản

Enabled	Xác định tính khả ứng của hộp văn bản
Font	Xác định Font chữ cho hộp văn bản
ForeColor	Xác định màu chữ của hộp văn bản
Height	Xác định chiều cao của hộp văn bản
Left	Xác định khoảng cách trái của hộp văn bản
Name	Xác định tên hộp văn bản. Tên ngầm định của biểu mẫu là Text1
Top	Xác định khoảng cách trên của hộp văn bản
Visible	Xác định tính thấy được của hộp văn bản
Width	Xác định chiều rộng của hộp văn bản
MaxLength	Thuộc tính này là giá trị số nguyên có giá trị lớn hơn hoặc bằng 0, nó quy định chiều dài tối đa của đoạn văn bản nhập vào trong textbox. Nếu thuộc tính này đặt bằng 0 (giá trị mặc định) thì sẽ không giới hạn chiều dài tối đa.
PasswordChar	Thuộc tính này thiết lập ký tự đóng vai trò mặt nạ nhập liệu. Thông thường thuộc tính này thường được sử dụng trong trường hợp yêu cầu người sử dụng nhập vào Password. Khi người sử dụng nhập các giá trị vào trong textbox, mỗi ký tự nhập vào sẽ được hiển thị thành ký tự đã thiết lập là mặt nạ tuy nhiên giá trị của thuộc tính Text vẫn giữ nguyên giá trị mà người sử dụng nhập vào.
MultiLine	Thuộc tính này có 2 giá trị True và False nếu bằng True Visual Basic sẽ tự động xuống dòng. Ta thường thiết lập thuộc tính này bằng True trong trường hợp nhập các thông tin như: nội dung một đoạn văn bản, các ghi chú, quá trình công tác của một nhân viên...
Text	Thuộc tính này chứa nội dung của TextBox. Thông thường ta có thể nhập tối đa 2048 ký tự vào TextBox. Nếu thiết lập thuộc tính MultiLine = TRUE ta có thể nhập vào 32K ký tự.
ScrollBars	Thuộc tính này được sử dụng để thiết lập chế độ cuộn văn bản trong textbox khi thuộc tính Multiline đặt bằng True. Các giá trị của thuộc tính như sau: <ul style="list-style-type: none">+ 0-None: Không có thanh cuộn+ 1- Horizontal: Chỉ có thanh cuộn ngang+ 2-Vertical: Chỉ có thanh cuộn dọc

	+ 3- Both: Cả thanh cuộn ngang và thanh cuộn dọc
Locked	Thuộc tính này có hai giá trị True, False nếu Locked = True lệnh Copy vẫn hoạt động trong khi lệnh Cut và Paste không hoạt động với TextBox

Thủ tục tình huống đáp ứng sự kiện

- Sự kiện Change: Sự kiện này xảy ra mỗi khi thuộc tính Text của textbox bị thay đổi
- Sự kiện LostFocus: Sự kiện này xảy ra mỗi khi Textbox mất sự quan tâm.
- Sự kiện GotFocus: Sự kiện này xảy ra mỗi khi Textbox nhận được sự quan tâm.
- Sự kiện KeyPress: Sự kiện này xảy ra mỗi khi người sử dụng ấn một phím bất kỳ khi textbox đang nhận được sự quan tâm.. Cấu trúc thủ tục tình huống như sau:

```
Private Sub <tên điều khiển>_KeyPress(KeyAscii As Integer)
```

 ‘ Tập hợp các lệnh

```
End Sub
```

Thủ tục tình huống này truyền vào một tham trị có kiểu là số nguyên KeyAscii là mã Ascii của phím được bấm.

3.5.4. Nút lệnh (Command Button 

Khái niệm: CommandButton là một nút lệnh mà ta có thể nhấn chuột vào, khi đó nó sẽ thực hiện một hành động nào đó kéo theo.

Các thuộc tính của nút lệnh

- Caption: Đây là thuộc tính thể hiện nhãn của nút lệnh. Ví dụ như: Thêm, Ghi, Xoá, Sửa, Đóng... Bình thường để nút lệnh thực hiện công việc của nó người sử dụng có thể nhấn chuột vào nút lệnh hoặc ấn phím Enter khi nút lệnh đang được chọn, nếu ta muốn gián cho nó một tổ hợp phím nóng ALT + <phím ký tự> thì đằng trước phím ký tự đó ta đặt vào dấu “&”.

Ví dụ: Tổ hợp phím ALT + T để chọn nút lệnh Thêm thì thuộc tính Caption của nút lệnh ta sẽ gán cho nó giá trị sau đây: &Ghi. Khi đó nút lệnh sẽ thể hiện như sau:



- Style: Thiết lập nút lệnh có cho phép hiển thị hình ảnh hay không. Thuộc tính này có hai giá trị:

+ 0-Standard: Nút lệnh bình thường, chỉ có nhãn mà không có hình ảnh đồ họa hiển thị bên trong.

+ 1-Graphical: Cho phép vừa có nhãn vừa có hình ảnh đồ họa hiển thị trong nút lệnh. Hình ảnh đồ họa hiển thị trong nút lệnh do thuộc tính Picture thiết lập.

- Picture: Thuộc tính này chỉ định tên file đồ họa sẽ hiển thị trong nút lệnh. Các dạng file đồ họa là Bitmap, Gif, Jpeg, Metafile, Icon. Thuộc tính này chỉ có tác dụng khi thuộc tính Style đã đặt bằng 1.

- Enabled: Thuộc tính này có hai giá trị True và False. Người lập trình có thể linh hoạt thiết lập giá trị cho thuộc tính này sao cho phù hợp với tiến trình hoạt động của chương trình.

■ Thủ tục tình huống đáp ứng sự kiện

Thủ tục tình huống đáp ứng sự kiện quan trọng nhất của nút lệnh đó là sự kiện người sử dụng click chuột vào nút lệnh. Thủ tục tình huống có khuôn mẫu như sau:

```
Private Sub <tên nút lệnh>_Click()
```

```
    ' Các lệnh sẽ được viết ở đây
```

```
End Sub
```

3.6. Một số qui ước trong ngôn ngữ

* Bộ kí tự

Visual Basic được xây dựng với bộ kí tự sau:

- Các chữ cái: 26 chữ cái hoa (A,B,.....,Z)

26 chữ thường (a,b,.....z)

- Các chữ số thập phân: 0,1,2,3,4,5,6,7,8,9

- Các dấu toán học thông dụng: + - * / > >= < <= = ()....

- Dấu gạch dưới _

- Các kí hiệu đặc biệt: . : & \$ # ^ ‘ “ ?

* Từ khoá (Keyword)

Từ khoá là các từ dành riêng cho Visual Basic, mỗi từ có một chức năng nhất định. Khi sử dụng phải dùng đúng với chức năng.

Một số từ khoá:

DIM	END	SUB	FUNCTION	PRIVATE
CONST	IF	THEN	FOR	WHILE
STATIC	SELECT	CASE	TYPE	PUBLIC
WEND	NEXT			

*Tên (Identifier)

Tên là một dãy kí tự được tạo thành từ các chữ cái, chữ số và dấu gạch nối. Tên thường dùng để đặt tên cho các đại lượng trong chương trình như tên biến, hằng, tên kiểu dữ liệu mới vv....

Kí tự đầu tiên của tên không được là chữ số. Tên không được trùng với từ khoá.

- Tên chuẩn: là tên do Visual Basic định nghĩa trước dùng chỉ tên các hàm, hằng, biến, thủ tục, hàm, sự kiện trong Visual Basic.

- Tên do người lập trình đặt: Đặt tùy ý nhưng phải tuân thủ theo nguyên tắc của Visual Basic như tên biến, tên kiểu dữ liệu mới, tên thủ tục, tên hàm.....

3.7. Các kiểu dữ liệu cơ bản

3.7.1. Kiểu logic (Boolean)

- Một dữ liệu kiểu logic chỉ nhận một trong hai giá trị sau: TRUE (đúng) hoặc FALSE (sai). Dữ liệu kiểu logic chiếm mất 2 byte trong bộ nhớ

- Quy ước: FALS < TRUE.

- Các phép toán đối với kiểu Logic: AND, OR, XOR, NOT.

3.7.2. Kiểu số nguyên

- Có ba kiểu số nguyên được tổng kết qua bảng sau:

Tên	Miền giá trị	Yêu cầu bộ nhớ
Byte	0 đến 255	1 byte
Integer	-32768 đến 32767	2 byte
Long	- 2147483648 đến 2147483647	4 byte

- Các phép tính với kiểu số nguyên:

Phép tính	Ý nghĩa	Phép tính	Ý nghĩa
+	cộng	/	chia
-	trừ	\	chia lấy phần nguyên
*	nhân	MOD	chia lấy số dư của hai số nguyên
^	phép lũy thừa		

Ví dụ: $5/2=2.50$

$5 \setminus 2=2$

$5 \text{ MOD } 2=1$

Chú ý:

- Khi viết các số nguyên phải tuân theo các quy định: không có khoảng trống giữa các số, dấu + hay - (nếu có) phải đặt ngay trước chữ số đầu tiên, không được sử dụng dấu chấm thập phân.

- Khi thực hiện các phép tính trên số nguyên, phải hết sức thận trọng xem các phép toán đó có cho kết quả vượt quá phạm vi biểu diễn của số nguyên hay không. Ví dụ a, b là dữ liệu kiểu Integer. Khi đó hai câu lệnh sau:

a=80;

b=(a*1245) \ 200;

cho kết quả sai vì $a*1245 = 99600$ vượt quá phạm vi của Integer.

3.7.3. Kiểu số thực

- Có bốn kiểu số thực, được tổng kết qua bảng sau:

Tên kiểu	Miền giá trị	Yêu cầu bộ nhớ
Single (Số thực) Có độ chính xác đơn	-3.402823E38 đến -1.401298E-45 1.401298E-45 đến 3.402823E+38	4 byte
Double (Số thực) Có độ chính xác kép	-1.79769313486231 E308 đến - 4.94065645841247E-34 4.94065645841247E-324 đến 1.79769313486232E308	8 byte
Currency (15 số nguyên và 4 số thập phân)	-922,337,203,685,477.5808 đến 922,337,203,685,477.5807	8 byte
Decimal	Có thể lưu số nguyên hay số thực	14 byte

Các giá trị thực được biểu diễn theo hai cách: dạng dấu phẩy tĩnh và dạng dấu phẩy động.

- Cách viết số thực theo dạng dấu phẩy tĩnh: viết dạng thập phân bình thường.

Ví dụ: 2.25, 5.0, -25.56812, 0.0256

- Cách viết số thực theo dạng dấu phẩy động: số được tách thành hai phần là định trị và bậc. Phần định trị là một số nguyên hay số thực viết dưới dạng dấu phẩy tĩnh. Phần bậc là một số nguyên. Hai phần cách nhau bởi chữ E hay e.

Ví dụ:

$123.456E-4 = 123.456 * 10^{-4} = 0.0123456$

$0.12E+5 = 0.12 * 10^5 = 12000.0$

$$-52.4E2 = -52.4 \cdot 10^2 = -5240.0$$

Các ký tự biểu diễn một số thực phải viết liền nhau.

- Các phép toán đối với số thực: + (cộng), - (trừ), * (nhân), / (chia).....
Không tồn tại các phép toán \, MOD cho kiểu số thực.

Các hàm số học sử dụng đối với dữ liệu kiểu số

Tên hàm	Cú pháp	Tác dụng
Abs	Abs(n)	Tìm giá trị tuyệt đối của một số
Atn	Atn(number) Ví dụ: Dim pi pi = 4 * Atn(1)	Tìm artangen
Cos	Cos(number) number đo bằng Radian	Tìm cosine
Exp	Exp(x)	Tìm giá trị hàm mũ e ^x (e=2.17..)
Int	Int(number) Ví dụ: Int(99.8)=99 Int(-99.8)=100	Trả về số nguyên lớn nhất trong những số nguyên nhỏ hơn số đó
Fix	Fix(number) Ví dụ: Fix(99.8)=99	Trả về phần nguyên của 1 số
Log10		Logarit cơ số 10
Rnd	Giá trị phát sinh ngẫu nhiên từ 0 đến nhỏ hơn 1	Gọi bộ tạo số ngẫu nhiên
Round	Round(number)	Làm tròn số
Sin	Sin(number) number đo bằng Radian	Tìm sin của 1 số
Sqr	Sqr(number)	Tính căn bậc 2 của 1 số

3.7.4. Kiểu xâu ký tự (String)

Một giá trị kiểu String là một dãy ký tự bất kỳ đặt trong 2 dấu nháy kép. Số ký tự của xâu phụ thuộc vào cách khai báo:

Khai báo xâu có chiều dài không cố định:

Dim St as String

Thì độ dài cực đại của xâu là 2 tỷ ký tự.

Khai báo xâu có chiều dài cố định:

Dim St as String *25

Nếu ta gán cho St một chuỗi ngắn hơn 25 ký tự thì Visual Basic sẽ thêm vào khoảng đuôi của St các ký tự khoảng trắng cho đầy 25 ký tự. Nếu ta gán cho St một chuỗi dài hơn 25 ký tự thì Visual Basic sẽ tự động cắt bỏ phần thừa.

Xâu không có ký tự nào gọi là xâu rỗng.

- Để biểu diễn 1 ký tự có thể dùng 1 trong 2 cách: Đặt ký tự trong 2 dấu nháy kép, dùng hàm Chr(n), dùng #n(trong đó n là mã ASCII của ký tự cần biểu diễn).

Ví dụ: Để biểu diễn ký tự A ta có thể dùng 2 cách: “A” hoặc Chr(65).

- Kiểu xâu ký tự chỉ có phép toán so sánh, sự so sánh dựa vào mã của từng ký tự trong xâu..

Các phép toán đối với xâu ký tự

Phép toán phổ biến nhất được thực hiện với các xâu ký tự đó là ghép hai xâu với nhau. Dấu & dùng để nối hai chuỗi

Ví dụ:

Ho = “Nguyễn Văn “

Ten = “Trung”

Hovaten = Ho & Ten

Dấu & nối các xâu theo thứ tự chúng ta trình bày chúng. Như vậy khác với phép cộng các số, thứ tự là quan trọng khi sử dụng dấu & để nối các xâu với nhau.

Trong nhiều ngôn ngữ lập trình, kể cả các phiên bản trước của Visual Basic có sử dụng dấu + để nối các xâu với nhau.

Một số hàm cơ bản được dùng cho dữ liệu kiểu xâu ký tự

Tên hàm	Cú pháp	Tác dụng
Space	Space(n) n = Số ký tự trắng lặp lại	Trả về xâu ký tự chỉ chứa n ký tự trắng
String	String(n,st)	Lặp lại xâu ký tự st n lần
Ltrim	LTrim(st)	Trả về xâu ký tự sau khi đã loại bỏ các ký tự trắng (nếu có) ở đầu xâu ký tự st
Rtrim	RTrim(st)	Trả về xâu ký tự sau khi đã loại bỏ các ký tự trắng (nếu có) ở cuối xâu ký tự st

Tên hàm	Cú pháp	Tác dụng
Trim	Trim(st)	Trả về chuỗi ký tự sau khi đã loại bỏ các ký tự trắng (nếu có) ở hai đầu chuỗi ký tự st
Len	Len(st)	Trả về độ dài của chuỗi ký tự st
Mid	Mid(st,i,n)	Trích ra trong chuỗi ký tự st n ký tự bắt đầu từ vị trí thứ i
Left	Left(st,n)	Trích ra n ký tự bên phía tay trái chuỗi ký tự st
Right	Right(st,n)	Trích ra n ký tự bên phía tay phải chuỗi ký tự st
InStr	Instr(i, st2,st1[,vbTextCompare]) (không phân biệt chữ hoa, thường)	Kiểm tra vị trí của một chuỗi ký tự trong một chuỗi ký tự khác Tìm chuỗi ký tự st1 trong chuỗi ký tự st2 bắt đầu từ vị trí thứ i. Nếu không tìm thấy hàm trả về giá trị = 0
Val	Val(st)	Trả về giá trị số trong chuỗi ký tự biểu diễn các chữ số st
Ucase	UCase(st)	Trả về chuỗi ký tự in hoa
Lcase	LCase(st)	Trả về chuỗi ký tự chữ thường
Asc	Asc(st)	Trả về mã ASCII của ký tự đầu tiên trong chuỗi ký tự
Chr	Chr(mã ASCII)	Trả về ký tự tương ứng với mã ASCII
Str	Str(số)	Trả về chuỗi ký tự biểu diễn giá trị số

Các hàm chuyển đổi kiểu từ dữ liệu dạng ký tự

Tên hàm	Cú pháp	Tác dụng
Cbool	CBool(biểu thức)	Chuyển sang giá trị kiểu Boolean
Cbyte	CByte(biểu thức)	Chuyển sang giá trị kiểu Byte
Ccur	Ccur(biểu thức)	Chuyển sang giá trị kiểu currency
Cdate	Cdate(biểu thức)	Chuyển sang giá trị kiểu date
CDbl	CDbl(biểu thức)	Chuyển sang giá trị kiểu Double
Cdec	Cdec(biểu thức)	Chuyển sang giá trị kiểu Decimal
Cint	Cint(biểu thức)	Chuyển sang giá trị kiểu Int
CLng	CLng(biểu thức)	Chuyển sang giá trị kiểu Long
CSng	CSng(biểu thức)	Chuyển sang giá trị kiểu Single

Tên hàm	Cú pháp	Tác dụng
CStr	CStr(biểu thức)	Chuyển sang giá trị kiểu String
Cvar	Cvar(biểu thức)	Chuyển sang giá trị kiểu Varial

CHƯƠNG 4

KHAI BÁO HẰNG, BIẾN, BIỂU THỨC CÂU LỆNH

4.1. Các hằng trong Visual Basic

* **Khái niệm hằng.**

Hằng là đại lượng có giá trị xác định và không thay đổi khi chương trình thực hiện. Giá trị của hằng được nhận diện thông qua tên của hằng. Trong Visual Basic có một số hằng như: xâu ký tự, hằng số, hằng có dữ liệu kiểu logic, kiểu thời gian...

* **Khai báo hằng.**

Cú pháp khai báo như sau:

Const <tên hằng> = <giá trị>

Ví dụ: Const UserName = "Nguyễn Văn A"

Const MaxUser As Integer = 10

Giá trị cũng có thể là các biểu thức số cho hằng hoặc khai báo các hằng căn cứ theo các hằng đã được định nghĩa trước. Bản thân Visual Basic đã cung cấp cho chúng ta vô số các hằng hữu dụng để làm việc với các hàm, các đối tượng và phương thức xây dựng sẵn. Nói chung các hằng này bắt đầu với "Visual Basic", các hằng cơ sở dữ liệu bắt đầu với "db",...

Ví dụ: Giả sử gọi mã của phím Enter trên bàn phím thay vì sử dụng mã ASCII của phím đó là 13 thì Visual Basic cung cấp cho ta một hằng đại diện cho mã ấy là hằng có tên là *vbKeyReturn*

4.2. Biến

* **Khái niệm**

Biến là một đại lượng mà giá trị của nó thay đổi khi thực hiện chương trình. Biến là tên của một vùng bộ nhớ lưu trữ dữ liệu, mỗi biến phải thuộc về một kiểu dữ liệu nhất định và được đặt tên.

* **Khai báo biến.**

Cú pháp khai báo biến:

Dim <biến> [As <kiểu dữ liệu>]

Ví dụ:

Dim Str As String

Dim i As Integer

Chúng ta có thể không cần định nghĩa kiểu dữ liệu của biến khi khai báo, lúc đó biến sẽ có kiểu là Variant.

Có thể khai báo nhiều biến trên một câu lệnh, tuy nhiên khi đó ta phải xác định loại dữ liệu cho mỗi biến

4.3. Biểu thức

Một biểu thức được tạo bởi các toán hạng và các toán tử để thực hiện một công thức toán học. Toán hạng có thể là hằng, hàm, biến.

Chú ý rằng một hằng, một biến, một hàm cũng được xem là một biểu thức, biểu thức đơn giản.

4.4. Quy định về viết dòng lệnh trong Visual Basic.

Khi viết các dòng lệnh Visual Basic phải tuân thủ các nguyên tắc sau đây:

- Thường một câu lệnh được viết trên một dòng của chương trình, một dòng của chương trình chứa tối đa 1023 kí tự.

- Nếu một dòng của chương trình chứa nhiều câu lệnh thì giữa các câu lệnh phải được ngăn cách với nhau bằng dấu : (hai chấm)

- Trong trường hợp một câu lệnh được viết trên nhiều dòng thì khi đó ở cuối dòng phải có một kí tự dấu cách và dấu gạch dưới đi liền nhau.

4.5. Cách viết dòng chú thích trong chương trình.

Trong quá trình lập trình để làm sáng tỏ một dòng lệnh hoặc một đoạn chương trình người ta thường đưa dòng chú thích vào trong chương trình. Dòng chú thích được viết theo qui tắc như sau:

Cách 1: ' dòng chú thích'

Cách 2: REM dòng chú thích

Ví dụ: Chú thích cho đoạn mã lệnh khai báo biến i là kiểu số nguyên

Cách 1: 'Khai báo biến i kiểu số nguyên'

Dim i as Integer

Cách2: REM Khai báo biến i kiểu số nguyên

Dim i as Integer

4.6. Câu lệnh gán.

Cú pháp:

Tên_biến= biểu thức

Hoặc

Tên_đôi_tượng.Thuộc_tính= biểu thức

Hoạt động: khi gặp lệnh này máy sẽ tính giá trị của biểu thức bên phải của lệnh và gán giá trị đó cho biến hoặc thuộc tính ở về trái .

Ví dụ: $s = s * b$

4.7. Câu lệnh END.

Khi Visual Basic gặp câu lệnh END, máy sẽ dừng lại. Nếu ta đang trong môi trường phát triển ứng dụng thì sẽ quay trở về môi trường phát triển. Trong môi trường độc lập, khi gặp câu lệnh End tất cả cửa sổ được mở ra sẽ bị đóng lại và chương trình bị xoá khỏi bộ nhớ. Ta có thể đặt câu lệnh END ở nhiều nơi trong chương trình tuy nhiên phong cách lập trình tốt là hạn chế tối đa câu lệnh này. Những lập trình viên chuyên nghiệp thích dùng chỉ 1 câu lệnh End, và thường đặt trong tình huống Query_UnLoad của biểu mẫu chính.

4.8. Lệnh in dữ liệu

4.8.1. Câu lệnh PRINT.

Câu lệnh này nhằm mục đích in giá trị của biểu thức lên trên một Form.

Cú pháp như sau:

Print <Biểu thức>[, <Biểu thức 2>].....

Gặp câu lệnh này, máy sẽ tính giá trị các biểu thức và in các giá trị đó trên biểu mẫu. Nếu hai biểu thức cách nhau dấu phẩy, các giá trị được in cách nhau một dấu cách. Nếu hai biểu thức cách nhau bởi dấu chấm phẩy (;) thì giá trị của biểu thức sau sẽ được in ở vùng tiếp theo. Một vùng gồm 14 kí tự đi liền nhau trên một biểu mẫu.

Ví dụ: Print “Diện tích tam giác là s=”,str(s)

Chú ý: Khi soạn thảo chương trình có thể gõ dấu hỏi “?” thay cho gõ từ “Print”. Visual Basic sẽ tự thay dấu “?” bằng từ “Print”

Ví dụ: ? “Diện tích tam giác là s=”,str(s)

4.8.2. Lệnh MSGBOX

Lệnh MsgBox tạo ra trên màn hình một cửa sổ nhỏ mà ta gọi là hộp thông báo. Hộp này chứa một thông báo mà chương trình muốn gửi cho người sử dụng và một nút lệnh OK. Khi người sử dụng đã xem xong thông báo thì nháy chuột vào nút lệnh OK, câu thông báo này sẽ biến mất.

Lệnh này có cú pháp như sau:

MsgBox Thông báo, [Kiểu], [Tiêu đề]

Trong đó:

Thông báo: Là một dãy kí tự tạo nên dòng thông báo đến người sử dụng.

Tiêu đề: Là dãy kí tự được dùng làm tiêu đề của hộp thông báo.

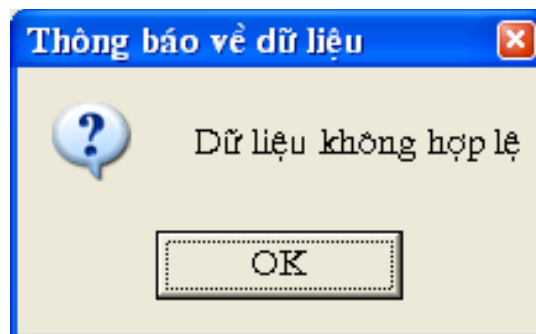
Kiểu: Là một số nguyên xác định biểu tượng dùng để cảnh báo. Tham số kiểu có giá trị được liệt kê trong bảng sau đây:

Giá trị của kiểu	Ý nghĩa
16	Hiện biểu tượng STOP
32	Hiện biểu tượng dấu hỏi (?)
48	Hiện biểu tượng dấu chấm than (!)
64	Hiện biểu tượng chữ i (information)
4096	Sử dụng hệ thống MODAL

Ví dụ: Giả sử ta có câu lệnh sau:

Msgbox "Dữ liệu nhập vào chưa đúng",32,"Thông báo về dữ liệu"

Câu lệnh này xuất hiện lên màn hình hộp thông báo như sau:



4.8.3. Dùng Text box

Cú pháp: TênTextBox.Text=<Biểu thức>

Ví dụ: - In ra TextBox câu “Chào Bạn” (tên textbox là txtkq)

Txtkq.text=”Chào Bạn”

- In ra TextBox giá trị của một biến a

Txtkq.Text=”Giá trị củav a=”+str(a)

4.9. Lệnh nhập dữ liệu

4.9.1. Dùng Textbox

Cú pháp: Biến= TênTextBox.Text

Ví dụ: - Nhập vào tên của sinh viên thông qua biến sv

Sv=Txtsv.Text

- Nhập vào giá trị cho biến a

a=Val(Txta.Text)

4.9.2. Hàm nhập dữ liệu INPUTBOX

Cú pháp: **INPUTBOX(Lời nhắc, tiêu đề, ngầm định)**

Trong đó:

Lời nhắc: Là dãy kí tự được dùng để hướng dẫn người sử dụng nhập dữ liệu

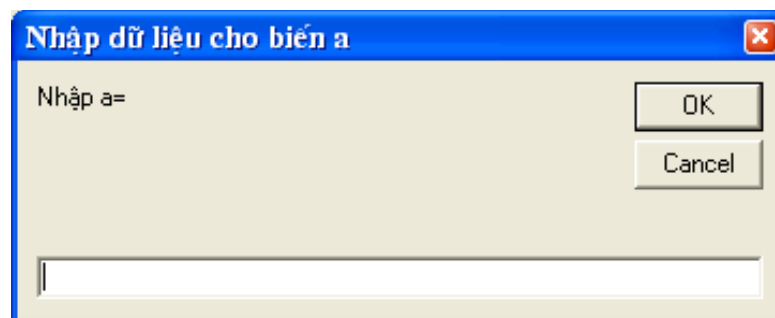
Tiêu đề: Là dãy kí tự được dùng làm tiêu đề của hộp nhập dữ liệu, hộp này xuất hiện mỗi khi hàm nhập dữ liệu được thực hiện.

Ngầm định: Là dãy kí tự được dùng làm giá trị ngầm định của dữ liệu được nhập

Ví dụ: muốn nhập giá trị cho biến a

a= Val(INPUTBOX("Nhập a=", "Nhập dữ liệu cho biến a")

Khi gặp lệnh này máy sẽ thực hiện hàm INPUTBOX và xuất hiện cửa sổ nhập liệu sau:



4.10. Chuyển đổi kiểu dữ liệu trong quá trình tính giá trị biểu thức

Trong quá trình tính giá trị của một biểu thức, các thành phần trong biểu thức có thể là các kiểu dữ liệu khác nhau. Kiểu của kết quả sẽ được xác định theo yêu cầu của người sử dụng. Dưới đây là bảng tóm tắt việc chuyển đổi dữ liệu nêu trên:

Câu lệnh	Tác dụng
Dim i as Integer Dim x as Single	Khai báo biến i là biến kiểu số nguyên Khai báo biến x là biến kiểu số thực.
I=1+"2"	Trong câu lệnh này chuỗi "2" được biến thành số nguyên 2, rồi cộng thêm với 1. Kết quả của phép cộng này là số 3 được gán cho biến I. Như vậy sau khi thực hiện câu lệnh này biến I có giá trị là 3
I="1"+"2"	Trong câu lệnh này, đầu tiên chuỗi "1" và "2" cộng lại với nhau tạo

	thành chuỗi "12". Sau đó chuỗi này được biến đổi thành số 12. Số 12 được gán cho biến I. như vậy sau câu lệnh này biến I có giá trị là 12.
X=5.5	Gán giá trị 5.5. cho biến X
X="5.5"	Trong câu lệnh này, đầu tiên chuỗi "5.5" được biến đổi thành số thực 5.5, sau đó gán cho biến X. Sau câu lệnh này biến x có giá trị là 5.5.
X="5,5"	Trong câu lệnh này, đầu tiên chuỗi "5,5" được biến đổi thành số thực 55, vì Visual Basic bỏ qua dấu “,” không cho đó là dấu phẩy thập phân, dùng để ngăn chia phần nguyên và phần lẻ của số. Sau đó gán cho biến X. Sau câu lệnh này biến X có giá trị là 55.
X="5B5"	Câu lệnh này không đúng vì xâu “5B5” không thể đổi thành số được

4.11. Cấu trúc điều kiện

4.11.1. Câu lệnh điều kiện thiếu

Cú pháp 1:

If <biểu thức điều kiện> Then <công việc>

Cú pháp 2:

If <biểu thức điều kiện> Then

<công việc>

End If

Giải thích

<Biểu thức điều kiện> là giá trị kiểu logic.

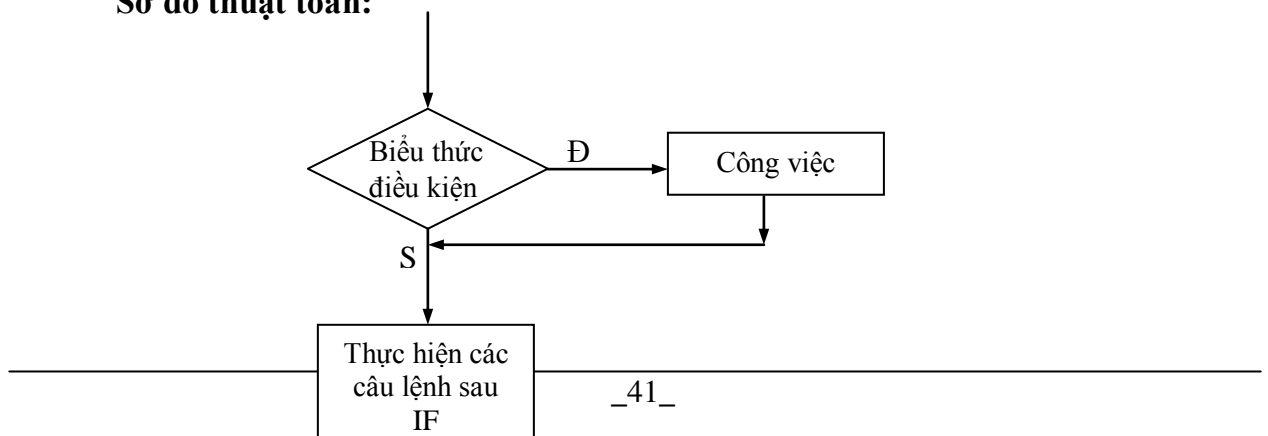
Hoạt động của câu lệnh như sau: Nếu biểu thức điều kiện cho giá trị đúng thì thực hiện <công việc> sau Then, ngược lại <công việc> sẽ không được thực hiện và máy thực hiện tiếp những câu lệnh đứng sau câu lệnh IF.

Cú pháp 1 thường được sử dụng khi <công việc> thường là một câu lệnh đơn.

Ví dụ:

IF Not IsNumeric(Text1.Text) THEN MsgBox “Đây không phải là số”

Sơ đồ thuật toán:



4.11.2. Câu lệnh điều kiện đủ

Cú pháp 1:

If <biểu thức điều kiện> Then

<công việc 1>

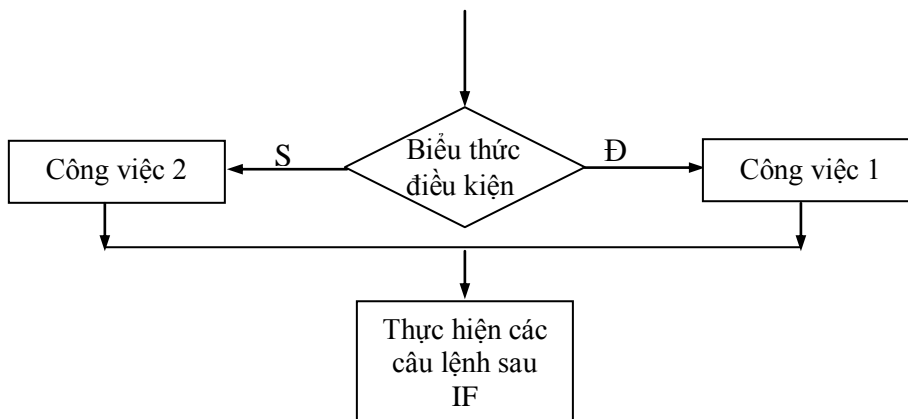
Else

<công việc 2>

End If

Khi gặp câu lệnh này chương sẽ kiểm tra biểu thức điều kiện, nếu điều kiện đúng máy sẽ thực hiện công việc 1 ngược lại sẽ thực hiện công việc 2.

Sơ đồ thuật toán:



Cú pháp 2:

If <Biểu thức điều kiện 1> Then

<Công việc 1>

ElseIf <Biểu thức điều kiện 2> Then

<Công việc 2>

ElseIf <Biểu thức điều kiện n>

<Công việc n>

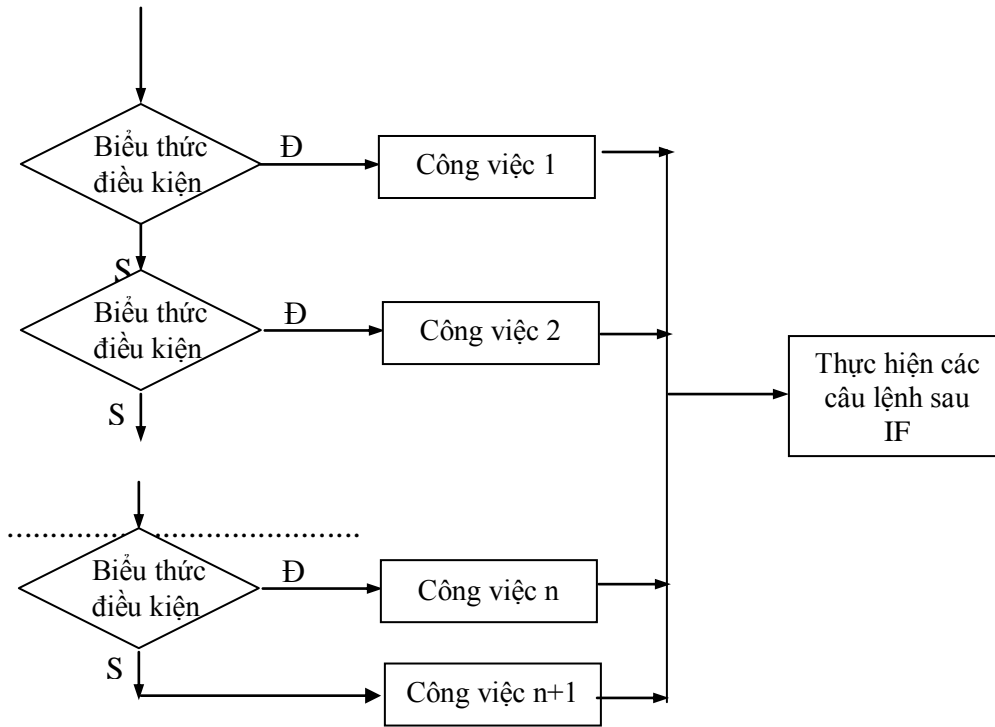
Else

<Công việc n+1>

End If

Khi gặp câu lệnh này máy tính sẽ kiểm tra biểu thức điều kiện 1, nếu điều kiện 1 đúng thì công việc 1 sẽ được thực hiện ngược lại sẽ kiểm tra biểu thức điều kiện 2 nếu biểu thức điều kiện 2 đúng thì công việc 2 sẽ được thực hiện... Với cấu trúc này 1 công việc sẽ được thực hiện khi biểu thức điều kiện tương ứng với nó cho giá trị đúng, trong trường hợp không có biểu thức điều kiện nào nào đúng thì sẽ đi thực hiện công việc thứ n+1.

Sơ đồ thuật toán:



4.12. Câu lệnh lựa chọn SELECT

Câu lệnh này cũng giống như câu lệnh If ... else ... End if. Tuy nhiên cấu trúc này có nhiều sự lựa chọn hơn. Cú pháp của câu lệnh như sau:

```
Select Case <Biểu thức>
    Case <giá trị 1>
        <công việc 1>
    Case <giá trị 2>
        <công việc 2>
    .....
    Case <giá trị n>
        <công việc n>
[ Case Else
```

<công việc n+1>]

End Select

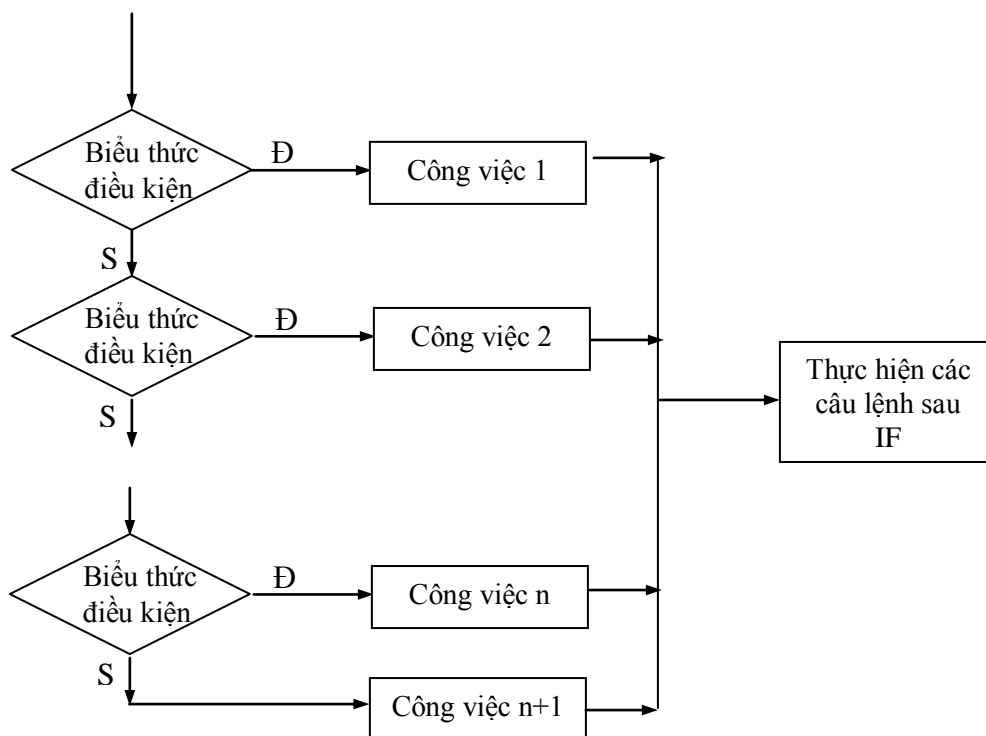
Trong đó, giá trị có thể là 1 giá trị cụ thể hoặc là miền giá trị hoặc một biểu thức

○ <biểu thức 1> To <biểu thức 2>: miền giá trị từ <biểu thức 1> đến <biểu thức 2>

○ Is <toán tử so sánh><Biểu thức> (hay <biểu thức><toán tử so sánh><Biểu thức>)

Hoạt động: Gặp lệnh này máy sẽ đi tính giá trị của biểu thức sau đó đem so sánh với các giá trị. Nếu nằm vào miền giá trị thứ i thì đi thực hiện công việc thứ i, ngược lại không trùng với miền giá trị nào thì đi thực hiện công việc thứ n+1

Sơ đồ thuật toán



Ví dụ : Xác định mùa trong năm dựa vào tháng

Select Case Month

Case Is <4

Msgbox “Mùa Xuân”

Case 4,5,6

Msgbox “Mùa hè”

Case 7 to 9

Msgbox “Mùa Thu”

Case Else

End Select

4.13. Cấu trúc lặp .

4.13.1. Khái niệm về lặp.

Lặp là một hiện tượng, một công việc được thực hiện đi thực lại liên tiếp nhiều lần tại cùng một khoảng thời gian xác định. Trong các ngôn ngữ lập trình người ta chia ra làm hai loại cấu trúc lặp:

- + Vòng lặp xác định: Là loại vòng lặp biết trước số lần lặp
- + Vòng lặp không xác định là loại vòng lặp không biết trước số lần lặp

4.13.2. Vòng lặp xác định.

Vòng lặp xác định là vòng lặp mà số lần lặp lại của công việc được biết trước. Trong Visual Basic vòng lặp xác định được thực hiện bởi câu lệnh For có cấu trúc như sau:

For <biến> = <biểu thức 1> To <biểu thức 2> [Step <biểu thức 3>]

<công việc>

[Exit For]

Next [<biến>]

Trong đó biểu thức 1 và biểu thức 2 là hai chỉ số đầu và cuối của biến, biểu thức 2 có thể có giá trị dương hay âm. Step nhằm để tăng hay giảm biến sau mỗi vòng lặp. Nếu không xác định rõ thì Step sẽ có giá trị là 1.

Nếu sử dụng câu lệnh Exit For thì khi gặp câu lệnh này máy tính sẽ nhảy ra khỏi vòng lặp.

Ví dụ 1: Tính tổng các số từ 1 đến 10

Dim Sum As Integer, i As Integer

Sum = 0

For i = 1 to 10

Sum = Sum + i

Next i

Ví dụ 2: Tính tổng $S=1+1/2+1/3+\dots+1/(n-1)+1/n$ Với n là số nguyên dương

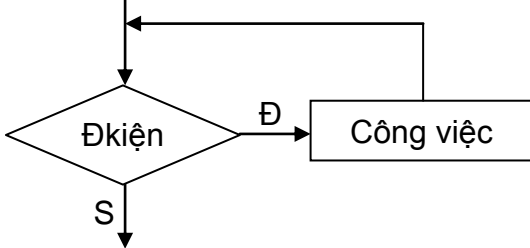
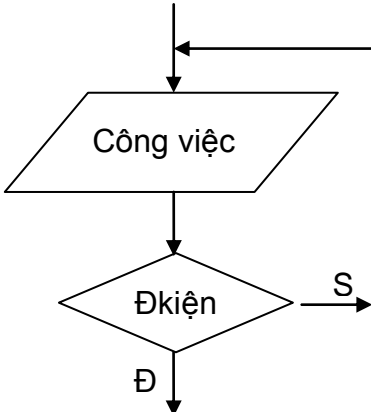
Dim n as integer

Dim i as integer

Dim S as Single

```
n=val(Inputbox("Nhập giá trị cho n="))
S=0
For =1 to n
    S=S+1/i
Next i
Print "Tong s=",str(S)
```

4.13.3. Vòng lặp không xác định

Vòng lặp While	Vòng lặp UnTil
<p>Cú pháp 1:</p> <p style="padding-left: 40px;">Do While <điều kiện lặp> <công việc> [Exit Do]</p> <p style="padding-left: 40px;">Loop</p> <p>Cú pháp 2:</p> <p style="padding-left: 40px;">Do <công việc> [Exit Do]</p> <p style="padding-left: 40px;">Loop While <điều kiện lặp></p> <p>Cú pháp 3:</p> <p style="padding-left: 40px;">While <điều kiện lặp> <công việc></p> <p style="padding-left: 40px;">Wend</p> <p>Hoạt động: Vòng lặp While chỉ hoạt động khi điều kiện đúng còn thoát khi điều kiện sai.</p> <p>Sơ đồ thuật toán cú pháp 1</p> 	<p>Cú pháp 1:</p> <p style="padding-left: 40px;">Do UnTil <điều kiện lặp> <công việc> [Exit Do]</p> <p style="padding-left: 40px;">Loop</p> <p>Cú pháp 2:</p> <p style="padding-left: 40px;">Do <công việc> [Exit Do]</p> <p style="padding-left: 40px;">Loop Until <điều kiện lặp></p> <p>Hoạt động: Vòng lặp UnTil chỉ hoạt động khi điều kiện sai còn thoát khi điều kiện đúng.</p> <p>Sơ đồ thuật toán cú pháp 1</p> 

Ví dụ 1: Tính tổng các số từ 1 đến 10

‘Dùng lệnh While

Dim Sum As Integer, i As Integer

i=0

Sum=0

Do While i<10

 Sum = Sum + i

 i = i + 1

Loop

‘Dùng lệnh Until

Dim Sum as Integer, i as Integer

i=0

Sum = 0

Do Until i>10

 Sum = Sum + i

 i = i+1

Loop

Ví dụ 2: Tìm ước số chung lớn nhất của hai số nguyên dương a,b

Dim a as integer, b as integer

Dim UCLN as Integer

Do

 a= val(Inputbox(“Nhập giá trị cho a=”))

 b= val(Inputbox(“Nhập giá trị cho b=”))

Loop Until a>0 and b>0

Do while a<>b

 If a>b then

 a=a-b

 Else

 b=b-a

 end if

Loop

UCLN=a

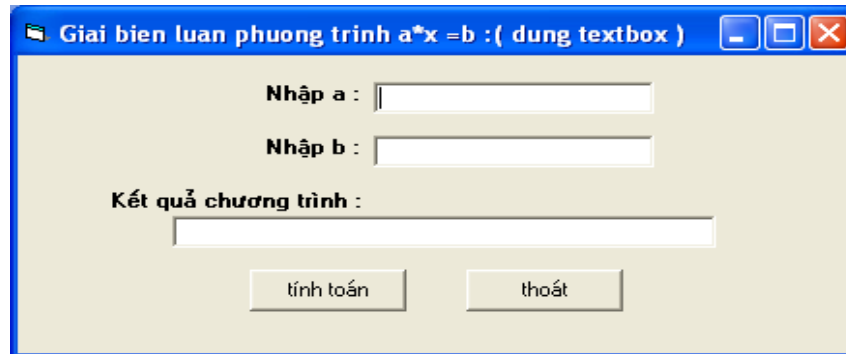
Print “ước số chung lớn nhất là:”,str(UCLN)

4.14. Một số ví dụ:

Ví dụ 1: Giải và biện luận phương trình $ax+b=0$ (Dùng Textbox để nhập liệu)

Cách 1: Dùng textbox để nhập liệu

1. Thiết kế giao diện



2. Bảng thuộc tính

STT	Tên đối tượng	Thuộc tính	Giá trị
1	biểu mẫu	Name Caption	FrmBai1_a Giai bien luan phuong trinh a*x =b :(dung textbox)
2	Textbox	Name Text	Txtnhapa
3	Textbox	Name Text	Txtnhapb
4	Textbox	Name Text	Txtkq
5	Nút lệnh	Name Caption	cmdTinhToan tính toán
6	Nút lệnh	Name Caption	cmdThoat thoát

3. Code lệnh

```
Private Sub CmdTinhToan_Click()
```

```

Dim x As Single
Dim a As Single
Dim b As Single
If (Not IsNumeric(Txtnhapa.Text)) Or (Not IsNumeric(Txtnhapb.Text)) Then
    MsgBox "ban phai nhap a, b la chu so"
    Exit Sub
End If
a = Val(Txtnhapa.Text)
b = Val(Txtnhapb.Text)
If a = 0 Then
    MsgBox "So a khong duoc la 0"
    Exit Sub
End If
x = -b / a
Ttxtkq.Text = a & "*x + " & b & " = " & x
End Sub

Private Sub CmdThoat_Click()
Unload Me
End Sub

```

Cách 2: Dùng hàm Inputbox để nhập liệu

1. Giao diện bài toán



2. Bảng thuộc tính

STT	Tên đối tượng	Thuộc tính	Giá trị
1	biểu mẫu	Name Caption	FrmBai1_b Giai bien luan phuong trinh a*x =b :(khong dung textbox)

2	Textbox	Name Text	TxtKq
3	Nút lệnh	Name Caption	cmdTinhToan tính toán
4	Nút lệnh	Name Caption	cmdThoat thoát

3. Code lệnh

```
Private Sub CmdTinhToan_Click()  
Dim x As Single  
Dim a As single  
Dim b as Single  
Do While True  
    a = val(InputBox("Nhap so a :"))  
    If IsNumeric(a) Then Exit Do  
    MsgBox "ban phai nhap a la chu so"  
Loop  
If a = 0 Then  
    MsgBox "So a khong duoc la 0"  
    Exit Sub  
End If  
Do While True  
    b = val(InputBox("Nhap so b :"))  
    If IsNumeric(b) Then Exit Do  
    MsgBox "ban phai nhap b la chu so"  
Loop  
x = -b / a  
txtkq.Text = a & "*x + " & b & " = " & x  
End Sub  
Private Sub CmdThoat_Click()  
Unload Me  
End Sub
```

CHƯƠNG 5

HÀM VÀ THỦ TỤC

5.1. Khái niệm chương trình con

Trong thực tế khi gặp một vấn đề lớn, phức tạp người ta không vội vàng giải quyết ngay vấn đề ấy mà tìm cách tách nó thành các vấn đề nhỏ đơn giản hơn. Việc này tương tự như trong dây chuyền sản xuất công nghiệp, người ta chia sản phẩm hoàn chỉnh thành nhiều sản phẩm nhỏ và mỗi nhà máy sẽ sản xuất những sản phẩm nhỏ đó, sau đó ghép lại thành sản phẩm hoàn chỉnh.

Tương tự như vậy trong lập trình, một vấn đề lớn và phức tạp sẽ tương ứng với một chương trình lớn và các vấn đề nhỏ tương ứng với các chương trình nhỏ gọi là các **chương trình con** (CTC). Đây là ý tưởng cơ bản của khái niệm lập trình có cấu trúc, làm cho toàn bộ chương trình sáng sủa, dễ hiệu chỉnh, dễ gỡ rối. Việc chia nhỏ chương trình thành các chương trình con (mỗi chương trình con gọi là một Module) gọi là “**chia để trị, chia để dễ điều khiển**”.

Mặt khác, trong khi lập trình có những đoạn chương trình được lặp đi lặp lại nhiều lần ở nhiều chỗ khác nhau. Để tránh rườm rà, viết đi viết lại, những đoạn chương trình này được thay thế bằng các chương trình con tương ứng, và khi cần ta chỉ việc gọi chương trình con modul đó với các tham số cần thiết mà không phải viết lại cả đoạn chương trình ấy.

Ví dụ: khi làm các bài toán lượng giác, thường xuyên ta cần tính SIN của một góc nào đó, như vậy ta nên lập một chương trình con có tên là SIN với tham số cần thiết là độ lớn của góc x.

Tóm lại: Chương trình con là một chương trình được viết theo những quy tắc đặc biệt để có thể được gọi thực hiện trong một chương trình khác.

Chương trình con giúp cho người lập trình có những thuận lợi sau:

- + Việc thử nghiệm, hiệu chỉnh, gỡ rối dễ dàng hơn.
- + Thực hiện nguyên tắc *"Viết một lần dùng nhiều lần, một người viết nhiều người dùng"*.

5.2. Hàm và thủ tục

■ **Hàm** (Function): là một CTC dùng để tính một giá trị nào đó và trả lại giá trị đó thông qua tên hàm, ví dụ hàm SIN cho giá trị SIN của x thông qua SIN(x).

■ **Thủ tục** (Sub): là một CTC dùng để thực hiện một công việc riêng biệt trên một tập hợp các tham số (hoặc không cần các tham số).

Sự khác nhau cơ bản và duy nhất của hai loại CTC này là hàm trả lại giá trị thông qua tên hàm và do đó được sử dụng trong biểu thức, thủ tục thì không trả lại kết quả thông qua tên của nó nên các thủ tục không thể viết trong các biểu thức.

Cấu trúc tổng quát của một hàm và thủ tục:

Function Tên_hàm(danh sách các tham số) As Kiểu_dữ_liệu

‘Các lệnh trong thân hàm

Tên_hàm=biểu thức giá trị

End Function

Sub Tên_thủ_tục(danh sách các tham số)

‘Các lệnh trong thân thủ tục

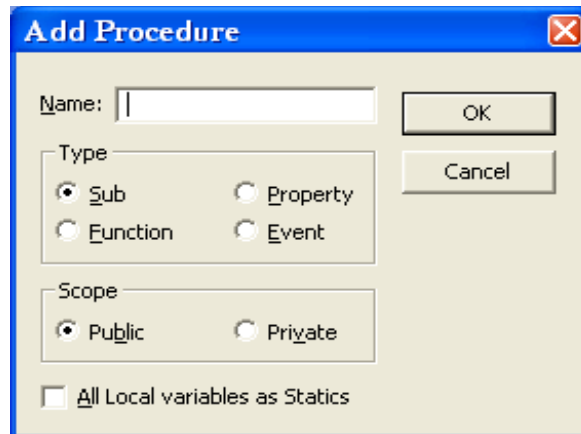
End Sub

Giải thích: Các tham số sau tên hàm và tên thủ tục chính là các giá trị được truyền cho hàm hoặc thủ tục khi chúng được gọi. Các tham số này là các tham số hình thức hay còn gọi là đối số. Đây là tùy chọn, nghĩa là trong lập trình người ta có thể dùng CTC không có tham số, điều đó chỉ cốt giúp cho lập trình sáng sủa, dễ đọc.

Cách xây dựng hàm và thủ tục trong Visual Basic:

Để tạo hàm và thủ tục ta làm như sau:

- Mở cửa sổ lệnh (Code)
- Chọn menu Tool
- Chọn Add Procedure. Hộp hội thoại Add Procedure xuất hiện:



- Xác định tên hàm hoặc thủ tục trong hộp Name
- Chọn Function hoặc Sub để xác định kiểu của CTC là hàm hay thủ tục
- Chọn Private để xác định CTC là cục bộ trong một biểu mẫu hoặc Public để xác định CTC có tác dụng trong toàn bộ chương trình.
- Chọn OK

Xuất hiện một trong hai mẫu sau: (giả sử chọn phạm vi là Private ở trên)

```
Private Function Tên_hàm() As Kiểu_dữ_liệu
End Function
```

```
Private Sub Tên_thủ_tục()
End Sub
```

Ví dụ: Viết hàm tìm giá trị lớn nhất trong ba số a, b, c

```
Private Function Max(a As Single, b As Single, c As Single) As Single
```

```
    Dim ketqua As Single
```

```
    ketqua=a
```

```
    If ketqua<b Then
```

```
        ketqua=b
```

```
    End If
```

```
    If ketqua<c Then
```

```
        ketqua=c
```

```
    End If
```

```
    Max= ketqua
```

```
End Function
```

Ví dụ: Viết thủ tục đổi giá trị trong hai biến a và b

```
Private Sub Doi(a As Single,b As Single)
```

```
    Dim tg As Single
```

```
    tg=a
```

```
    a=b
```

```
    b=tg
```

```
End Sub
```

5.3. Truyền tham số cho chương trình con

Để sử dụng hàm hoặc thủ tục ta chỉ việc viết lời gọi chúng và truyền vào các tham số thực sự (các giá trị cụ thể) thay thế cho các tham số hình thức khai báo trong tiêu đề của CTC.

Cú pháp:

■ Lời gọi hàm

Tên_Hàm(Danh sách tham số thực sự)

Chú ý: hàm phải luôn được gọi trong biểu thức.

■ Lời gọi thủ tục:

Call Tên_thủ_tục(Danh sách các tham số thực sự)

Hoặc:

Tên_thủ_tục Danh sách các tham số thực sự

Lời gọi hàm thường đặt trong biểu thức để tham gia vào quá trình tính giá trị của biểu thức. Lời gọi thủ tục thường được viết trong một chương trình chính như một câu lệnh độc lập.

Ví dụ:

■ Khi tính giá trị biểu thức S theo x, a, b trong đó có sử dụng hàm SIN và COS. Ta gọi hàm này trong chương trình chính như sau:

$$S = 2 * x^2 + \cos(a) - \sin(b)$$

■ Khi xây dựng xong thủ tục DOI thực hiện việc kiểm tra hai số x và y, nếu $x > y$ thì trao đổi giá trị x và y cho nhau. Trong chương trình chính ta gọi thủ tục ấy như sau

Call Doi(x,y)

Hoặc

Doi x,y

Việc chuyển tham số cho chương trình con là một cơ cấu thay thế tương ứng, nó cho phép một quá trình có thể lặp đi lặp lại nhiều lần với các toán hạng khác nhau. Ví dụ: Call Doi(x,y) sẽ thay thế x vào vị trí của a, y vào vị trí của b, tương tự với Call Doi(c,d).

Số lượng các tham số thực sự phải bằng số lượng các tham số hình thức, trừ những trường hợp tham số hình thức được chỉ định với tùy chọn Optional. Kiểu của tham số thực sự phải phù hợp với kiểu của tham số hình thức tương ứng.

Có hai cách truyền tham số cho chương trình con: tham trị và tham biến

a) Chuyển tham số bằng biến (tham biến):

Tham số hình thức trong phần tiêu đề của CTC sẽ được đặt sau từ khóa **ByRef** (mặc định).

Ví dụ:

Sub Doi(ByRef a as Single, ByRef b as Single)

Hoặc

Sub Doi(a as Single, b as Single)

Các tham số thực sự là các tham biến thì giá trị của nó có thể được thay đổi trong CTC.

Ví dụ:

```
Sub Doi(a As Single, b As Single, c As Single)
```

```
    Dim tg As Single
```

```
    tg=a
```

```
    a=b
```

```
    b=tg
```

```
    MsgBox "Trong CTC a = " + Str(a) + " ; b = " + Str(b)
```

```
End Sub
```

```
Private Sub CmdChayCT_Click()
```

```
    Dim a As Single, b As Single
```

```
    a = 2
```

```
    b = 3
```

```
    MsgBox "CTC có tham biến"
```

```
    MsgBox "Truoc khi gọi CTC a = " + Str(a) + " ; b = " + Str(b)
```

```
    Call Doi(a,b)
```

```
    MsgBox "Sau khi gọi CTC a = " + Str(a) + " ; b = " + Str(b)
```

```
End Sub
```

Kết quả là:

CTC có tham biến

Truoc khi gọi CTC a = 2 ; b = 3

Trong CTC a = 3 ; b = 2

Sau khi gọi CTC a = 3 ; b = 2

b) Chuyển tham số bằng giá trị (tham trị)

Tham số hình thức trong phần tiêu đề của CTC sẽ được đặt sau từ khóa **ByVal**.

Ví dụ:

```
Sub Doi(ByVal a as Single, ByVal b as Single)
```

Khi này các tham số hình thức tương ứng nhận giá trị của tham số thực như là giá trị ban đầu vào thời điểm thay vào CTC. CTC sau đó có thể thay đổi giá trị của các tham số này bên trong CTC bằng phép gán, song nó sẽ không làm thay đổi giá trị của tham số thực.

Ví dụ:

```
Sub Doi(ByVal a as Single, ByVal b as Single)
```

```
    Dim tg As Single
```

```
    tg=a
```

```
    a=b
```

```
    b=tg
```

```
    MsgBox “Trong CTC a = ” + Str(a) + “ ; b = ” + Str(b)
```

```
End Sub
```

```
Private Sub CmdChayCT_Click()
```

```
    Dim a As Single, b As Single
```

```
    a = 2
```

```
    b = 3
```

```
    MsgBox “CTC có tham trị”
```

```
    MsgBox “Truoc khi gọi CTC a = ” + Str(a) + “ ; b = ” + Str(b)
```

```
    Call Doi(a,b)
```

```
    MsgBox “Sau khi gọi CTC a = ” + Str(a) + “ ; b = ” + Str(b)
```

```
End Sub
```

Kết quả là:

CTC có tham trị

Truoc khi gọi CTC a = 2 ; b = 3

Trong CTC a = 3 ; b = 2

Sau khi gọi CTC a = 2 ; b = 3

5.4. Biến toàn cục, biến địa phương, khái niệm tầm tác dụng

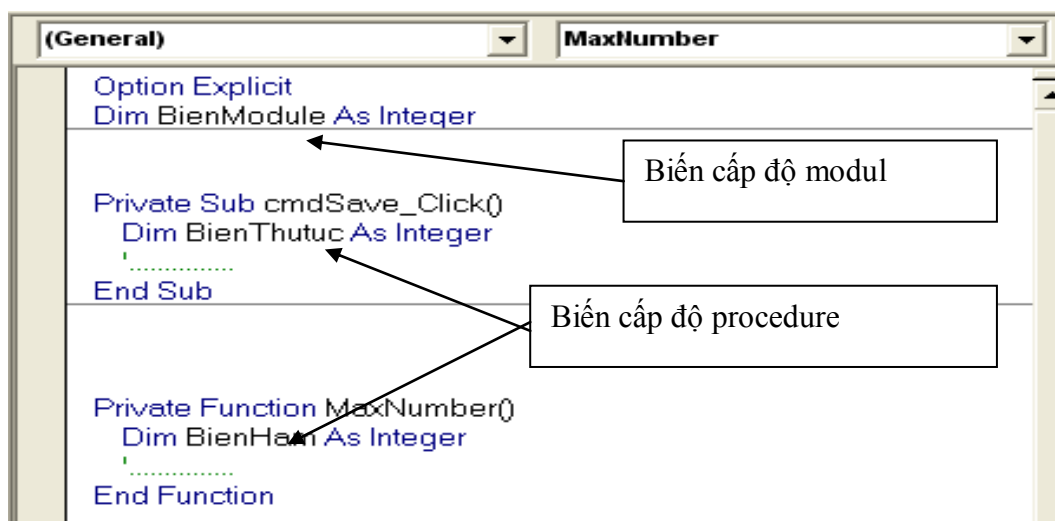
Visual Basic 6 là ngôn ngữ lập trình thủ tục, hướng sự kiện. Tập hợp các thủ tục, biến cố, các hàm liên quan đến một biểu mẫu được gọi là đơn thể chương trình biểu mẫu. Tập hợp các thủ tục, biến cố, các hàm không liên quan đến một biểu mẫu được gọi là đơn thể chương trình độc lập. Tập hợp thủ tục, hàm biến cố liên quan được đặt trong các

modul hoặc các đơn thể riêng. Một đơn thể có thể liên quan đến một biểu mẫu hoặc không liên quan đến một biểu mẫu nào cả.

Một biến có thể được khai báo bên trong một thủ tục/hàm hay khai báo ở phần đầu của đơn thể lập trình, trong phần khai báo General. Đối với các biến khai báo trong thủ tục hoặc hàm ta dùng từ khóa **Dim**, khi đó các biến này chỉ có tác dụng trong hàm hoặc thủ tục đó, khi ra khỏi hàm và thủ tục, biến sẽ mất tác dụng. Biến kiểu này được gọi là biến địa phương (hay biến cục bộ).

Đối với các biến khai báo trong các đơn thể chương trình ta có thể dùng các từ khóa **Private, Dim, Public, Global**. Nếu dùng từ khóa Private hoặc Dim, biến sẽ chỉ có tác dụng trong đơn thể chương trình đó, tức là nó có thể được sử dụng cho các hàm và thủ tục nằm trong đơn thể đó. Nếu dùng từ khóa Public hoặc Global thì biến có phạm vi tác dụng trong toàn bộ chương trình, nghĩa là nó có thể được sử dụng trong các đơn thể chương trình của dự án, trong tất cả các hàm và thủ tục của các đơn thể ấy. Khi đó người ta nói biến có phạm vi toàn cục.

Hình dưới đây sẽ cho ta thấy vị trí của khai báo biến:



Lưu ý:

Ở các CTC khác nhau, ở các mức độ khai báo khác nhau, ta có thể khai báo một biến có cùng tên với biến ở CTC khác, mức độ khai báo khác. Tên biến khi này không phải là một biến duy nhất mà là các biến khác nhau với tác dụng khác nhau. Ví dụ trong CTC A có biến địa phương X và trong CTC B có biến địa phương cũng tên là X. Lúc đó, máy sẽ dùng hai ô nhớ khác nhau để lưu trữ hai biến, khi ra khỏi CTC A, biến địa phương X trong CTC A mất tác dụng và khi ra khỏi CTC B, biến địa phương X trong B mất tác dụng.

Nếu một biến được khai báo với từ khóa Static thì biến đó có phạm vi tác động toàn cục dù nó được khai báo trong hàm hoặc thủ tục. Khi gọi CTC, sau mỗi lần sử dụng giá

trị của biến sẽ được nhớ lại và thoát khỏi CTC, giá trị của nó sẽ giữ nguyên giá trị nhận được trong CTC.

Ví dụ:

Fuction Cong(Number)

Static SoLuong

SoLuong= SoLuong + Number

Cong = SoLuong

End Function

Với đoạn chương trình trên, sau mỗi lần gọi hàm, khi thoát khỏi hàm Cong, biến SoLuong vẫn giữ lại giá trị của mình.

Phạm vi và tầm tác dụng của CTC tương tự như đối với biến. Không dùng từ khóa Dim khi đặt phạm vi tác dụng của CTC, có thể sử dụng các từ khóa Private, Public.

CHƯƠNG 6

DỮ LIỆU CÓ CẤU TRÚC KIỂU MẢNG

6.1. Khái niệm về mảng

Mảng là một tập hợp hữu hạn các phần tử có cùng tên, cùng kiểu dữ liệu. Giữa các phần tử của mảng được phân biệt với nhau qua số thứ tự của các phần tử hay còn gọi là chỉ số của các phần tử. Số phần tử của mảng có thể được xác định ngay từ khi khai báo mảng (trong mảng tĩnh) hoặc có thể thêm các phần tử trong quá trình xử lý (với mảng động). Mảng có biên trên và biên dưới, các thành phần trong mảng là liên tục giữa hai biên.

Trong trường hợp bài toán có nhiều giá trị tương tự nhau, thí dụ như trong bài toán cần quản lý điểm thi của một sinh viên, nếu phải đặt tên khác nhau cho từng biên (thí dụ: SV_mon1, SV_mon2, SV_mon3, SV_mon4, SV_mon5...) Khi đó phải đặt mỗi tên cho 1 môn học thì khá bất tiện trong quá trình xử lý.

Để thuận tiện trong quá trình xử lý có thể dùng một tên chung cho tất cả các môn học, rồi nói đến điểm của từng môn một bằng cách dùng một con số gọi là chỉ mục. Khi đó nhóm các điểm trên được gọi là 1 mảng lưu các giá trị điểm thi của sinh viên.

Muốn truy cập đến một phần tử nào của mảng ta viết như sau:

<Tên mảng>.(Chỉ số)

6.2. Khai báo mảng tĩnh

Mảng tĩnh là mảng có số phần tử được xác định ngay từ khi khai báo mảng và số phần tử này không thay đổi trong chương trình.

6.2.1. Khai báo mảng 1 chiều

Để khai báo mảng 1 chiều có 2 cách

Cách 1 :

DIM Ten_mang(n) As Kiểu_dữ_liệu

Trong đó:

n là số nguyên dùng để chỉ số lượng các phần tử trong mảng. Nếu khai báo như trên ta sẽ thu được mảng gồm n+1 phần tử và phần tử đầu tiên có chỉ số là 0, phần tử cuối cùng có chỉ số là n.

Quay lại với bài toán trên giả sử ta cần lưu điểm của 1 sinh viên trong 1 học kì với số môn học là 10. Khi đó ta sẽ khai báo

Dim StudentMarks(9) as Integer

Kế đó cần nhập điểm của sinh viên bằng cách viết **StudentMarks(i)**, mà i là chỉ mục $i < 9$. Giả dụ ta muốn nhập tất cả điểm của sinh viên

StudentMarks(0) = 6 ‘ Diem mon hoc 1

StudentMarks(1) = 6 ' Diem mon hoc 2

StudentMarks(2) = 7

StudentMarks(3) = 5

StudentMarks(4) = 9

StudentMarks(5) = 6

StudentMarks(6) = 8

StudentMarks(7) = 9

StudentMarks(8) = 10

StudentMarks(9) = 6 ' Diem mon hoc 10

Cách 2 :

DIM Ten_mang(chỉ số đầu TO chỉ số cuối) As Kiểu_dữ_liệu

Khi khai báo theo cách 2 ta thu được mảng có số phần tử:

$$n = (\text{chỉ số cuối} - \text{chỉ số đầu}) + 1.$$

Lưu ý: Với phần tử đầu có chỉ số bằng chỉ số đầu, phần tử cuối có chỉ số bằng chỉ số cuối.

Quay lại với bài toán nếu ta khai báo mảng để lưu trữ điểm 10 môn học ta khai báo:

Dim StudentMarks(1 to 10) AS Integer

Khi đó điểm môn 1 sẽ được gán vào StudentMarks(1), điểm môn 10 sẽ được gán vào StudentMarks(10).

6.2.2. Khai báo mảng nhiều chiều

Khi làm việc với giá trị các phần tử của một ma trận n chiều cách đơn giản nhất là khai báo một mảng có số chiều tương ứng. Để khai báo một mảng gồm n chiều ta thực hiện theo 1 trong 2 cách sau:

Cách 1 :

Dim Tên_mảng (n₁, n₂, n₃,... ,n_m) As Kiểu_dữ_liệu

Với n₁+1 là số phần tử của chiều 1

Với n₂+1 là số phần tử của chiều 2

...

Với n_m+1 là số phần tử của chiều thứ m

Ví dụ: khi khai báo mảng 2 chiều có tên Ma_Tran

Dim Ma_Tran(4,5) As Integer

Khi đó ta sẽ thu được mảng 2 chiều với số phần tử lần lượt là 5 và 6. Tổng số phần tử lúc này là $5*6=30$ phần tử

Để gán giá trị cho phần tử có chỉ số hàng là 3, chỉ số cột là 4 ta thực hiện:
`Ma_Tran(3,4) = 10`

Cách 2 :

Dim Tên_mảng (a1 to b1,a2 to b2,...,am to bm) As Kiểu_dữ_liệu

a_1 và b_1 lần lượt là chỉ số phần tử đầu và cuối chiều 1

a_2 và b_2 lần lượt là chỉ số phần tử đầu và cuối chiều 2

...

a_m và b_m lần lượt là chỉ số phần tử đầu và cuối chiều m

Khi đó tổng số phần tử trong mảng lúc này là: $n = (b_1 - a_1 + 1) * (b_2 - a_2 + 1) * \dots * (b_m - a_m + 1)$

6.3. Khai báo mảng động (Dynamic Array)

Mảng động là mảng có các phần tử trong mảng không được xác định trước, trong quá trình xử lý, số lượng các phần tử của mảng mới được khai báo hoặc các phần tử trong mảng có thể bổ xung thêm phù hợp với các yêu cầu của bài toán.

Để khai báo một mảng động ta khai báo :

DIM Tên_mảng() As Kiểu_dữ_liệu

Vì không để một con số ở giữa hai dấu ngoặc đơn nên VB6 biết là dùng mảng động và số phần tử của nó có thể sẽ thay đổi trong tương lai. Khi nào muốn thay đổi số phần tử của mảng động dùng **ReDim**:

ReDim Tên_mảng(n)

Hoặc

ReDim Preserve Tên_mảng(m)

Với từ khoá **ReDim** VB6 sẽ khởi tạo lại tất cả các giá trị phần tử của mảng (nếu chúng đã tồn tại) khi đó giá trị sẽ được gán=0 với tất cả các phần tử kiểu số, gán="" với các phần tử kiểu xâu.

Với từ khoá **ReDim Preserve** khi đó VB6 chỉ thêm vào mảng phần tử mới và giá trị các phần tử cũ đã tồn tại trong mảng vẫn không đổi.

Ví dụ:

`Dim Matrix () As Integer` ‘ Khai báo mảng Matrix là mảng động

...

`ReDim Matrix(5)` ‘ Gán mảng Matrix là mảng có 6 phần tử

...

ReDim Matrix(6) ‘ Gán mảng Matrix là mảng có 7 phần tử và các phần tử trong mảng đều bị gán =0

...

ReDim Preserve Matrix (7) ‘ Mở rộng Matrix là mảng có 8 phần tử trong đó các phần tử cũ của mảng vẫn giữ nguyên giá trị. Phần tử thứ 8 được thêm vào sẽ được gán =0

6.4. Một số thuật toán về mảng

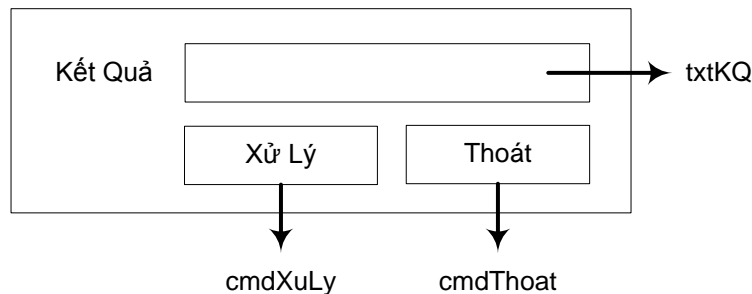
Dạng 1: Tìm giá trị lớn nhất hoặc nhỏ nhất trong mảng

Ví dụ: Nhập vào một mảng số nguyên gồm n phần tử. Tìm giá trị lớn nhất trong mảng. In kết quả ra màn hình. Yêu cầu:

- Sử dụng các đối tượng Form, Lable, Textbox, Command Button,... để thiết kế giao diện bài toán
- Viết mã lệnh bài toán.

Hướng dẫn:

- Thiết kế giao diện



- Mã lệnh bài toán.

```
Private Sub cmdThoat_Click()
    End
End Sub

Private Sub cmdXuLy_Click()
    Dim i As Integer, n As Integer, max As Integer
    Dim a(1 To 100) As Integer
    Do
        n = Val(InputBox("Nhập n = "))
```



```
Loop Until n > 0 And n <= 100
For i = 1 To n
    a(i) = Val(InputBox("Nhập a(" + Str(i) + ") = "))
Next i
max = a(1)
For i = 2 To n
    If a(i) > max Then
        max = a(i)
    End If
Next i
txtKQ.Text = "Gia Tri Lon Nhat = " + Str(max)
End Sub
```

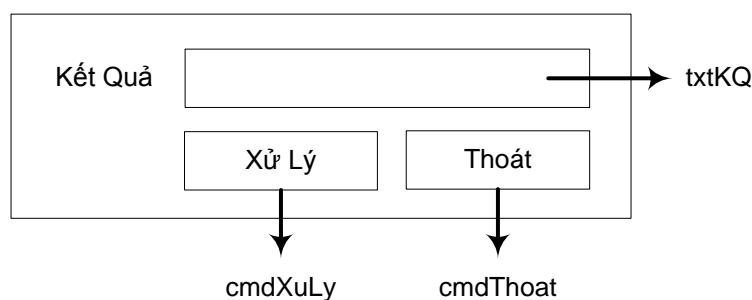
Dạng 2: Sắp xếp các phần tử của mảng theo thứ tự tăng hoặc giảm dần

Ví dụ: Nhập vào một mảng số nguyên gồm n phần tử. Sắp xếp các phần tử trong mảng theo thứ tự giảm dần. In kết quả ra màn hình. Yêu cầu:

- Sử dụng các đối tượng Form, Lable, Textbox, Command Button,... để thiết kế giao diện bài toán
- Viết mã lệnh bài toán.

Hướng dẫn:

- Thiết kế giao diện



- Mã lệnh bài toán.

```
Private Sub cmdThoat_Click()
    End
End Sub
Private Sub cmdXuLy_Click()
```

Dim i As Integer, j As Integer, n As Integer, tg As Integer

Dim a(1 To 100) As Integer

Do

n = Val(InputBox("Nhập n = "))

Loop Until n > 0 And n <= 100

For i = 1 To n

a(i) = Val(InputBox("Nhập a(" + Str(i) + ") = "))

Next i

For i = 1 To n-1

For j = i+1 to n

If a(i) < a(j) Then

tg = a(i)

a(i) = a(j)

a(j) = tg

End If

Next j

Next i

txtKQ.Text = ""

For i = 1 to n

txtKQ.Text = txtKQ.Text + Str(a(i)) + “;”

Next i

End Sub

Dạng 3: Chèn một phần tử b vào vị trí k của mảng

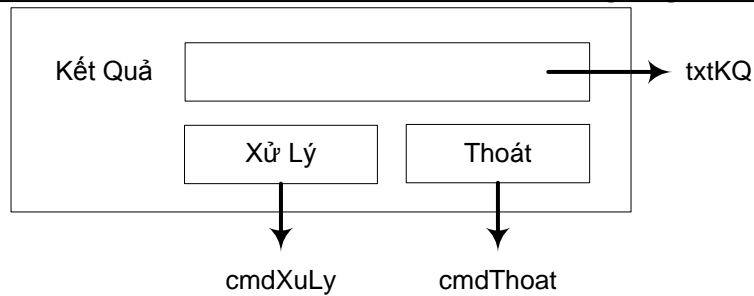
Ví dụ: Nhập vào một mảng số nguyên gồm n phần tử. Chèn số b vào vị trí thứ k trong mảng. In kết quả ra màn hình. Yêu cầu:

■ Sử dụng các đối tượng Form, Lable, Textbox, Command Button,... để thiết kế giao diện bài toán

■ Viết mã lệnh bài toán.

Hướng dẫn:

■ Thiết kế giao diện



■ Mã lệnh bài toán.

```
Private Sub cmdThoat_Click()
```

```
End
```

```
End Sub
```

```
Private Sub cmdXuLy_Click()
```

```
Dim i As Integer, b As Integer, k As Integer, n As Integer
```

```
Dim a(1 To 100) As Integer
```

```
Do
```

```
    n = Val(InputBox("Nhập n = "))
```

```
Loop Until n > 0 And n <= 100
```

```
For i = 1 To n
```

```
    a(i) = Val(InputBox("Nhập a(" + Str(i) + ") = "))
```

```
Next i
```

```
b = Val(InputBox("Nhập b = "))
```

```
Do
```

```
    k = Val(InputBox("Nhập k = "))
```

```
Loop Until k > 0 And k <= n
```

```
For i = n To k Step -1
```

```
    a(i+1) = a(i)
```

```
Next i
```

```
a(k) = b
```

```
n = n + 1
```

```
txtKQ.Text = ""
```

```
For i = 1 to n
```

```
    txtKQ.Text = txtKQ.Text + Str(a(i)) + “;”
```

Next i

End Sub

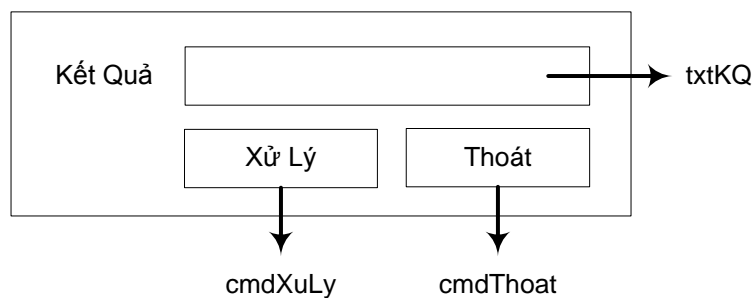
Dạng 4: Xóa đi phần tử ở vị trí thứ k của mảng

Ví dụ: Nhập vào một mảng số nguyên gồm n phần tử. Xóa đi phần tử ở vị trí thứ k trong mảng. In kết quả ra màn hình. Yêu cầu:

- Sử dụng các đối tượng Form, Lable, Textbox, Command Button,... để thiết kế giao diện bài toán
- Viết mã lệnh bài toán.

Hướng dẫn:

- Thiết kế giao diện



- Mã lệnh bài toán.

```
Private Sub cmdThoat_Click()
```

```
End
```

```
End Sub
```

```
Private Sub cmdXuLy_Click()
```

```
Dim i As Integer, k As Integer, n As Integer
```

```
Dim a(1 To 100) As Integer
```

```
Do
```

```
    n = Val(InputBox("Nhập n = "))
```

```
Loop Until n > 0 And n <= 100
```

```
For i = 1 To n
```

```
    a(i) = Val(InputBox("Nhập a(" + Str(i) + ") = "))
```

```
Next i
```

```
Do
```

```
    k = Val(InputBox("Nhập k = "))
```

```

Loop Until k > 0 And k <= n
For i = k To n - 1
    a(i) = a(i+1)
Next i
n = n - 1
txtKQ.Text = ""
For i = 1 to n
    txtKQ.Text = txtKQ.Text + Str(a(i)) + ";"
Next i
End Sub
    
```

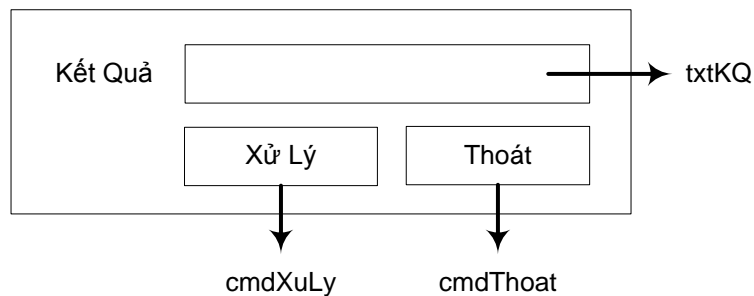
Dạng 5: Một số bài toán khác về mảng

Ví dụ: Nhập vào một mảng số nguyên gồm n phần tử. Tính trung bình cộng của các phần tử chia hết cho 3 trong mảng. In kết quả ra màn hình. Yêu cầu:

- Sử dụng các đối tượng Form, Lable, Textbox, Command Button,... để thiết kế giao diện bài toán
- Viết mã lệnh bài toán.

Hướng dẫn:

- Thiết kế giao diện



- Mã lệnh bài toán.

```

Private Sub cmdThoat_Click()
    End
End Sub

Private Sub cmdXuLy_Click()
    Dim i As Integer, n As Integer, tbc As Single, dem As Integer
    Dim a(1 To 100) As Integer
    
```

```
Do
    n = Val(InputBox("Nhập n = "))
Loop Until n > 0 And n <= 100
For i = 1 To n
    a(i) = Val(InputBox("Nhập a(" + Str(i) + ") = "))
Next i
tbc = 0
d = 0
For i = 1 To n
    If a(i) MOD 3 = 0 Then
        tbc = tbc + a(i)
        d = d + 1
    End If
Next i
If d = 0 Then
    txtKQ.Text = "Mang không tồn tại phần tử nào chia hết cho 3"
Else
    tbc = tbc/dem
    txtKQ.Text = "TBC = " + Str(dem)
End If
End Sub
```

Bài giảng Tin học đại cương

Thời lượng: 60 tiết

NỘI DUNG CHƯƠNG TRÌNH

☐ Tin học căn bản (20 tiết)

- ✓ Cơ bản về CNTT và máy tính (4 tiết)
- ✓ Sử dụng hệ điều hành Windows (12 tiết)
- ✓ Công nghệ Internet (4 tiết)

☐ Tin học văn phòng (40 tiết)

- ✓ Sử dụng phần mềm soạn thảo văn bản MS Word (16 tiết)
- ✓ Sử dụng phần mềm bảng tính điện tử MS Excel (16 tiết)
- ✓ Sử dụng phần mềm trình chiếu MS PowerPoint (8 tiết)

Module 1

**CƠ BẢN VỀ CÔNG NGHỆ
THÔNG TIN VÀ MÁY TÍNH**

I/ Các khái niệm cơ bản

- 1/ Khái niệm về tin học:** Tin học là ngành khoa học công nghệ nghiên cứu các phương pháp, các quá trình xử lý thông tin một cách tự động dựa trên phương tiện kỹ thuật là máy tính điện tử.
- 2/ Máy tính điện tử (Computer):** Là một thiết bị điện tử dùng để lưu trữ và xử lý thông tin theo các chương trình định trước do con người định ra.

II/ Các giai đoạn phát triển của máy tính

- **Thế hệ thứ nhất (1945-1955) máy tính dùng đèn điện tử:** Máy tính đầu tiên có tên ENIAC (Electronic Numerical Integrator and Computer) nặng khoảng 30 tấn (1946) ở Mỹ.
- **Thế hệ thứ hai (1955-1965) máy tính dùng thiết bị bán dẫn:** Dùng linh kiện mới là Transistor (được phòng thí nghiệm Bell phát triển năm 1948). Bộ nhớ máy tính được tăng lên đáng kể và trở nên nhỏ gọn hơn.
- **Thế hệ thứ ba (1965-1980) dùng mạch hợp tích hợp IC:** Công nghệ điện tử giờ đã phát triển rất nhanh cho phép đặt hàng chục Transistor vào một vỏ chung gọi là con chip.

II/ Các giai đoạn phát triển của máy tính (tt)

- **Thế hệ thứ tư (1980-199x) sử dụng công nghệ (VLSI):** Vào những năm 80 công nghệ (VLSI Very Large Scale Integrator) ra đời cho phép tích hợp trong một con chip hàng triệu Transistor khiến cho máy tính trở nên nhỏ hơn, nhanh hơn với tốc độ hàng triệu phép tính một giây là nền tảng cho chiếc máy tính PC (Personal Computer) ngày nay.
- Năm 1981, IBM giới thiệu máy tính cá nhân PC (Personal Computer) đầu tiên.

III/ Thông tin và biểu diễn thông tin trong máy tính

1/ Khái niệm về thông tin: Bất cứ thông báo hay một tín hiệu gì đều được coi là một thông tin. Việc trao đổi hay tiếp nhận thông tin theo nghĩa thông thường được con người trao đổi theo nhiều cách khác nhau (thính giác, thị giác, khứu giác ...).

2/ Biểu diễn thông tin trong máy tính: Do máy tính được chế tạo dựa trên các thiết bị điện tử chỉ có hai trạng thái đóng và mở, tương ứng với hai số 0 và 1. Nên để lưu trữ thông tin trong máy, máy tính dùng hệ đếm nhị phân (Binary) tức là hệ đếm được biểu diễn với hai chữ số 0 và 1.

3/ Các đơn vị đo thông tin:

➤ **Đơn vị cơ sở: Bit (0 hoặc 1)**

3/ Các đơn vị đo thông tin (tt)

- Đơn vị cơ bản: Byte: 1 Byte = 8 Bit
- Các bội số của Byte:
 - ✓ Kilobyte: 1 KB = 2^{10} = 1024 Byte
 - ✓ Megabyte: 1 MB = 1024 KB
 - ✓ Gigabyte: 1GB = 1024 MB
 - ✓ Terabyte: 1TB= 1024 GB

4/ Chuyển đổi giữa các hệ đếm

➤ **Hệ 10 sang hệ 2:** Thực hiện liên tiếp các phép chia cho 2 cho đến khi thương số bằng 0. Số nhị phân tương ứng là các kết quả của phép dư chia cho 2 lấy từ đáy lên.

➤ **Hệ 2 sang hệ 10:**

✓ Xét số trong hệ cơ số 2 với biểu diễn như sau:

$$N_2 = d_n d_{n-1} \dots d_1 d_0$$

Khi đó trong hệ cơ số 10 số N sẽ là:

$$N_{10} = d_n * 2^n + d_{n-1} * 2^{n-1} + \dots + d_1 * 2^1 + d_0 * 2^0$$

IV/ Cấu trúc cơ bản của máy tính

Hệ thống máy tính bao gồm hai hệ thống con:

- 1/ Phần cứng:** Bao gồm toàn bộ máy và các thiết bị ngoại vi là các thiết bị điện tử được kết hợp với nhau. Nó thực hiện chức năng xử lý thông tin ở mức thấp nhất tức là các tín hiệu nhị phân.
- 2/ Phần mềm:** Là các chương trình (Programs) do người sử dụng tạo ra điều khiển các hoạt động phần cứng của máy tính để thực hiện các nhiệm vụ phức tạp theo yêu cầu của người sử dụng. Phần mềm của máy tính được phân làm 2 loại: Phần mềm hệ thống (System Software) và phần mềm ứng dụng (Applications Software).

V/ Các loại máy tính

1/ **Máy tính lớn (Mainframe):** Là loại máy tính có kích thước vật lý lớn, mạnh, phục vụ tính toán phức tạp.



V/ Các loại máy tính (tt)

2/ Siêu máy tính (Super Computer): Là một hệ thống gồm nhiều máy lớn ghép song song có tốc độ tính toán cực kỳ lớn và thường dùng trong các lĩnh vực đặc biệt, chủ yếu trong quân sự và vũ trụ. Siêu máy tính Deep Blue là một trong những chiếc thuộc loại này.



V/ Các loại máy tính (tt)

3/ Máy tính cá nhân PC (Personal Computer): Còn gọi là máy tính để bàn (Desktop). Hầu hết các máy tính được sử dụng trong các văn phòng, gia đình.



V/ Các loại máy tính (tt)

4/ Máy tính xách tay (Laptop): Máy tính Laptop là tên của một loại máy tính nhỏ, gọn có thể mang đi theo người, có thể chạy bằng pin. Một tên gọi khác “Notebooks” chỉ một Laptop nhỏ.



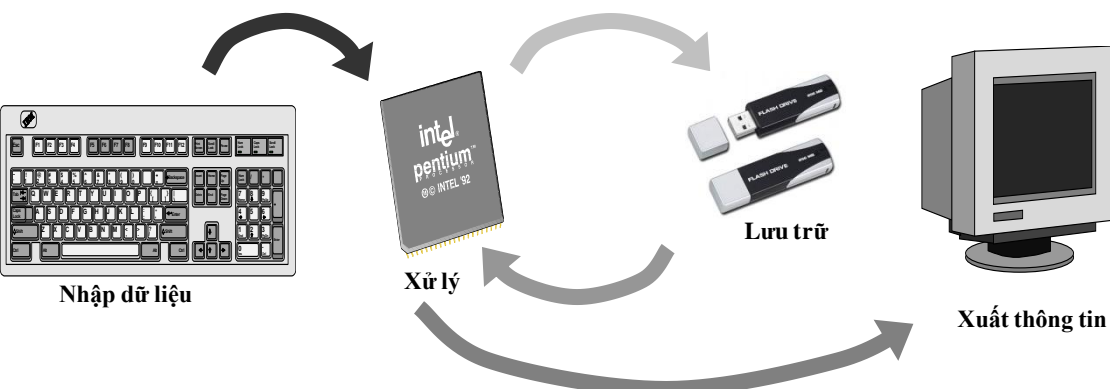
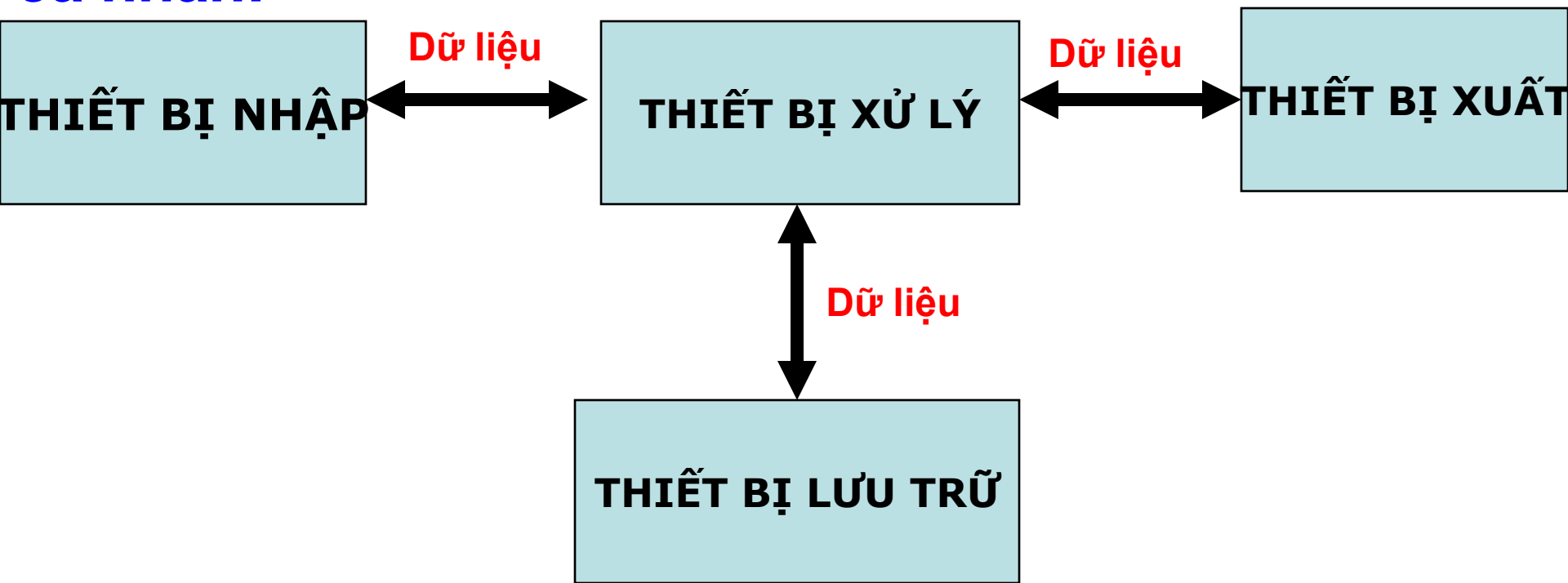
V/ Các loại máy tính (tt)

5/ Máy tính bỏ túi (Pocket PC): Hiện nay, thiết bị kỹ thuật số cá nhân (PDA) có chức năng rất phong phú, như kiểm tra e-mail, ghi chú ngắn gọn, xem phim, lướt Internet, nghe nhạc hay soạn tài liệu văn phòng... nhiều máy hiện nay được tích hợp chức năng điện thoại di động.



VI/ CẤU TẠO CỦA MỘT MÁY TÍNH CÁ NHÂN

Sơ đồ mô tả các bộ phận cơ bản trong một máy tính cá nhân:



Chu trình xử lý thông tin

VII/ CẤU TẠO CỦA MỘT MÁY TÍNH CÁ NHÂN (tt)

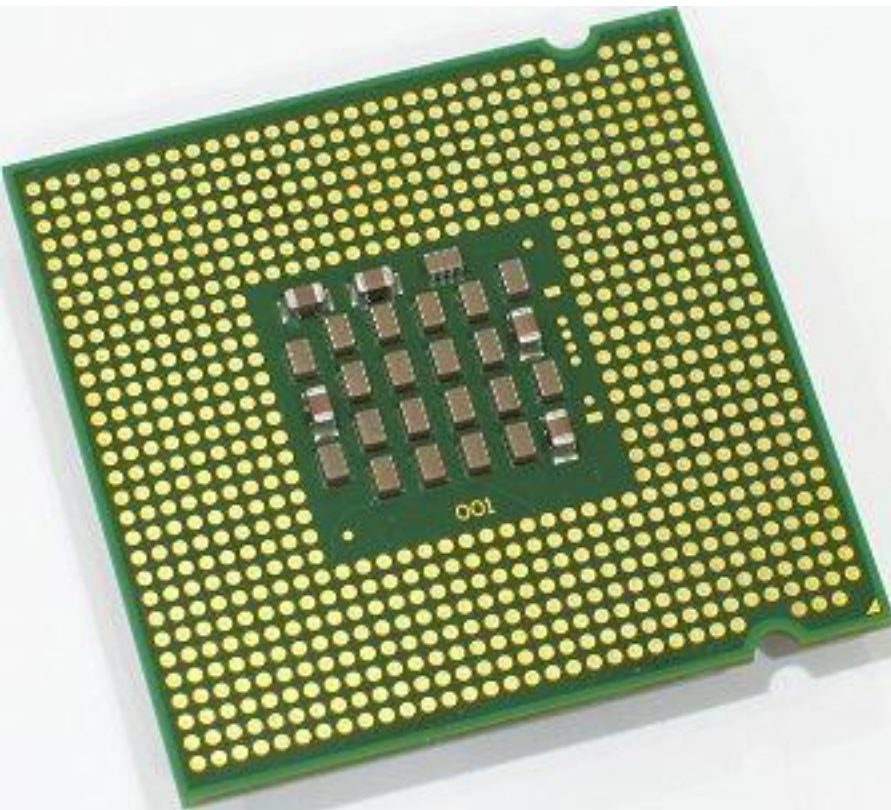
1/ Thiết bị nhập: Là thiết bị có nhiệm vụ đưa thông tin vào máy tính để xử lý.

➤ **Các thiết bị nhập thông dụng:** Chuột, bàn phím, máy quét, webcam.



VII/ CẤU TẠO CỦA MỘT MÁY TÍNH CÁ NHÂN (tt)

2/ Thiết bị xử lý: Xử lý thông tin, điều khiển hoạt động máy tính. Thiết bị xử lý bao gồm: bo mạch chủ, bộ vi xử lý.



Bộ xử lý trung tâm CPU (Central Processing Unit)


- Khối xử lý trung tâm là bộ não của máy tính, điều khiển mọi Hoạt động của máy tính bao gồm 4 thành phần chính:
 - ✍ **Khối điều khiển** (Control Unit): Xác định và sắp xếp các lệnh theo thứ tự điều khiển trong bộ nhớ.
 - ✍ **Khối tính toán** (Arithmetic Logical Unit): Là nơi thực hiện hầu hết các thao tác tính toán của toàn bộ hệ thống như: +, -, *, /, >, <...
 - ✍ **Đồng hồ** (Clock): Không mang theo nghĩa đồng hồ thông thường, mà là bộ phận phát xung nhịp nhằm đồng bộ hoá sự **Hoạt động** của CPU.
 - ✍ **Thanh ghi** (Register): Là nơi lưu giữ tạm thời các chỉ thị từ bộ nhớ trong khi chúng được xử lý. Tốc độ truy xuất thông tin nơi đây là nhanh nhất.
- 💾 CPU là một bộ phận quan trọng nhất trong máy tính, quy định tốc độ của máy tính.

VI/ CẤU TẠO CỦA MỘT MÁY TÍNH CÁ NHÂN (tt)

3/ Bộ nhớ máy tính (Thiết bị lưu trữ): Được dùng để lưu trữ thông tin và dữ liệu. Bộ nhớ máy tính được chia làm 2 loại: Bộ nhớ trong và bộ nhớ ngoài.

3.1/ Bộ nhớ trong (bộ nhớ trong gắn trực tiếp vào bo mạch chủ): Là nơi lưu giữ chương trình và xử lý thông tin chủ yếu là dưới dạng nhị phân. Có hai loại bộ nhớ trong là RAM và ROM.

➤ **RAM (Random Access Memory):** Hay Bộ nhớ truy cập ngẫu nhiên: Là thiết bị lưu trữ dữ liệu tạm thời trong quá trình máy tính làm việc, dữ liệu sẽ bị mất vĩnh viễn khi không còn nguồn điện cung cấp.

 **Máy tính sẽ chạy nhanh hơn nếu có nhiều RAM.**



Bộ nhớ trong (tt)

- **ROM (Read Only Memory):** Bộ nhớ chỉ đọc (ROM) là một loại chip nhớ đặc biệt được thiết lập từ khi sản xuất máy, nó lưu trữ các phần mềm có thể đọc nhưng không thể viết lên được. Thông tin không bị mất khi tắt máy.



3/ Bộ nhớ máy tính (tt)

3.2/ **Bộ nhớ ngoài:** Là các thiết bị lưu trữ gắn gián tiếp vào bo mạch chủ thông qua dây cáp dữ liệu, các khe cắm mở rộng

...

- ✓ Bộ nhớ ngoài là nơi lưu trữ hệ điều hành, phần mềm ứng dụng, dữ liệu của máy tính.
- ✓ Bộ nhớ ngoài bao gồm ổ cứng, đĩa mềm, CD, ổ cứng USB...



4/ Thiết bị xuất

□ Các thiết bị xuất dùng để hiển thị kết quả xử lý của máy tính. Một số thiết bị tiêu biểu bao gồm: Màn hình, máy in, máy chiếu, loa...



VII/ Phần mềm (Software)

1/ Phần mềm hệ thống (System Software): Bao gồm Hệ điều hành (Operating System), các phần mềm đi kèm thiết bị phần cứng (Driver).

➤ **Khái niệm hệ điều hành:** Là một hệ thống phần mềm điều hành mọi **Hoạt động** cơ bản của máy tính và các thiết bị ngoại vi. Nó là nền tảng cho các ứng dụng và chương trình chạy trên nó.

Các chức năng cơ bản của HĐH:

- ✓ Điều khiển việc **Hoạt động** của máy tính và các thiết bị ngoại vi.
- ✓ Tổ chức cấp phát và thu hồi vùng nhớ.
- ✓ Điều khiển việc thực thi chương trình.
- ✓ Quản lý việc truy xuất thông tin.

➤ **Một số HĐH thông dụng:**

1/ Phần mềm hệ thống (tt)

 **MS DOS:** Hệ điều hành đơn nhiệm, làm việc với giao diện dòng lệnh.

 **Các phiên bản của hệ điều hành Windows:**

- Windows 3.x, Windows 95, Windows 98, Windows Me.
- Windows 2000 Pro, Windows XP, Windows Vista.
- Windows NT, Windows 2000 Server, Windows 2003 Server: dành riêng cho máy chủ - hệ điều hành mạng.

 Ngoài ra còn có các hệ điều hành: Linux, Unix, OS/2



2/ Phần mềm ứng dụng

□ **Phần mềm ứng dụng:** Là các chương trình được thiết kế nhằm phục vụ nhu cầu sử dụng của con người để có thể hoàn thành một hay nhiều công việc nào đó.

VIII/ Virus máy tính

- 1/ Virus máy tính là gì?** Virus máy tính là một chương trình phần mềm, và chương trình này được thiết kế để có thể tự mình làm những việc mà người viết ra nó đã định trước. Đặc điểm đặc trưng của Virus là khả năng tự nhân bản.
- ❑ Virus có khả năng phá hỏng phần mềm, xóa sạch dữ liệu, ăn cắp mật khẩu...



2/ Phòng chống Virus

- Cài phần mềm diệt Virus BKAV, D32, Norton Anti Virus, Symatec.
- Quét Virus trước khi sử dụng các đĩa mềm, USB.
- Sao lưu dữ liệu thường xuyên.

Module 2

SỬ DỤNG HỆ ĐIỀU HÀNH WINDOWS

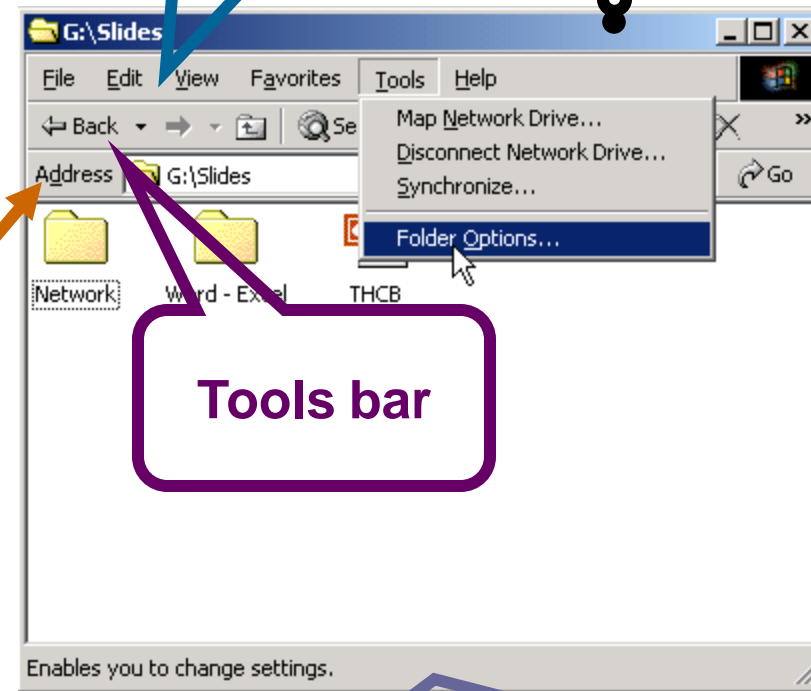
Bài 1: CÁC KHÁI NIỆM CƠ BẢN VỀ HỆ ĐIỀU HÀNH WINDOWS

1/ GIỚI THIỆU HỆ ĐIỀU HÀNH WINDOWS: Hệ điều hành Windows là hệ điều hành được hãng Microsoft phát triển theo ý tưởng giao tiếp người-máy qua các cửa sổ chứa biểu tượng nên người dùng có thể dễ dàng thao tác bằng bàn phím hay chuột máy tính.

- **Windows** là một hệ điều hành đa nhiệm, tức là trong cùng một thời điểm người dùng có thể mở được nhiều chương trình cùng một lúc tùy theo bộ nhớ của máy mình đang sử dụng.
- HĐH Windows dùng các cửa sổ để giao tiếp với người sử dụng.
- Cửa sổ đầu tiên là màn hình nền Desktop. Các cửa sổ tiếp theo sẽ nằm trên màn hình Desktop.

2/ Cấu trúc của một Window

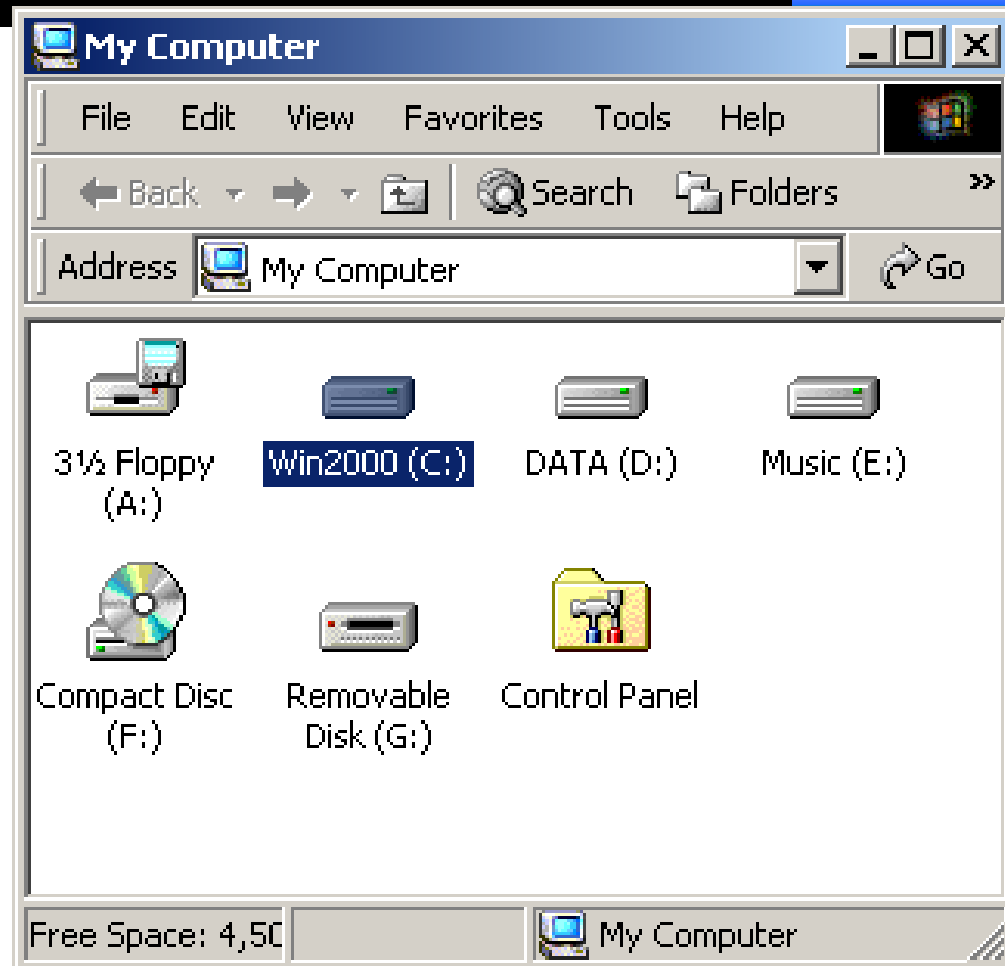
1. Thanh tiêu đề (Title Bar) chứa biểu tượng, tiêu đề cửa sổ, nút thu nhỏ, phóng to, tắt cửa sổ.
2. Thanh thực đơn lệnh (Menu bar) chứa các lệnh để thao tác.
3. Thanh công cụ (Tool bar): chứa các nút lệnh giúp người sử dụng thao tác nhanh với ứng dụng.
4. Thanh địa chỉ (Address bar): hiển thị đường dẫn của thư mục.
5. Thanh trạng thái (Status bar): hiển thị thông tin hỗ trợ người sử dụng.



3/ Cấu trúc lưu trữ

➤ Tất cả dữ liệu của máy tính đều lưu trữ trong các thiết bị lưu trữ ngoài.

➤ Cửa sổ My Computer quản lý tất cả các thiết bị lưu trữ ngoài: đĩa mềm, đĩa cứng, đĩa CD, DVD, USB ...

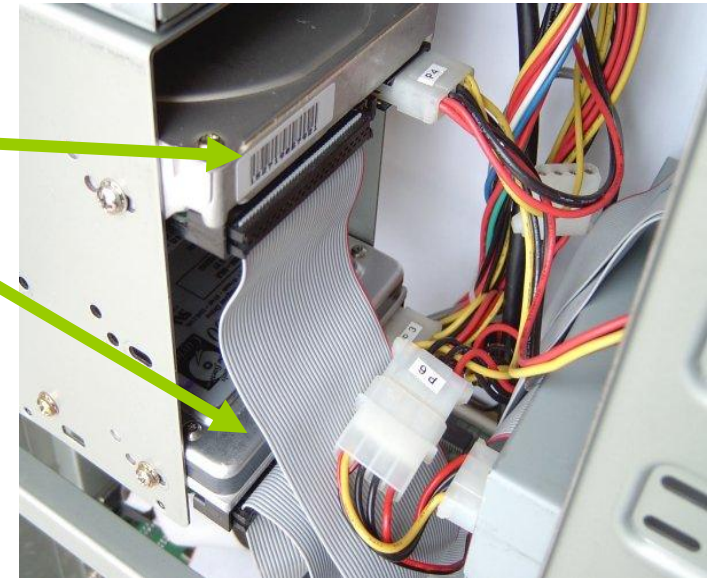


3/ Cấu trúc lưu trữ



☐ Quy ước đặt tên các ổ đĩa:

- A, B: đĩa mềm
- C:, D:, .. phân vùng ổ cứng.
- Các ký tự tiếp theo cho ổ CD, USB,... tùy thuộc vào số lượng đĩa gắn vào máy...



4/ Khái niệm tệp tin (File)

- Là tập hợp các thông tin có cùng bản chất và được lưu trữ như một đơn vị lưu trữ dữ liệu trên các vật mang thông tin (đĩa từ, băng từ, USB....), tùy theo từng kiểu tệp mà nội dung chứa đựng trong đó sẽ khác nhau.
- Tệp tin có 3 đặc trưng chính là: Tên tệp, kích thước và ngày tháng cập nhật.
- **Tên tệp (Name)** : Gồm có 2 phần là **Tên tệp** và **phần mở rộng**
 - ✓ **Tên tệp**: Là một chuỗi các ký tự, không chứa ký tự đặc biệt như: * ? \ / " : < >. Với hệ điều hành MSDOS tối đa là 8 ký tự và không chứa khoảng trống, với hệ điều hành Windows tối đa là 255 ký tự.

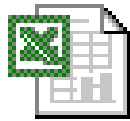
4/ Khái niệm tệp tin (tt)

✓ **Phân mở rộng:** Là cách viết tắt cho kiểu tệp tin, tôi đa là 3 ký tự, giữa chúng không chứa khoảng trống và ký tự đặc biệt. Phần này thường do hệ thống tự quy định và đặc trưng cho từng kiểu tệp. Phần mở rộng để xác định tệp tin đó được tạo ra từ ứng dụng nào, hoặc được ứng dụng nào sử dụng. Phần mở rộng ngăn cách với tên tệp bằng một dấu chấm (.).

➤ Một số kiểu tệp tin thông dụng: .exe – tệp tin tự thi hành, .doc – tệp tin văn bản Word, .txt tệp tin văn bản sơ cấp, .mp3 – tệp tin nhạc nén, .jpg – tệp tin hình ảnh ...



BAI TAP
EXCEL



BANG LUONG



CANH THIEN
NHIEN



GIAY MOI



NHUNG CHIEU
KHONG CO EM



TAI LIEU



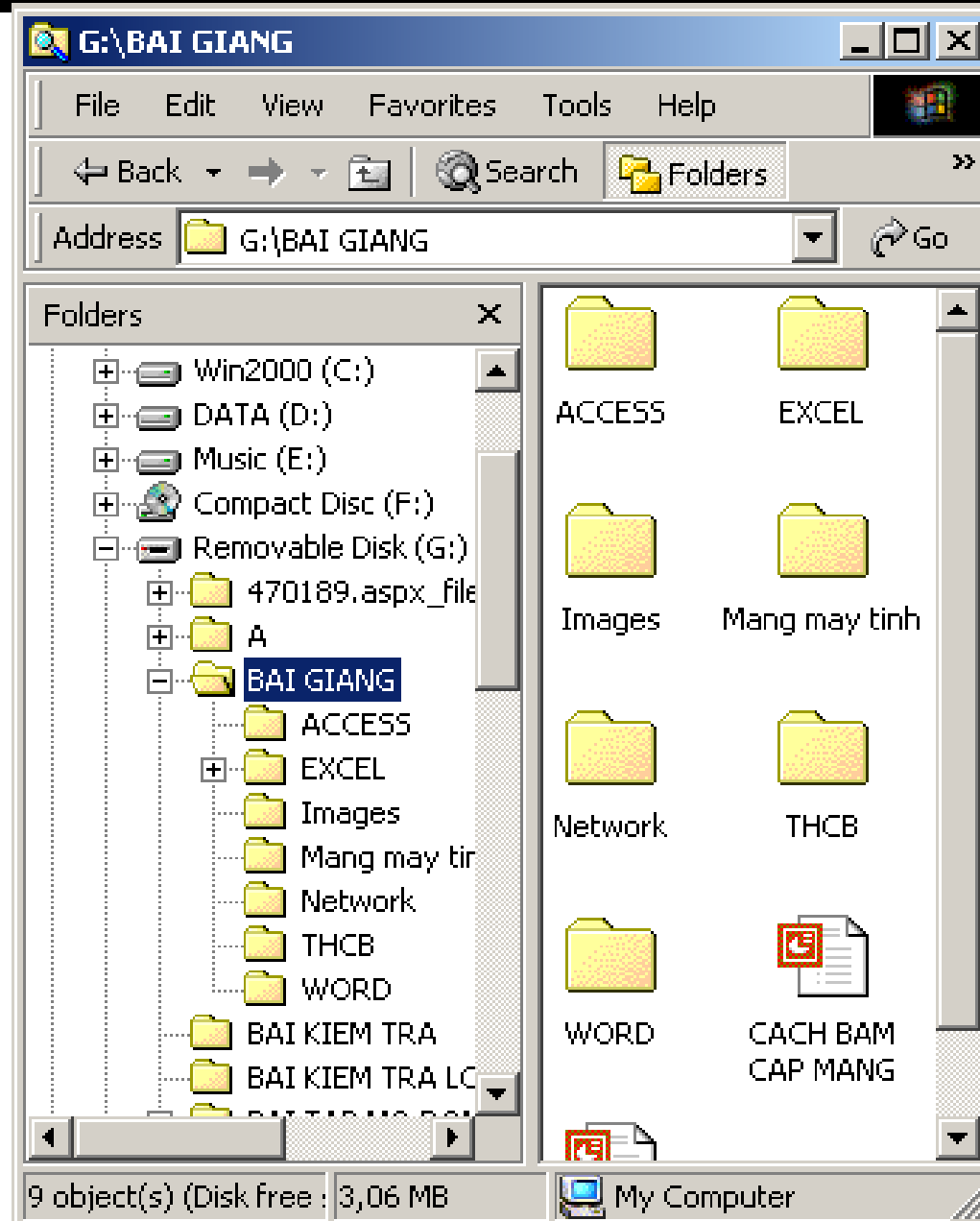
THONG BAO








VAN BAN

5/ Khái niệm thư mục (Folder)

Để tạo thuận lợi cho quá trình lưu trữ và truy xuất dữ liệu **Windows** cho phép tổ chức các dữ liệu được lưu dưới dạng cây thư mục. Với mỗi ổ đĩa là một thư mục gốc trong đó chứa các tập tin hoặc các thư mục con. Bản thân trong mỗi thư mục con có thể chứa trong nó các tập tin và các thư mục con khác.

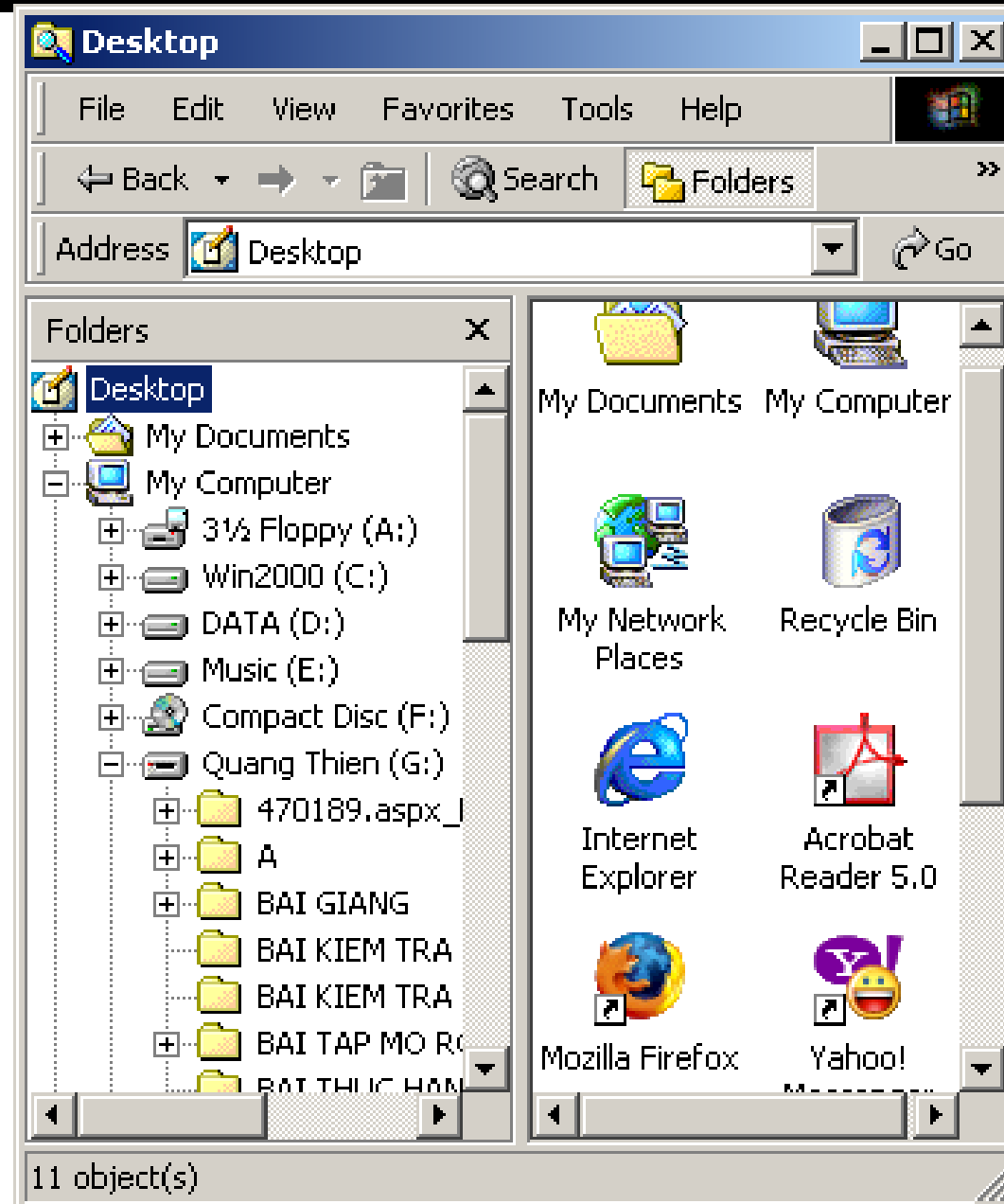


5/ Khái niệm thư mục (tt)







-  Biểu tượng của thư mục là màu vàng.
-  Thư mục không có phần mở rộng.
-  Các ổ đĩa trong cửa sổ My Computer là các thư mục gốc quản lý các thư mục và tập tin.
-  Thư mục con là thư mục nằm trong thư mục khác.
-  Thư mục hiện hành là thư mục đang làm việc.

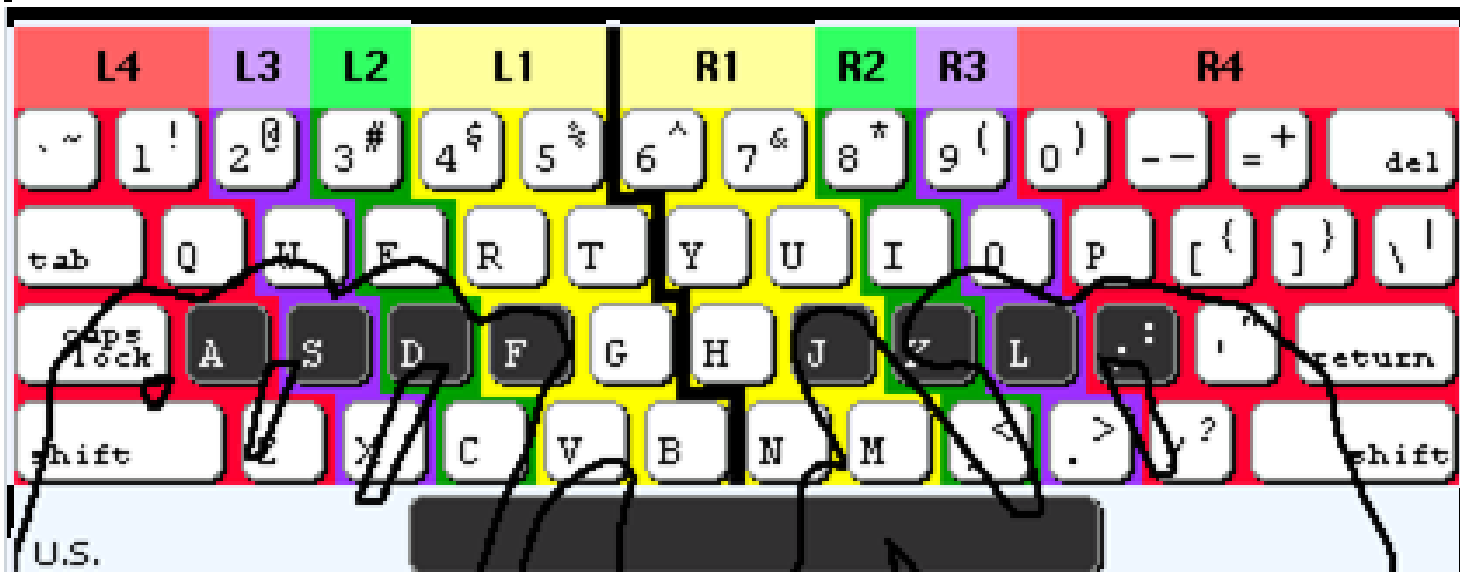
6/ Khái niệm lổi tắt (Shortcut)

- Shortcut là một tập tin đặc biệt chỉ chứa đường đến một tập tin, hoặc một thư mục.
- Shortcut dùng để truy cập nhanh đến một tập tin, thư mục.
- Biểu tượng của lổi tắt luôn có hình mũi tên đặc trưng đi kèm.



7/ Các phím cơ bản trên bàn phím

-  **Space bar (dài nhất):** tạo 1 ký tự trống.
-  **← (Back Space):** xóa về trước 1 ký tự.
-  **Delete:** xóa về sau 1 ký tự (nằm bên phải con trỏ).
-  **Windows:** mở Start Menu.
-  **Caps Lock:** bật/tắt chế độ gõ chữ hoa.
-  **Shift:** gõ phím phụ (ký tự trên đầu) trên phím 2 chức năng.







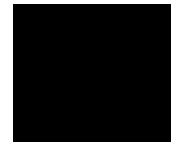
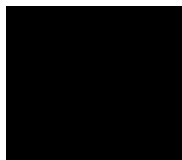
8/ Một số chức năng của chuột

- 🖱️ **Nhấn đơn:** chọn 1 đối tượng (nút bên trái), thực hiện 1 lệnh trên menu.
- 🖱️ **Nhấn đúp:** mở một cửa sổ (nút bên trái)
- 🖱️ **Nhấn phải:** hiện menu ngữ cảnh của các đối tượng.
- 🖱️ **Bi lăn:** cuộn màn hình làm việc.



Các dạng con trỏ chuột

-  **Mũi tên trắng:** trạng thái bình thường.
-  **Đồng hồ cát:** Đang bận thực hiện một lệnh nào đó.
-  **Mũi tên 2 chiều (màu đen):** thay đổi kích thước.
-  **Mũi tên 4 chiều (màu đen):** chọn, di chuyển vị trí.



9/ Khởi động & Tắt máy

- **Khởi động:** Nhấn nút Power (lớn nhất) trên thùng máy.
- **Tắt máy:** Vào Start – Shutdown (Turn Off). Chọn Shutdown (Turn Off) trong hộp thoại.



Câu hỏi và bài tập

Câu 1: Phát biểu nào sau đây không chính xác về tên các ổ đĩa trong My Computer.

- a) Đĩa mềm luôn có tên là A:
- b) Đĩa CD luôn có tên là E:
- c) Đĩa cứng luôn bắt đầu từ C:
- d) Tên B: Không dùng để đặt tên cho các ổ cứng, USB.

Câu 2: Tên tiếng Việt nào sau không trùng với tên tiếng Anh dùng chỉ các thành phần một Window.

- a) **Title bar:** thanh tiêu đề.
- b) **Tools bar:** thanh thực đơn lệnh.
- c) **Scroll bar:** thanh cuộn.
- d) **Status bar:** thanh trạng thái

Câu hỏi và bài tập (tt)

Câu 3: Đối tượng nào trong các đối tượng sau trực tiếp chứa dữ liệu của máy tính.

- a) Thư mục - folder.
- b) Ổ đĩa – drive
- c) Tập tin – file.
- d) Bộ vi xử lý - CPU.

Câu 4: HĐH Windows lưu trữ trên thiết bị nào trong các thiết bị sau?

- a) RAM
- b) ROM
- c) Đĩa mềm
- d) Đĩa cứng

Bài 2

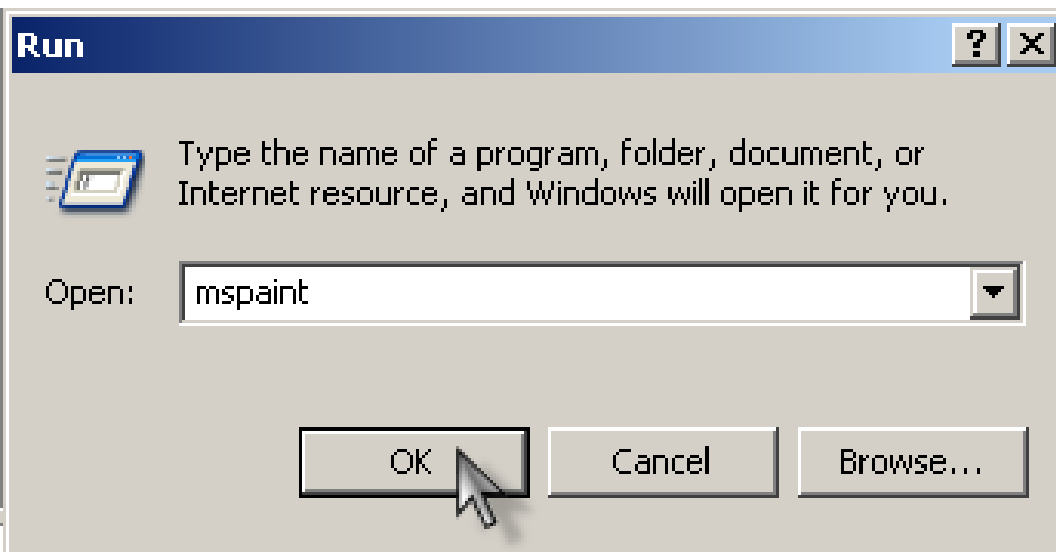
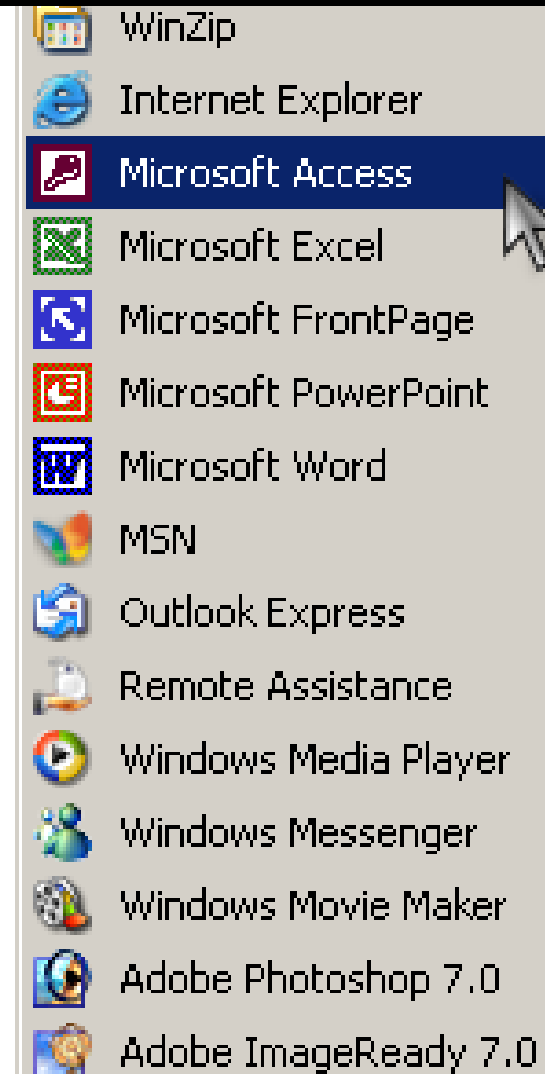
THAO TÁC VỚI CỬA SỐ

1/ Khởi động các ứng dụng

Trên màn hình Dekstop: Nhấn đúp vào các biểu tượng.

Trên Start Menu: Vào menu Programs. Kích chuột vào ứng dụng.

Cửa sổ Run: Nhập tên của ứng dụng cần chạy. Nhấn OK. Ví dụ: winword, Excel, notepad, calc, mspaint.

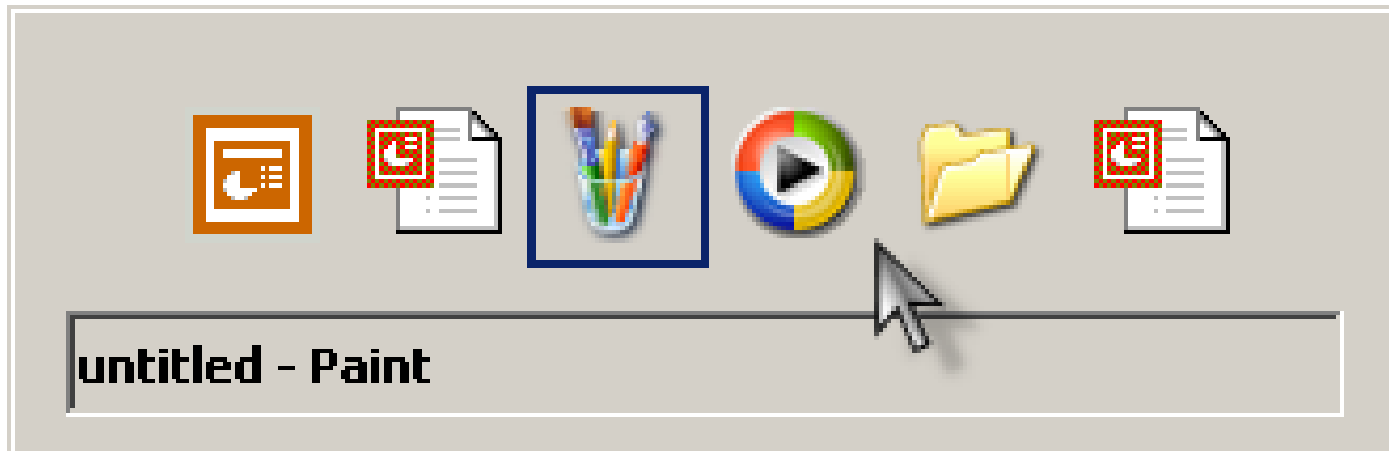


2/ Chuyển đổi qua lại giữa các cửa sổ

- Kích chuột vào tiêu đề cửa sổ trên thanh tác vụ (Taskbar)

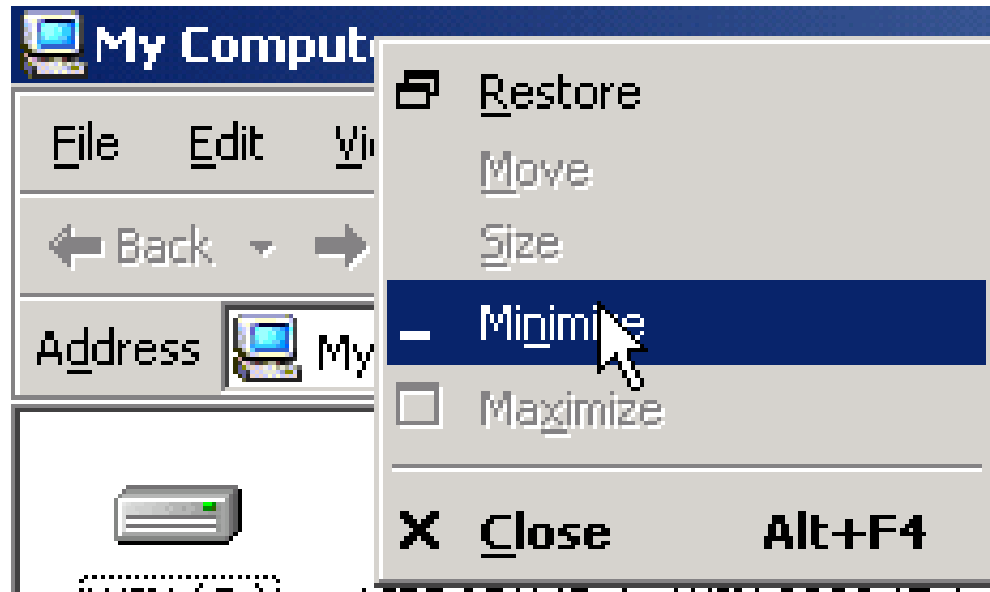


- Dùng tổ hợp phím Alt + Tab.



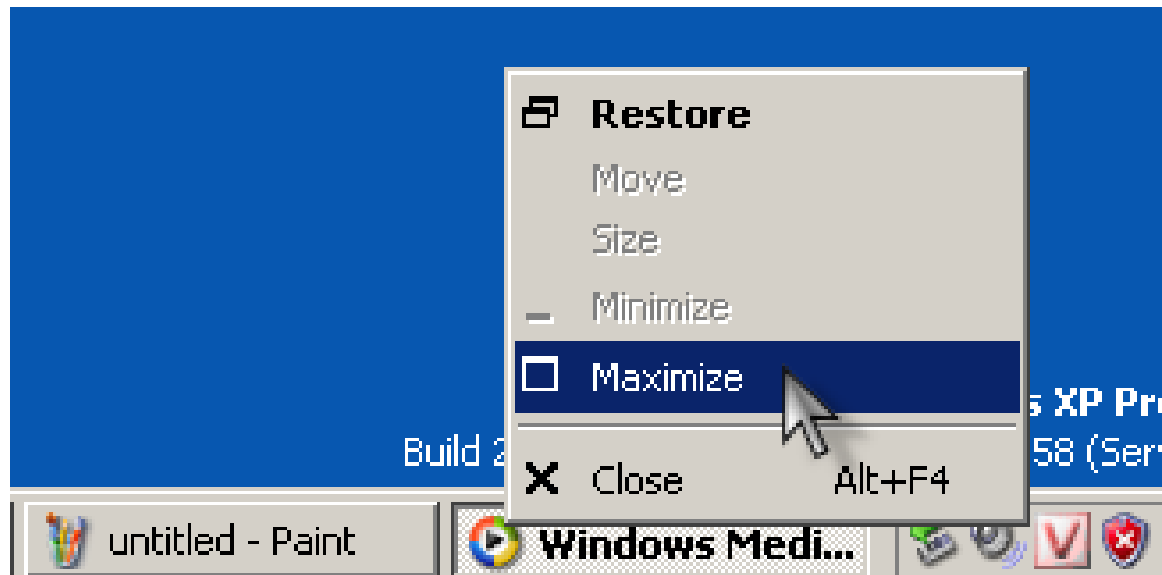
3/ Thu nhỏ cửa sổ

- Nhấn nút _ (Minimize) trên thanh tiêu đề của cửa sổ.
- Hoặc kích phải trên tiêu đề cửa sổ. Chọn Minimize.
- Windows + D: thu nhỏ tất cả các cửa sổ.



4/ Phóng to cửa sổ

- Nhấn nút Maximize trên thanh tiêu đề (Title bar).
- Kích chuột phải trên thanh tiêu đề. Chọn Maximize.
- Hoặc kích đúp chuột vào thanh tiêu đề.



5/ Đóng cửa sổ

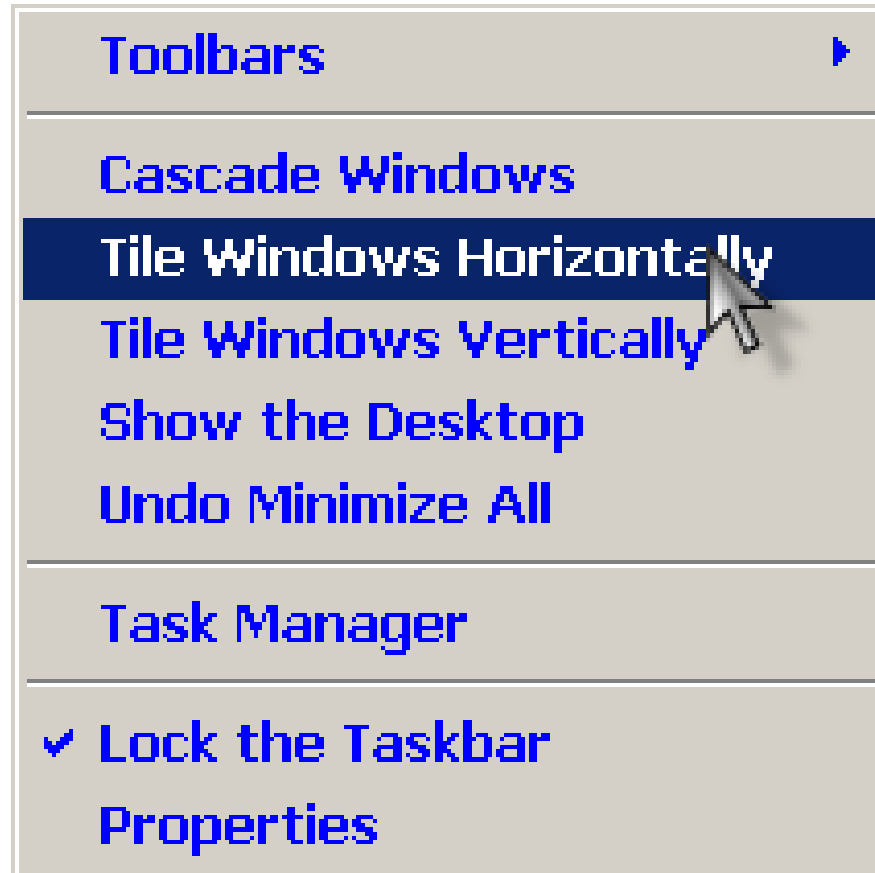
- Nhấn nút Close (X) trên thanh tiêu đề.
- Nhấn tổ hợp phím Alt + F4.
- Kích phải trên tiêu đề cửa sổ. Chọn Close.
- Vào menu File – chọn Close.

6/ Sử dụng phím tắt

- Nhấn phím F10 để kích hoạt thực đơn lệnh của ứng dụng bất kỳ.
- Dùng tổ hợp phím **Alt + Ký tự gạch chân** trên menu, nút nhấn, ... để thực hiện lệnh thay cho chuột.
- Trên menu: dùng ký tự gạch chân để thực hiện lệnh.

7/ Sắp xếp các cửa sổ

- ❑ Mở nhiều cửa sổ.
- ❑ Nhấp phải chuột trên thanh Taskbar và chọn một trong các kiểu sắp xếp: Sắp so le, sắp theo hàng hoặc sắp theo cột.



Bài 3

LÀM VIỆC VỚI MÀN HÌNH NỀN DESKTOP

1/ Cấu trúc Desktop

- Các biểu tượng mặc định:
 - **My Documents:** Thư mục dành riêng của người dùng.
 - **My Computers:** Quản lý các ổ đĩa.
 - **My Networks Place:** Truy cập tài nguyên trên mạng nội bộ.
 - **Recycle Bin:** Thùng rác
 - **Internet Explorer:** Truy cập Internet.
- Thanh tác vụ (**Taskbar**): quản lý các cửa sổ.
- Start Menu:** Chứa các menu liên kết đến các ứng dụng, các tiện ích khác của HĐH...



My Documents



My Computer



My Network
Places



Recycle Bin

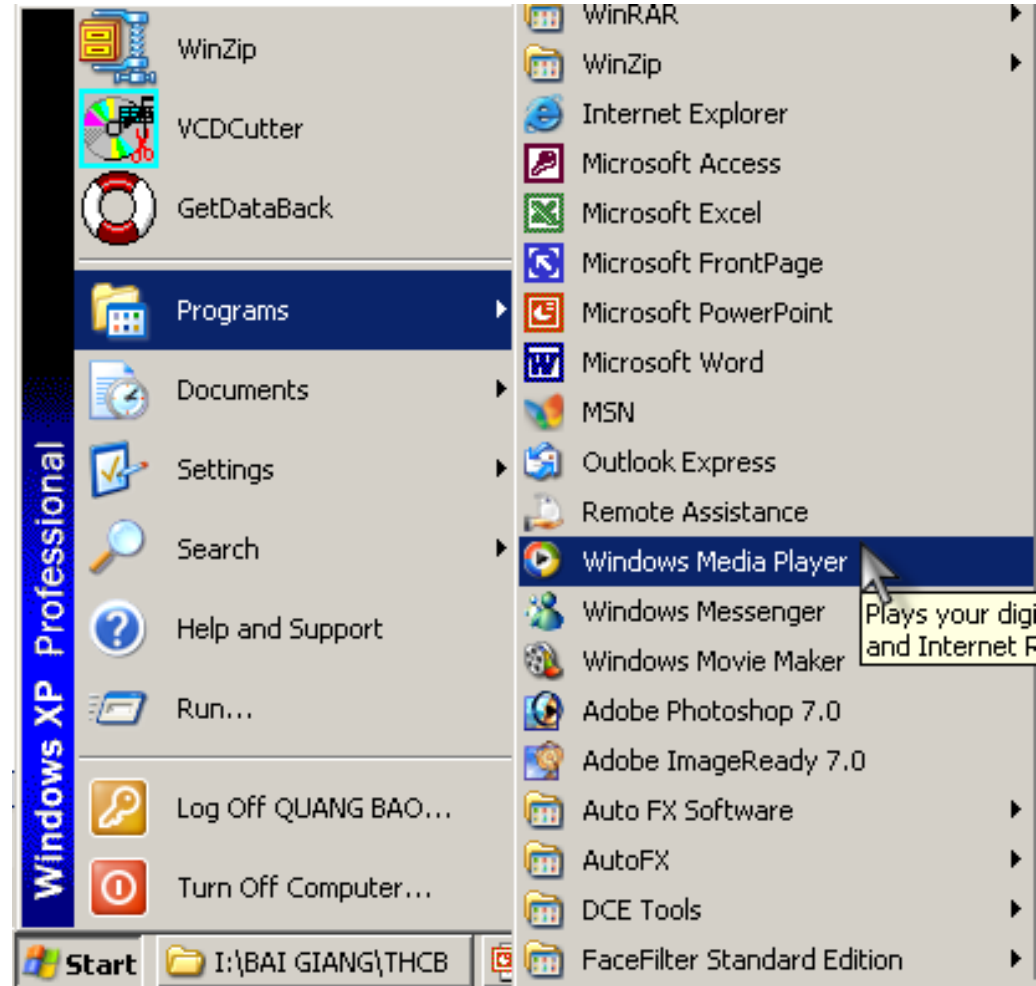


Internet
Explorer

2/ Sử dụng Start Menu

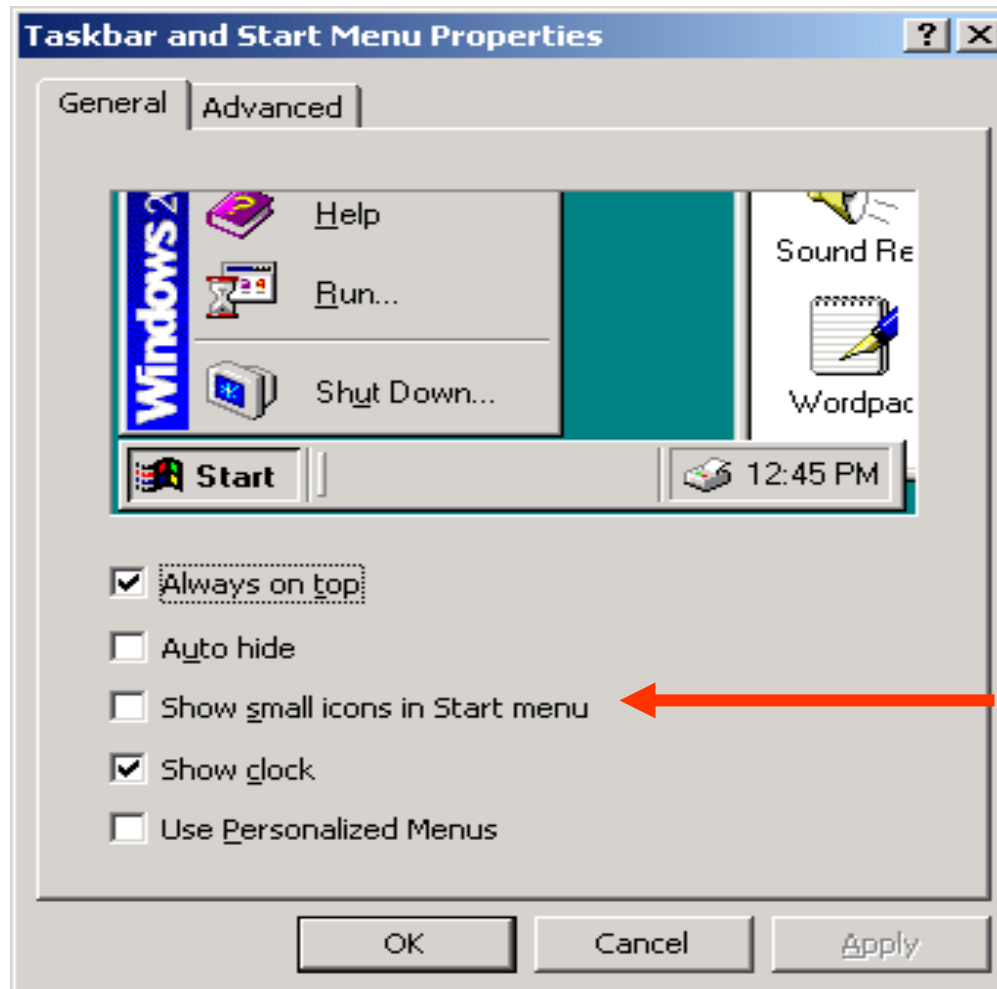
2.1/ Thao tác với Start Menu:

- Nhấn phím Windows, hoặc kích chuột vào nút Start để kích hoạt.
- Di chuyển chuột (hoặc phím mũi tên) để chọn.
- Kích chuột (hoặc nhấn Enter) để thực hiện lệnh.






2.2/ Tùy biến Start Menu

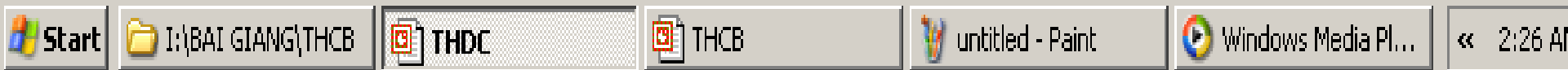
- Click phải trên Taskbar. Chọn **Properties**.
- Show small icon in Start Menu**: hiển thị các biểu tượng dạng nhỏ nhất trên Start Menu.



3/ Sử dụng Taskbar

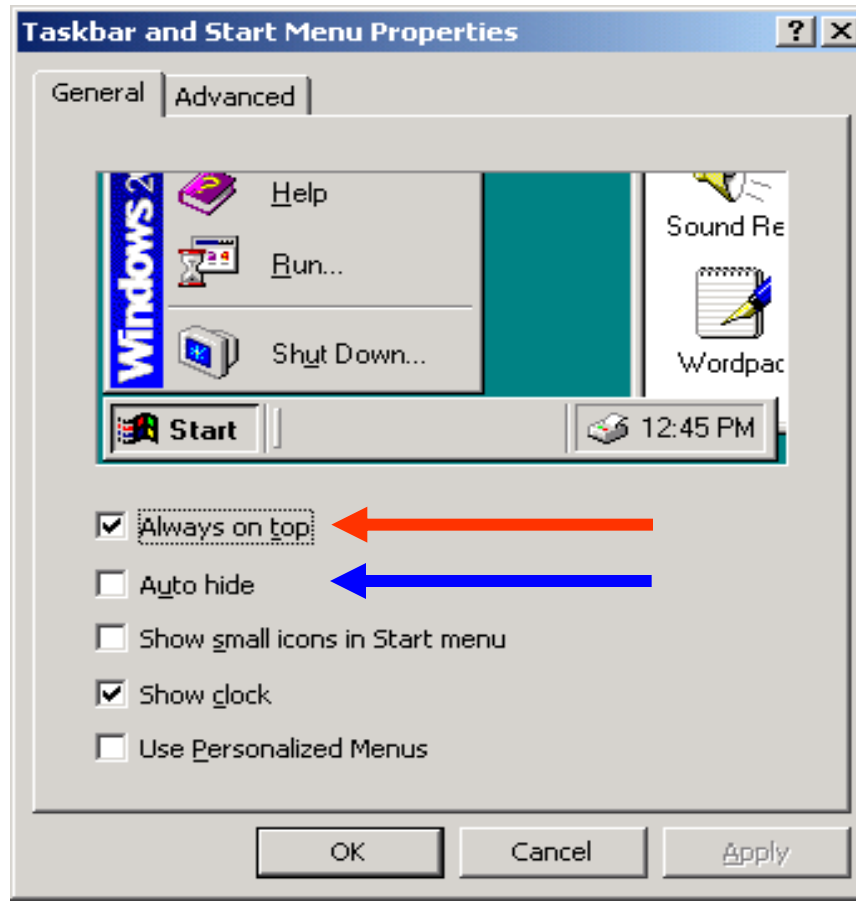


-  Taskbar dùng để quản lý các cửa sổ (ứng dụng) đang hoạt động trên máy.
-  Taskbar có thể nằm ở 4 biên của màn hình Desktop.
-  Kích thước của Taskbar có thể thay đổi bởi người sử dụng.



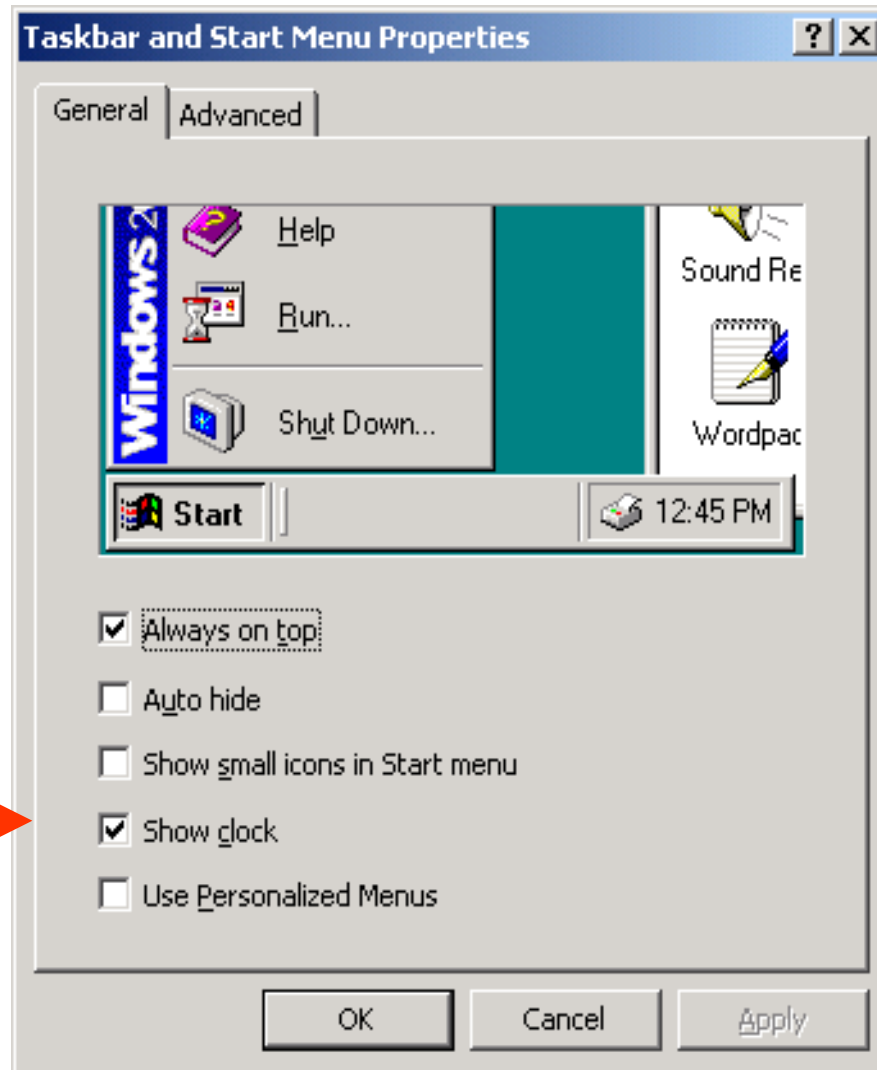
3.1/ Tùy biến Taskbar

- Kích phải trên Taskbar. Chọn **Properties**.
- **Auto Hide**: Chế độ tự động ẩn Taskbar khi làm việc với cửa sổ.
- **Always top**: Luôn hiển thị Taskbar lên trên cùng.



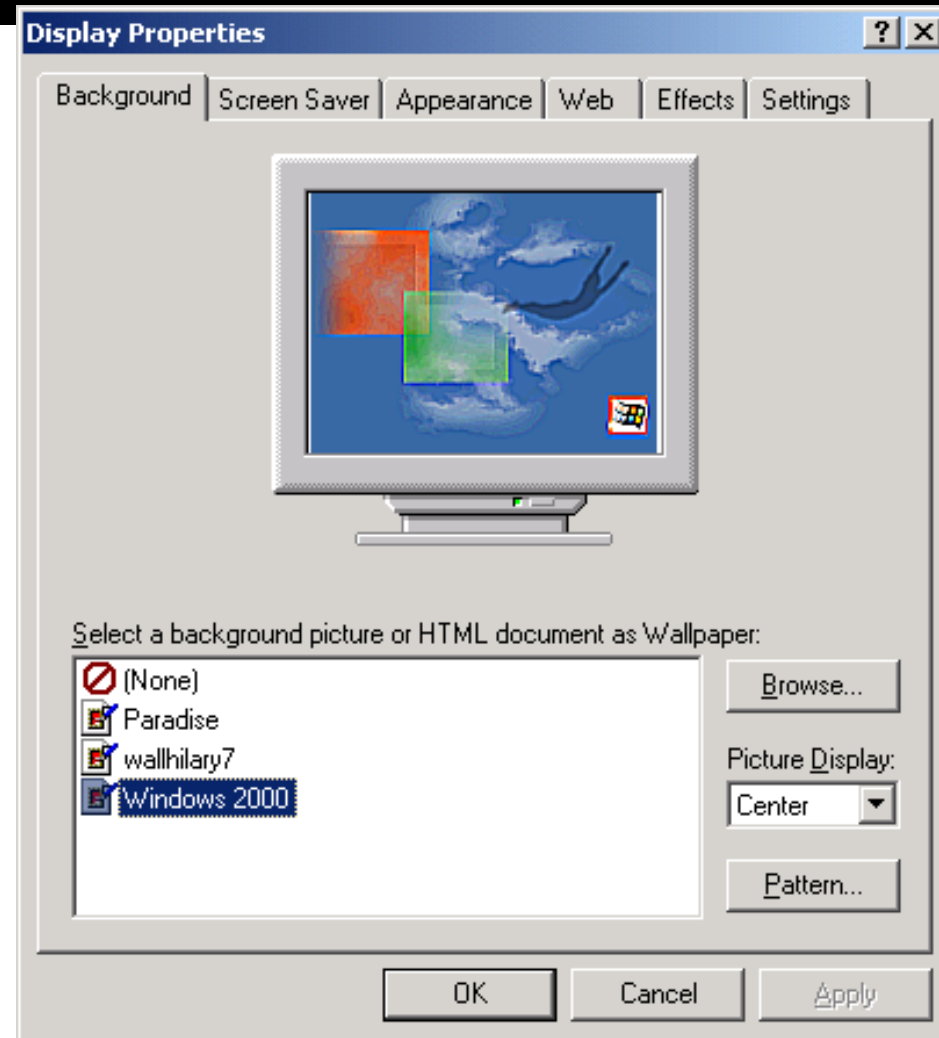
3.2/ Tùy biến khay hệ thống

- ❖ Kích phải trên khay hệ thống. Chọn **Properties**.
- ❖ **Show Clock**: hiển thị đồng hồ.



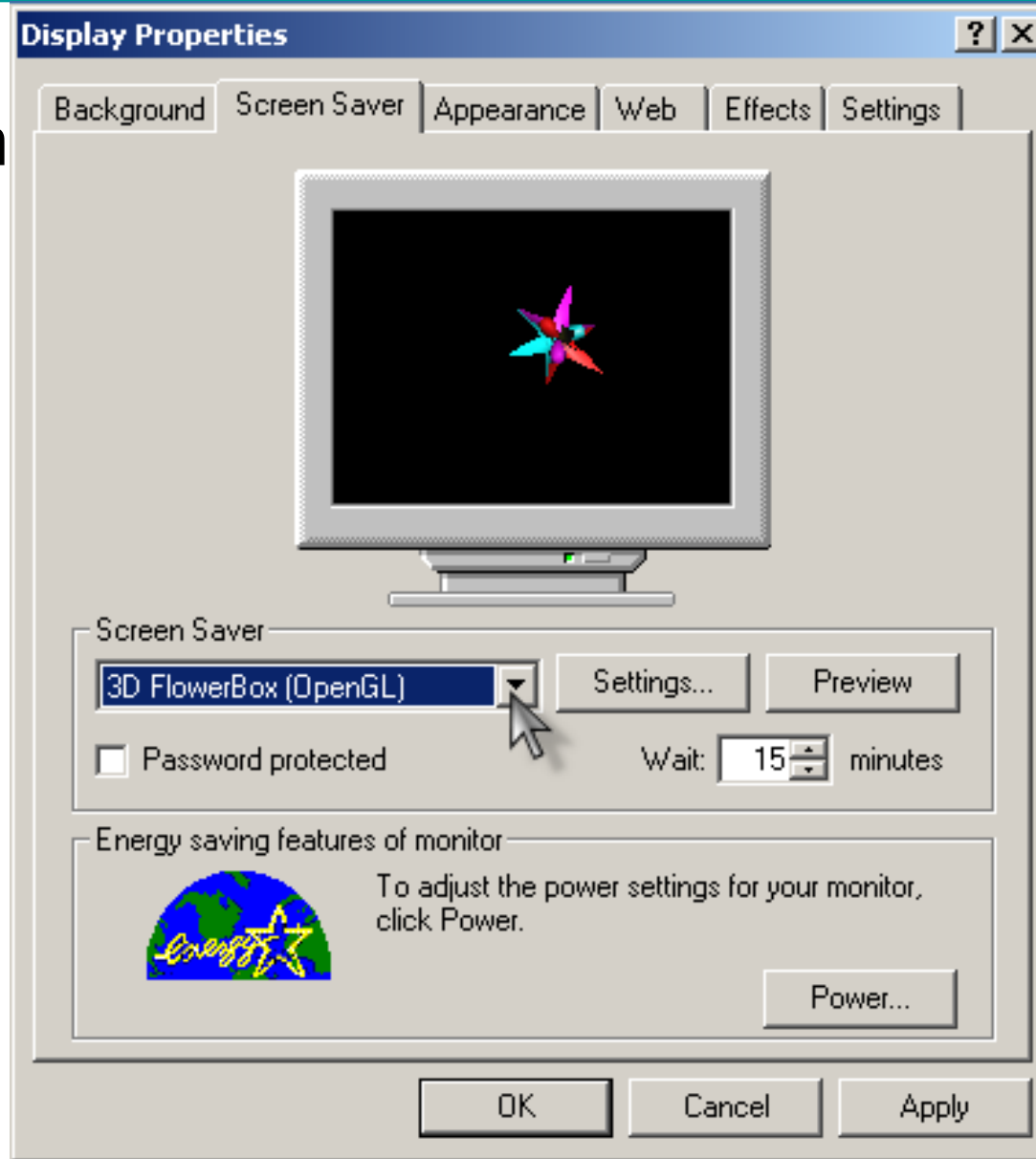
4/ Thay đổi màn hình nền

- Kích phải tại khoảng trống trên Desktop. Chọn **Properties**.
- Chọn thẻ Background (Desktop)
- Chọn 1 hình trong danh sách bên dưới.
- Nhấn nút **Browse** để tìm chọn hình lưu ở nơi khác.




5/ Chế độ bảo vệ màn hình (Screen Saver)


- Kích phải tại khoảng trống Desktop. Chọn Properties.
- Chọn thẻ Screen Saver. Chọn một kiểu trong danh sách.
- Thiết lập thời gian đợi trong mục **Wait**
- **Settings:** thay đổi theo ý người sử dụng.
- **Preview:** xem trước.

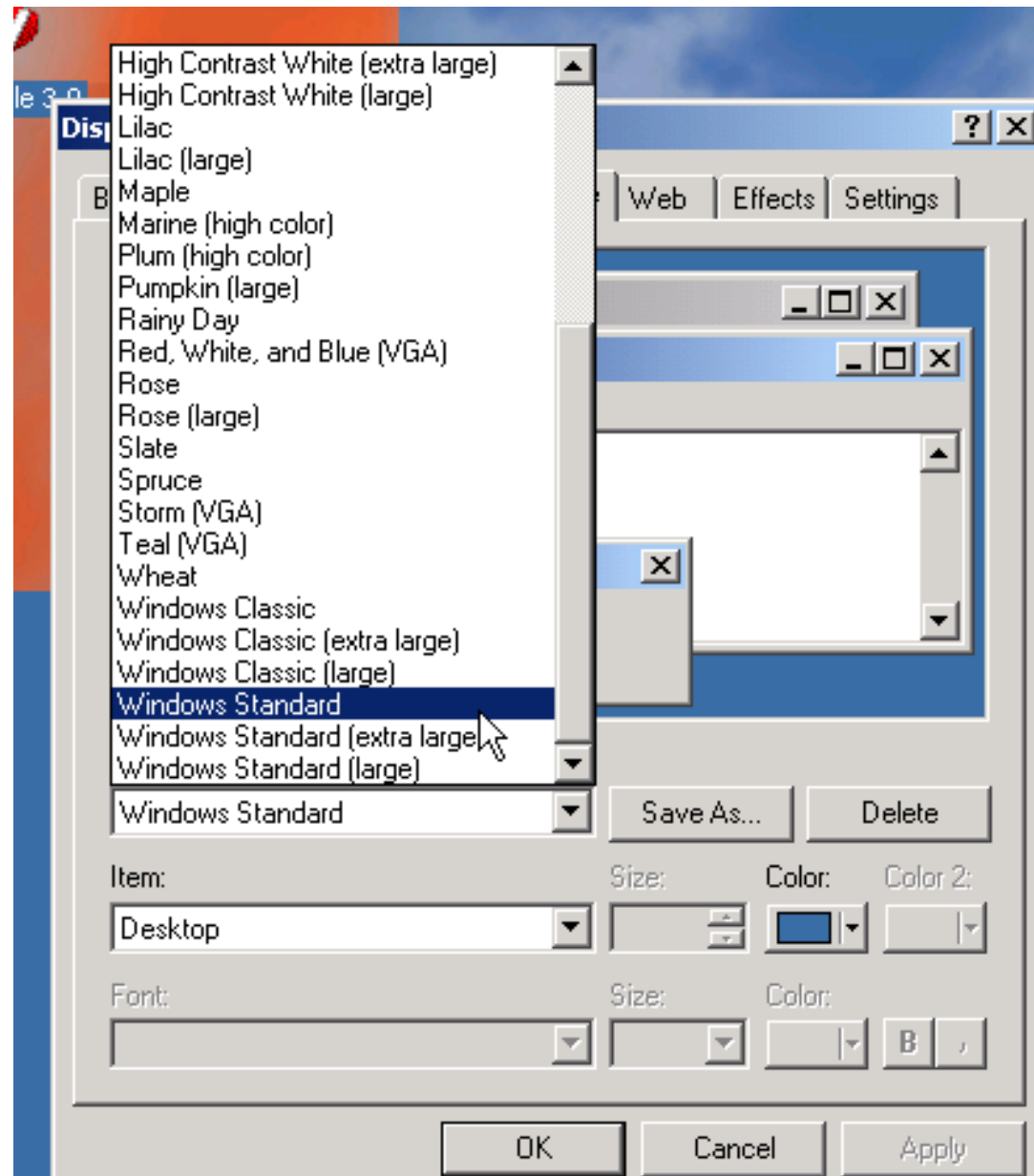


6/ Thay đổi giao diện Windows

 Kích phải chuột tại khoảng trống trên Desktop. Chọn **Properties**. Chọn thẻ **Appearance**.

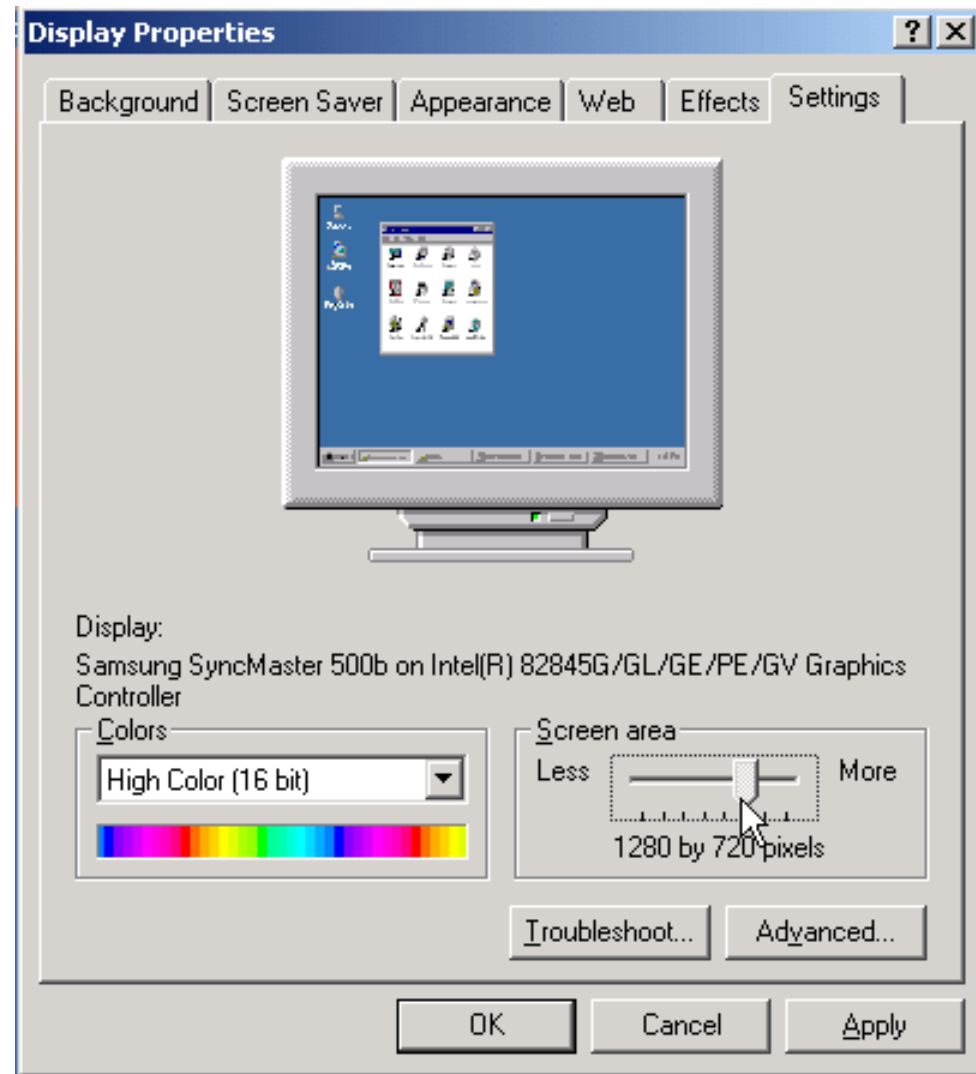
 Chọn một mẫu trong **Scheme**.

 Trả về kiểu mặc định: Chọn mẫu **Windows Standard**.



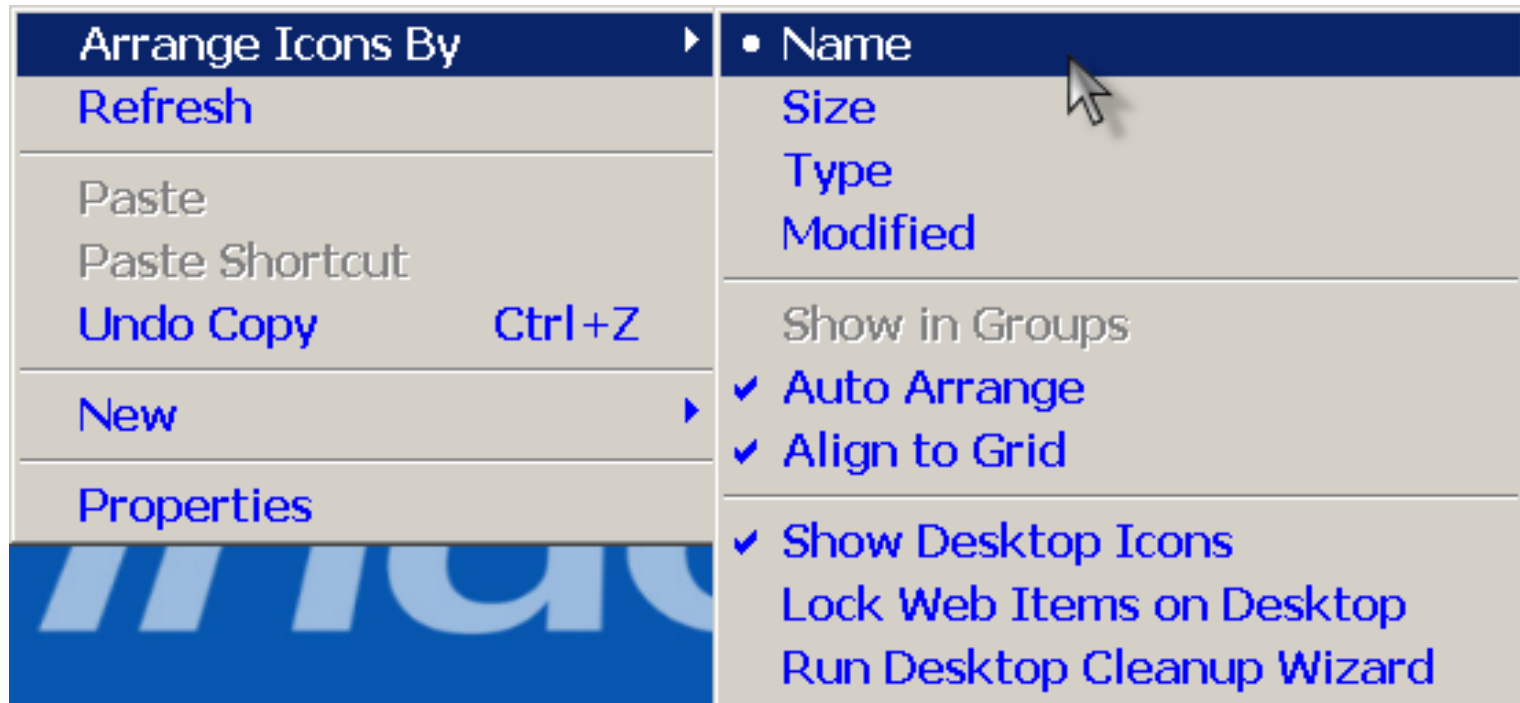
7/ Thiết lập độ phân giải

- ✍ Trong **Display Properties**. Chọn thẻ **Settings**.
- ✍ Chọn số lượng màu trong mục **Colors**. Mặc định là **High color 24bit**.
- ✍ Chọn độ phân giải trong mục **Screen Area**. Chuẩn là 800 X 600.
- ✍ Chọn độ phân giải lớn thì chữ hiển thị sẽ nhỏ và ngược lại.



8/ Sắp xếp các biểu tượng trên Desktop

- ❑ Kích phải tại vị trí trống trên Desktop.
- ❑ Chọn Arrange Icons.
 - Auto Range: Sắp xếp tự động.
 - By Name: Sắp theo tên.
 - By Type: Theo kiểu tập tin, thư mục.
 - By Size: Theo dung lượng của biểu tượng.
 - By Date: Sắp theo ngày tạo lập.

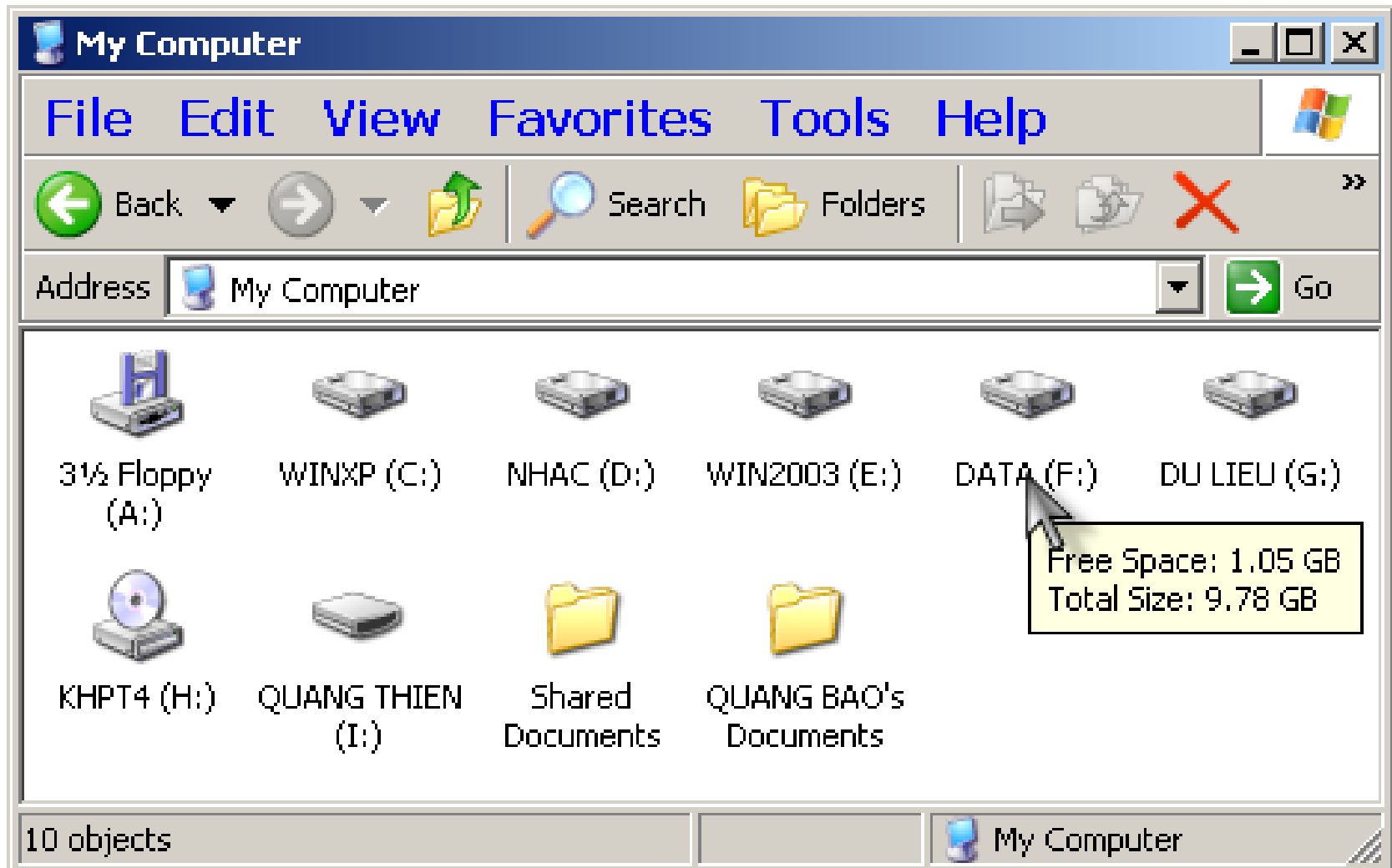


BÀI 4

QUẢN LÝ TỆP TIN VÀ THƯ MỤC VỚI MY COMPUTER

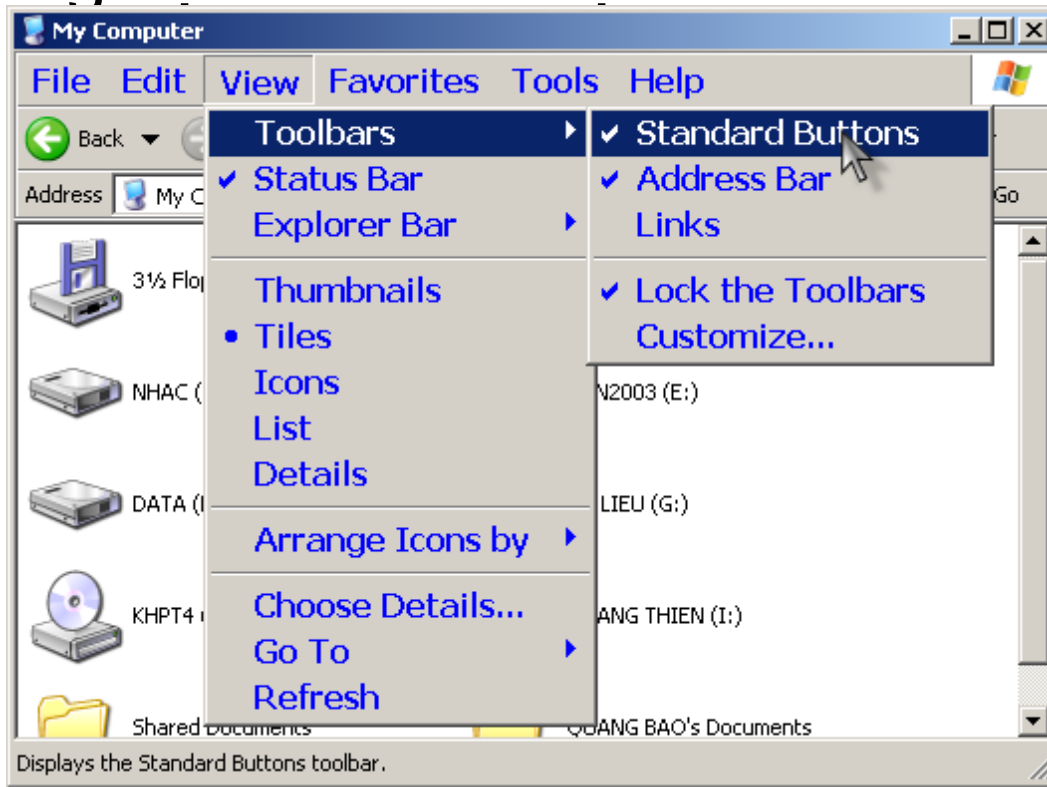
I/ Khởi động

❑ Nhấn đúp vào biểu tượng **My Computer** trên màn hình Desktop.



II/ Hiện thị thanh công cụ

- ❑ Windows Explorer cần 2 thanh công cụ:
 - **Standard Buttons:** Chứa các lệnh từ menu giúp thao tác nhanh với tập tin và thư mục.
 - **Address bar:** hiển thị đường dẫn của tập tin, thư mục.
- ❑ Vào menu **Views – Toolbars**. Đánh dấu vào các thanh công cụ cần hiển thị.



III/ Làm việc với tệp tin & thư mục

1/ Tạo mới tệp tin (NEW):

Mục đích: Chứa nội dung văn bản, hình họa, video...

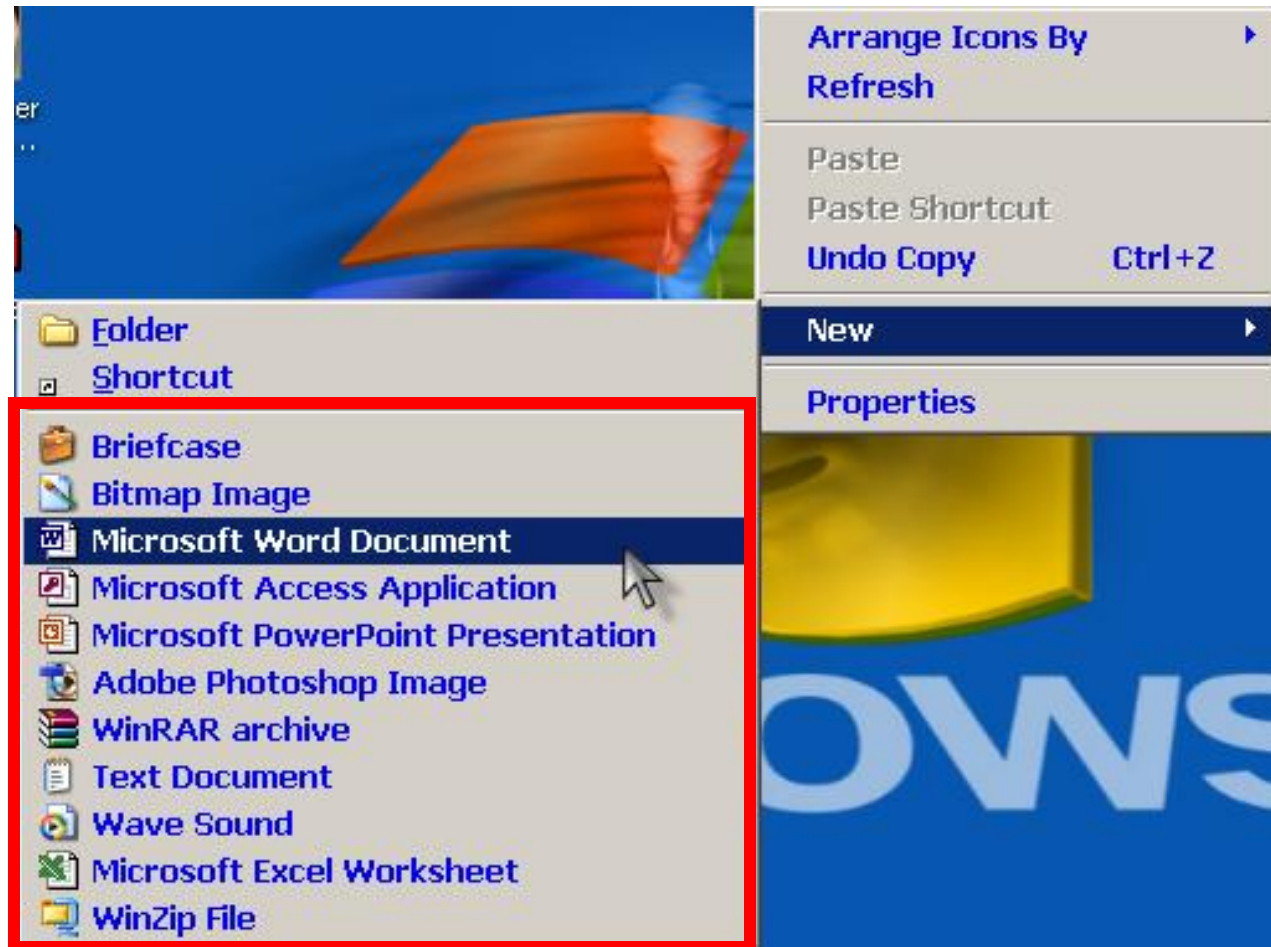
Thực hiện:

➤ Chọn nơi lưu tệp tin (đĩa, thư mục).

➤ Kích phải chuột tại vị trí trống.

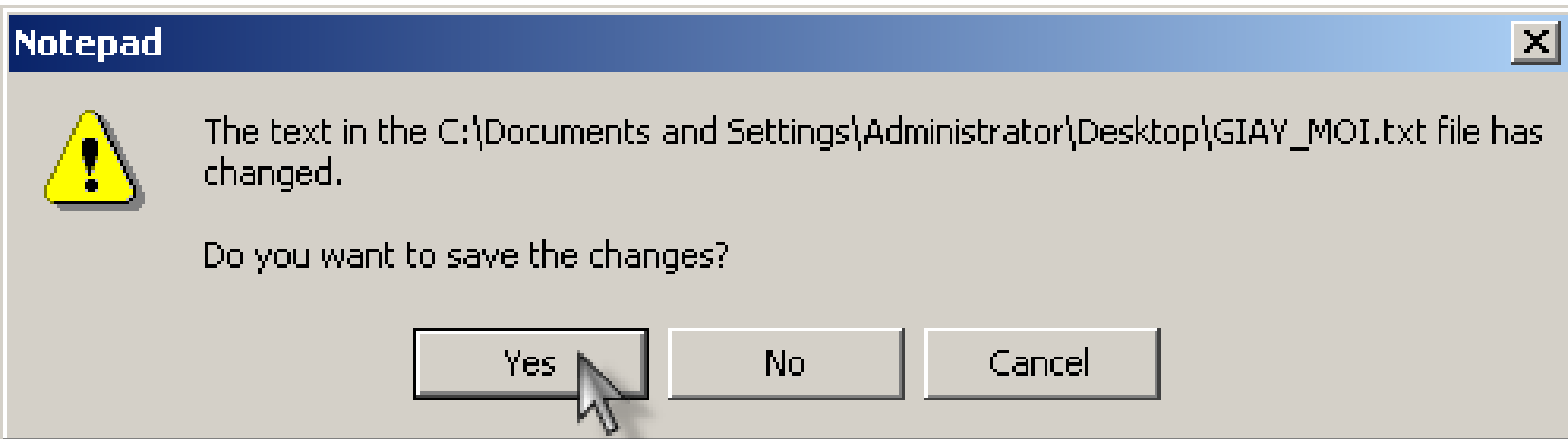
➤ Chọn New – Kích chọn 1 kiểu tệp tin trong danh sách.

➤ Đặt tên cho tệp tin (không được trùng tên với tệp tin đã có).



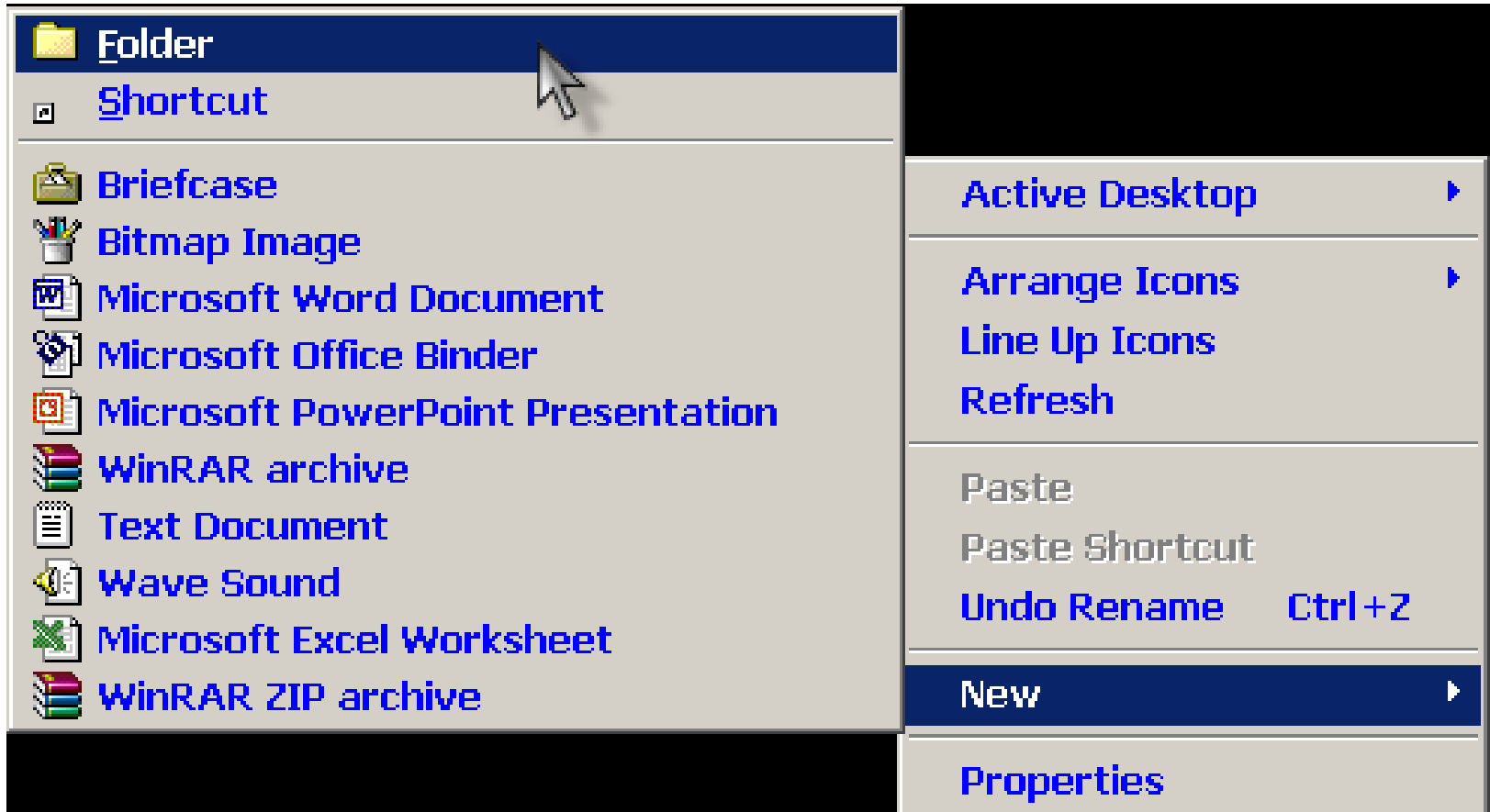
2/ Nhập nội dung cho tập tin

- Nhấn đúp vào tên tập tin.
- Nhập nội dung bất kỳ.
- Đóng cửa sổ của tập tin lại.
- Chọn Yes để lưu nội dung tập tin lại.



3/ Tạo mới 1 thư mục (New)

- Chọn nơi đặt thư mục.
- Kích phải chuột tại vị trí trống.
- Chọn New – Folder.
- Nhập tên cho thư mục vừa tạo.



4/ Tạo mới 1 lối tắt

□ Cách 1:

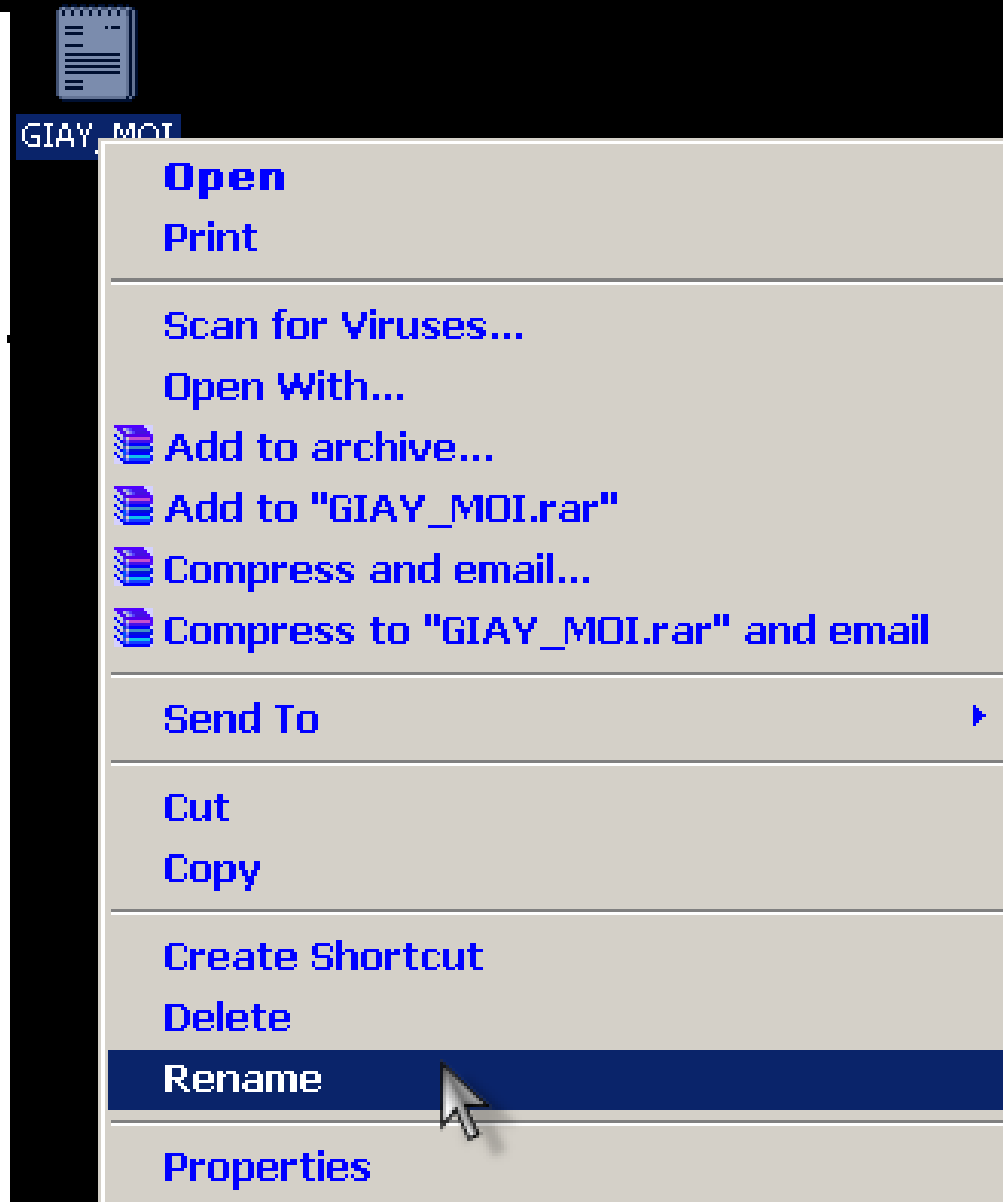
- Truy cập vào nơi chứa ứng dụng.
- Kích phải chuột trên đối tượng cần tạo lối tắt.
Chọn **Send to – Desktop (Create Shortcut)**

□ Cách 2:

- Kích phải tại vị trí trống trên **Desktop**. Chọn **New – Shortcut**. Nhấn nút **Browse** chọn đường dẫn của tập tin, thư mục cần tạo lối tắt. Nhấn **OK**. Nhấn **Next**.
- Nhập tên cho **shortcut**. Nhấn **Finish**.

5/ Đổi tên (Rename)

- ❑ **Cách 1:** Kích phải chuột trên đối tượng cần đổi tên. Chọn Rename. Nhập tên mới.
- ❑ **Cách 2:** Kích chọn đối tượng. Nhấn phím F2. Nhập tên mới.

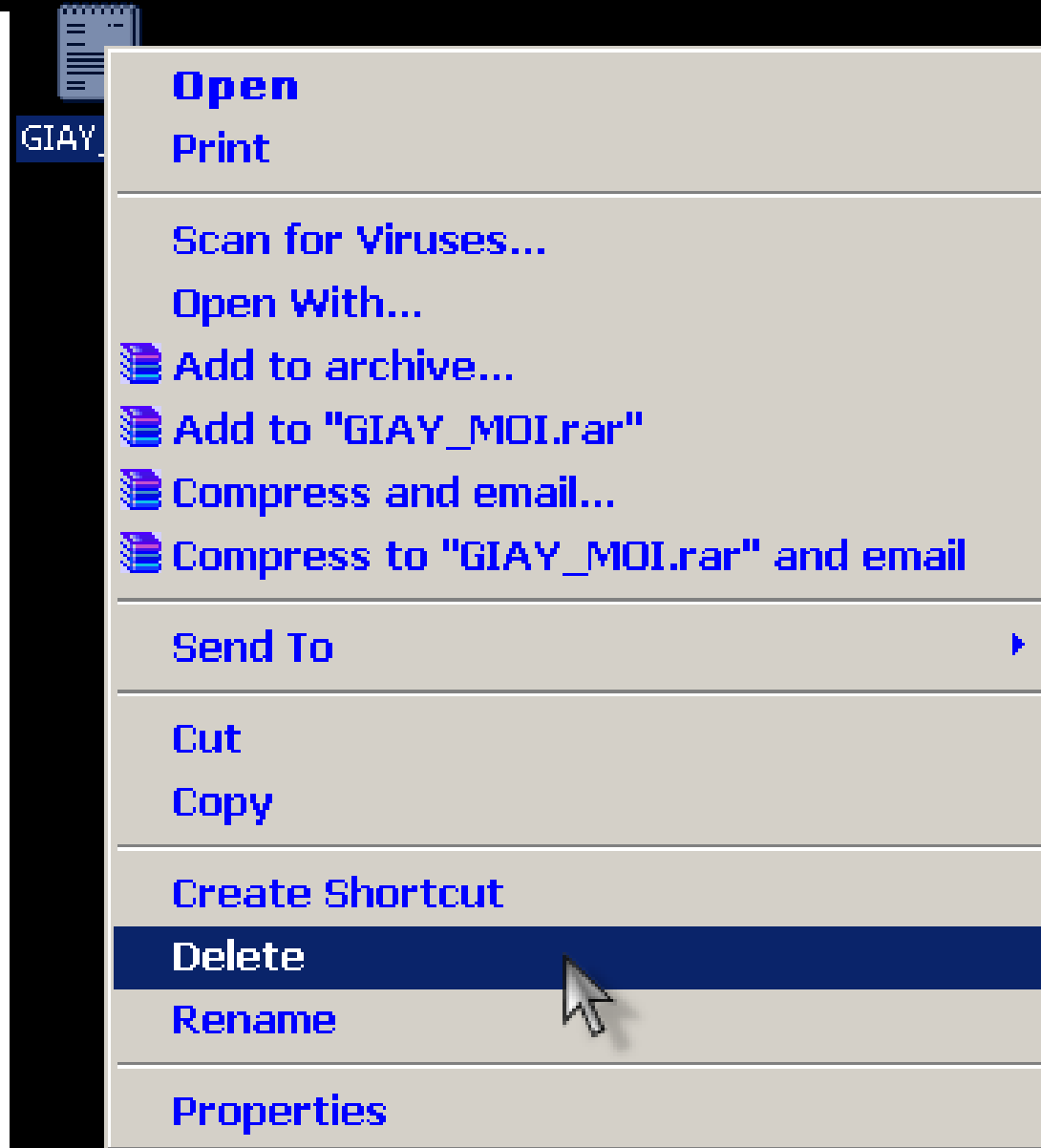


6/ Xóa (Delete)

❑ **Cách 1:** Kích phải chuột trên đối tượng cần xóa. Chọn Delete. Nhấn Yes.

❑ **Cách 2:** Kích chuột chọn đối tượng cần xóa. Nhấn phím Delete. Nhấn Yes.

💾 **Lưu ý:** Để xóa hẳn đối tượng khỏi mà không cho vào trong thùng rác thì nhấn giữ phím **Shift** khi xóa



7/ Sao chép (Copy) và di chuyển (Move)

☐ Sao chép (Copy):

- Kích phải chuột trên đối tượng cần sao chép. Chọn **Copy** – tức sao chép. (hoặc nhấn **Ctrl + C**).
- Truy cập vào thư mục - nơi cần copy đến. Kích phải chọn **Paste** – tức dán. (hoặc nhấn **Ctrl + V**)

☐ Di chuyển (Move):

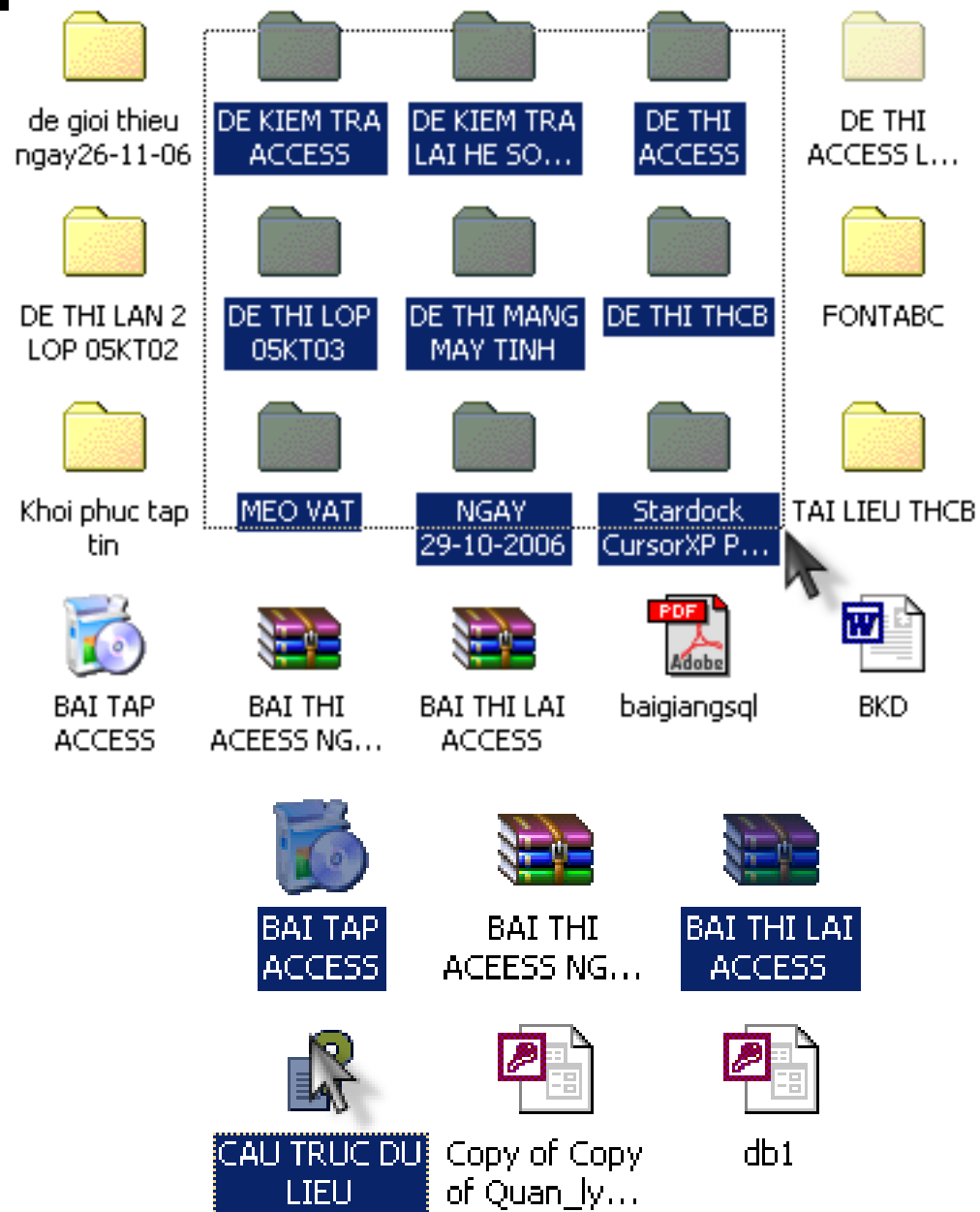
- Kích phải chuột trên đối tượng cần di chuyển. Chọn **Cut** – tức cắt (hoặc **Ctrl + X**).
- Truy cập vào thư mục - nơi cần copy đến. Kích phải chọn **Paste** – tức dán. (Hoặc **Ctrl + V**)

BÀI TẬP

- Trong ổ đĩa C:\ tạo 2 thư mục tx_QuangNgai và tp_DaNang
- Trong thư mục tx_QuangNgai tạo 2 thư mục con là DaiViet và Daotao.
- Trong thư mục DaiViet tạo 2 tệp tin vanban.txt và tệp giaymoi.txt
- Đổi tên thư mục tx_QuangNgai thành thư mục tp_QuangNgai
- Sao chép tất cả các thư mục trong thư mục tp_QuangNgai vào trong thư mục tp_DaNang
- Đổi tên thư mục tp_DaNang thành VietNam
- Xóa thư mục VietNam

8/ Chọn các đối tượng

- ☐ **Chọn nhiều đối tượng gần nhau:** Nhấn giữ Shift + phím mũi tên (Hoặc **Kích giữ chuột và vây vùng hiển thị các đối tượng cần chọn**).
- ☐ **Chọn nhiều đối tượng rời rạc:** Giữ Ctrl + **kích chuột** chọn từng đối tượng cần chọn.
- ☐ **Ctrl + A:** Chọn tất cả các đối tượng.

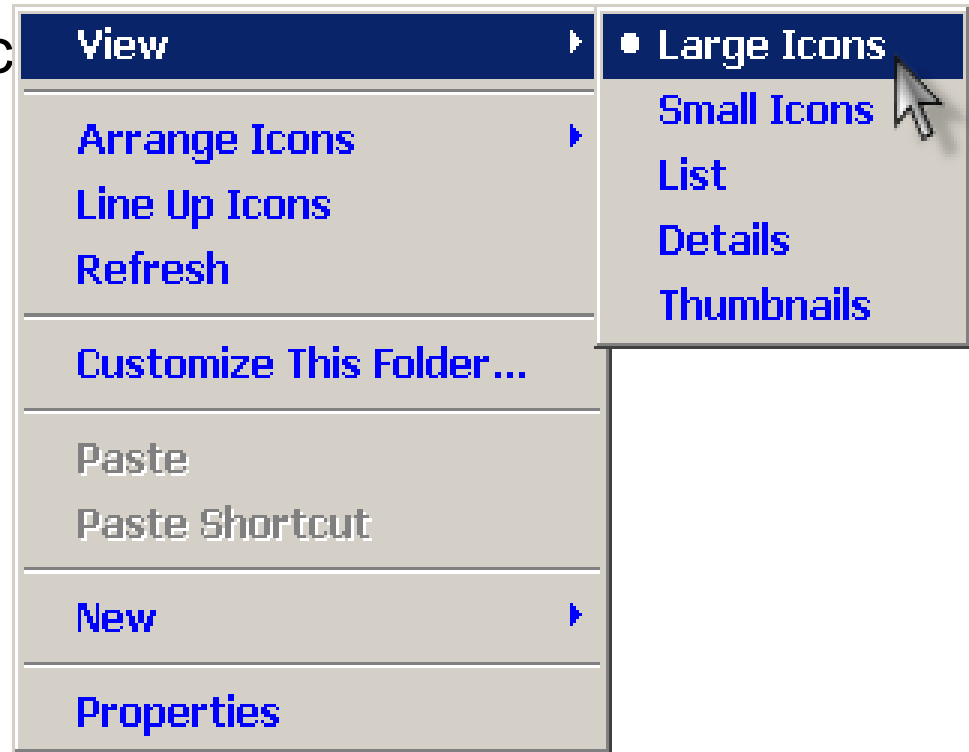


9/ Kiểu hiển thị

☐ Kích phải chuột tại vị trí trống bất kỳ trong thư mục cần đổi kiểu hiển thị.

☐ **Chọn View:**

- **Large Icons:** Biểu tượng phóng to.
- **Small Icons:** Biểu tượng thu nhỏ.
- **List:** Dạng danh sách liệt kê.
- **Details:** Chi tiết (Tên, Ngày giờ, dung lượng...)
- **Thumbnails:** Thu nhỏ nội dung bên trong – dành để xem trước hình ảnh.



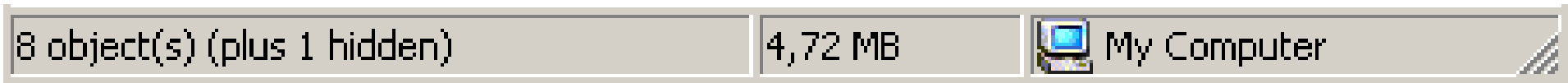
10/ Sử dụng thanh công cụ



- Back:** quay về cửa sổ trước đó.
- Forward:** tiến tới trang trước khi quay về.
- Up:** Lên thư mục cha.
- Searchs:** Ấn/hiện cửa sổ tìm kiếm.
- Folders:** Ấn/hiện cây thư mục.

11/ Sử dụng thanh trạng thái (Status Bar)

- ✎ Vào menu Views – Status Bar để hiển thị thanh trạng thái.
- ✎ Thanh trạng thái cho biết:
 - ✓ Tổng số tập tin & thư mục có trong thư mục hiện tại (Object).
 - ✓ Tổng dung lượng.



12/ Xem thuộc tính các phân vùng ổ cứng

- ✘ Kịch phải trên đĩa chọn Properties. Chọn thẻ **General**.
- ✓ **Used space:** dung lượng đĩa dùng.
- ✓ **Free space:** dung lượng còn trống
- ✓ **Capacity:** Tổng dung lượng đĩa.
- ✓ **Label:** Nhãn đĩa

The screenshot shows the 'Win2000 (C:) Properties' dialog box with the 'General' tab selected. It displays the following information:

- Label: Win2000
- Type: Local Disk
- File system: NTFS
- Used space: 1,483,591,680 bytes (1,38 GB)
- Free space: 4,808,712,192 bytes (4,47 GB)
- Capacity: 6,292,303,872 bytes (5,85 GB)

A pie chart below the statistics shows the distribution of space: a blue slice for used space and a pink slice for free space. The text 'Drive C' is centered below the chart, and a 'Disk Cleanup...' button is located to the right. At the bottom, there are checkboxes for 'Compress drive to save disk space' (unchecked) and 'Allow Indexing Service to index this disk for fast file searching' (checked), along with 'OK', 'Cancel', and 'Apply' buttons.

Category	Value	Human-Readable
Used space	1,483,591,680 bytes	1,38 GB
Free space	4,808,712,192 bytes	4,47 GB
Capacity	6,292,303,872 bytes	5,85 GB

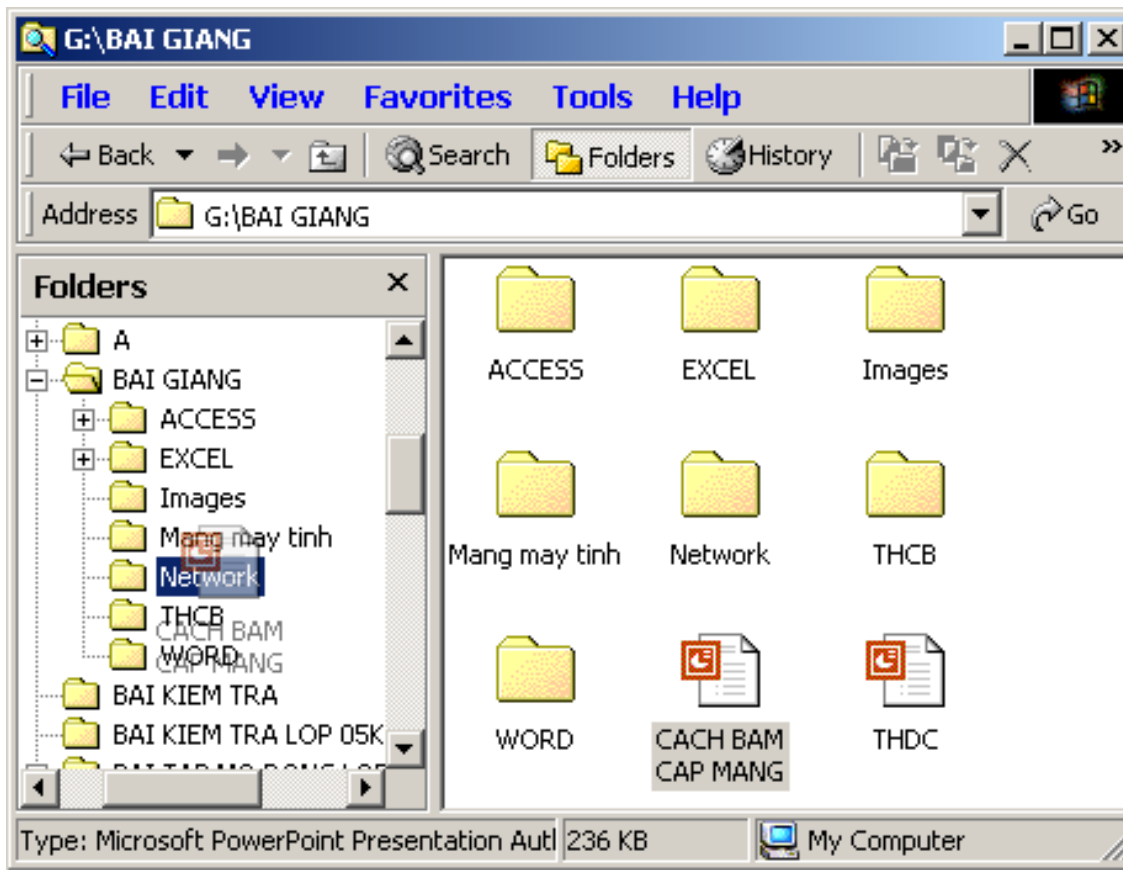
IV/ SỬ DỤNG WINDOWS EXPLORER

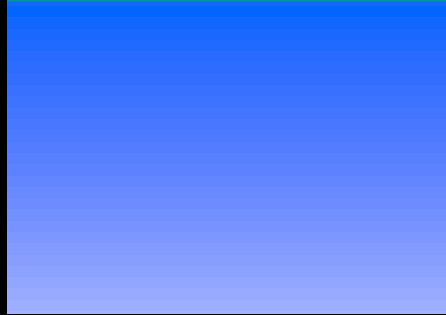
- **Windows Explorer** là một chương trình ứng dụng được tích hợp sẵn trên môi trường Windows – cho phép người sử dụng thao tác đến các tài nguyên trong hệ thống máy tính dưới dạng cấu trúc cây (cấu trúc thừa kế).
- Khởi động Windows Explorer:
 - ✗ Cách 1: Nháy chuột phải lên biểu tượng My Computer, chọn Explore
 - ✗ Cách 2: Vào My Computer, sau đó kích lên nút Folders trên thanh công cụ
 - ✗ Cách 3: Nhấn tổ hợp phím Windows+E

IV/ WINDOWS EXPLORER (tt)

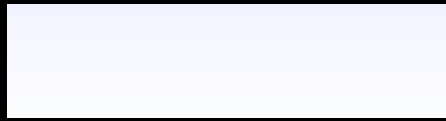
☐ Sử dụng kéo & thả (Drag & Drop):

- Dùng chuột kéo 1 thư mục, tập tin vào thư mục khác để thực hiện lệnh di chuyển.
- Kết hợp phím Ctrl + Kéo thả để thực hiện lệnh sao chép.
- Kéo 1 đối tượng qua ổ đĩa khác tên là thực hiện lệnh sao chép.





BÀI 5



**SỬ DỤNG THÙNG RÁC –
RECYCLE BIN**

RECYCLE BIN

1/ Khái niệm: Recycle Bin (thùng tái chế) dùng để chứa các tập tin thư mục bị xóa.

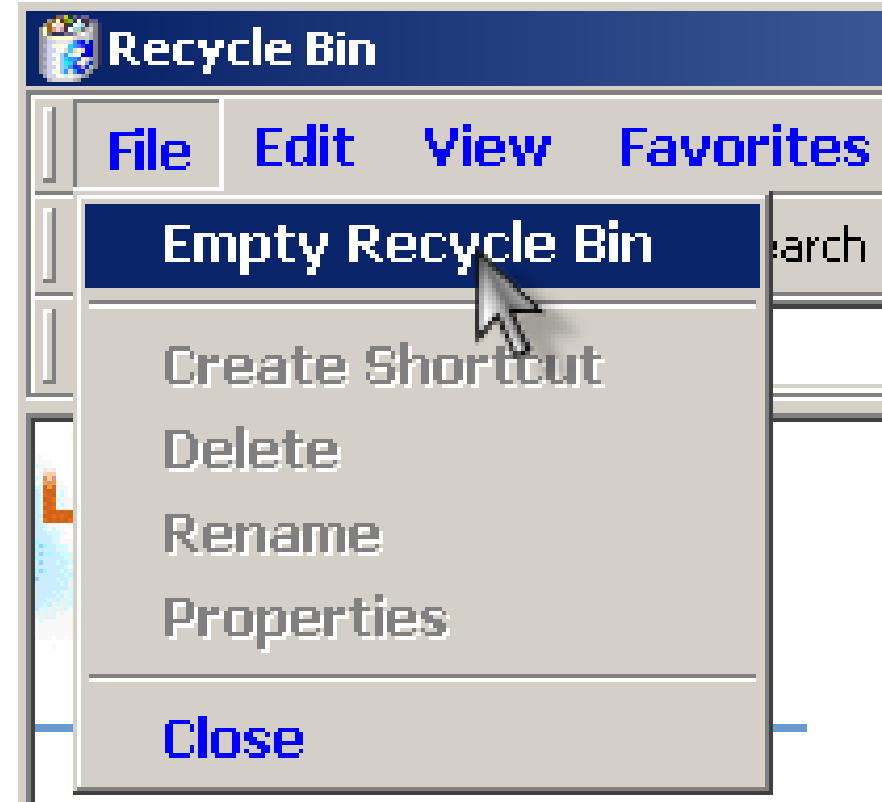
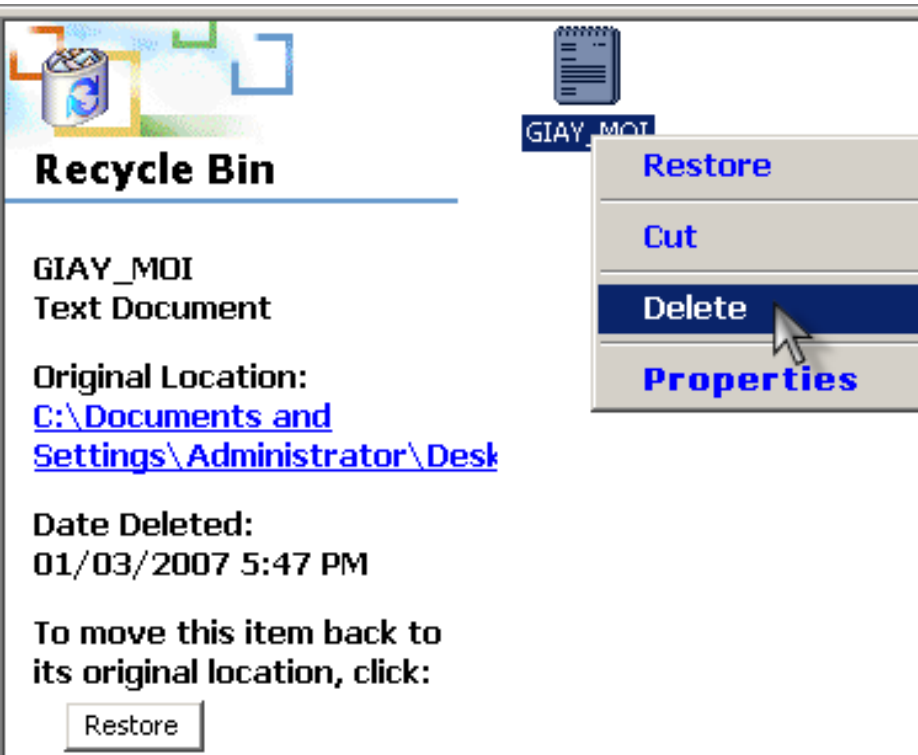
2/ Khôi phục các tập tin, thư mục bị xóa:

- Nhấn đúp vào biểu tượng Recycle Bin trên màn hình nền **Desktop**.
- Kích phải chuột trên đối tượng cần khôi phục.
- Chọn **Restore**.

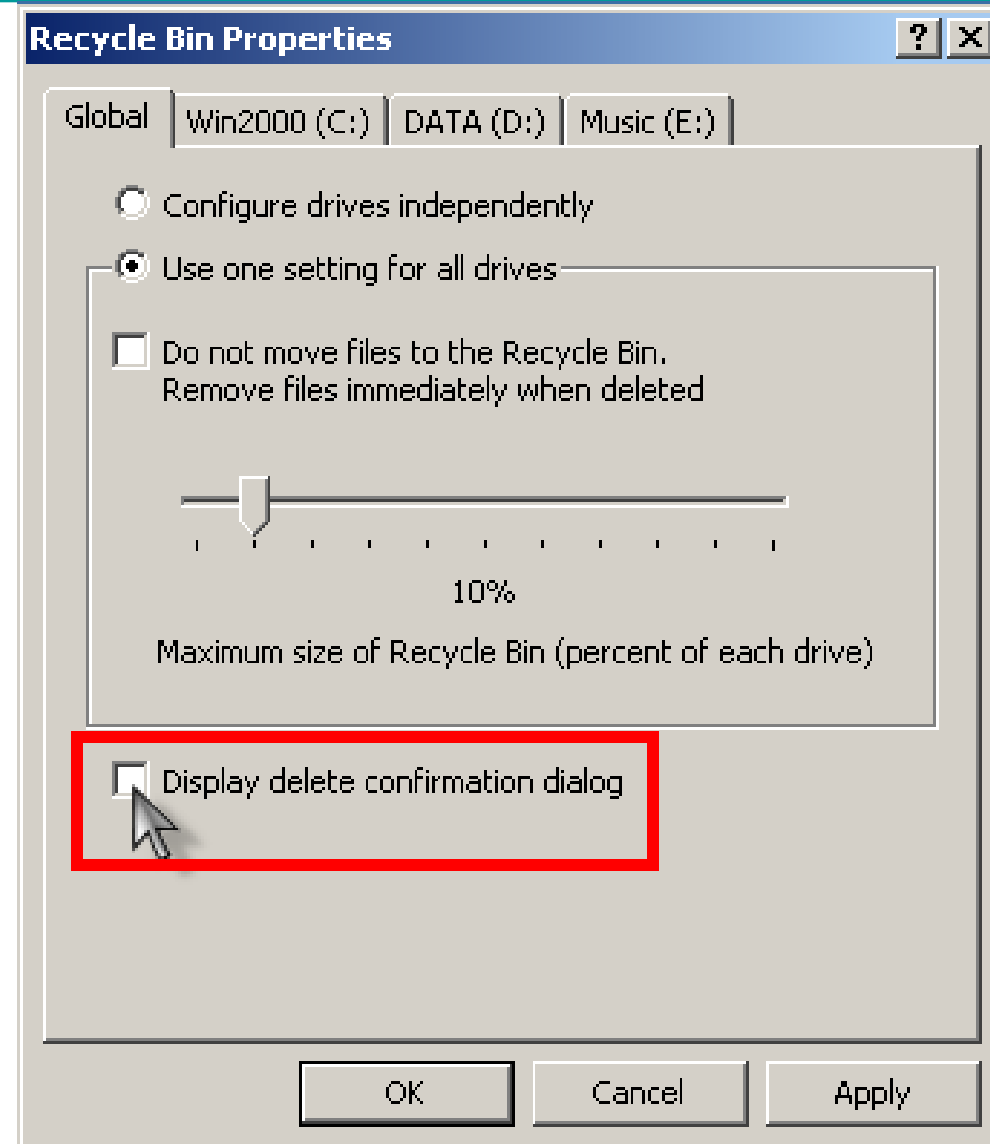


3/ ĐỒ rác

- Để xóa một tập tin, thư mục khỏi thùng rác. Kích phải chuột chọn Delete. Nhấn Yes.
- Để xóa tất cả trong thùng rác. Vào File, chọn Empty Recycle Bin.

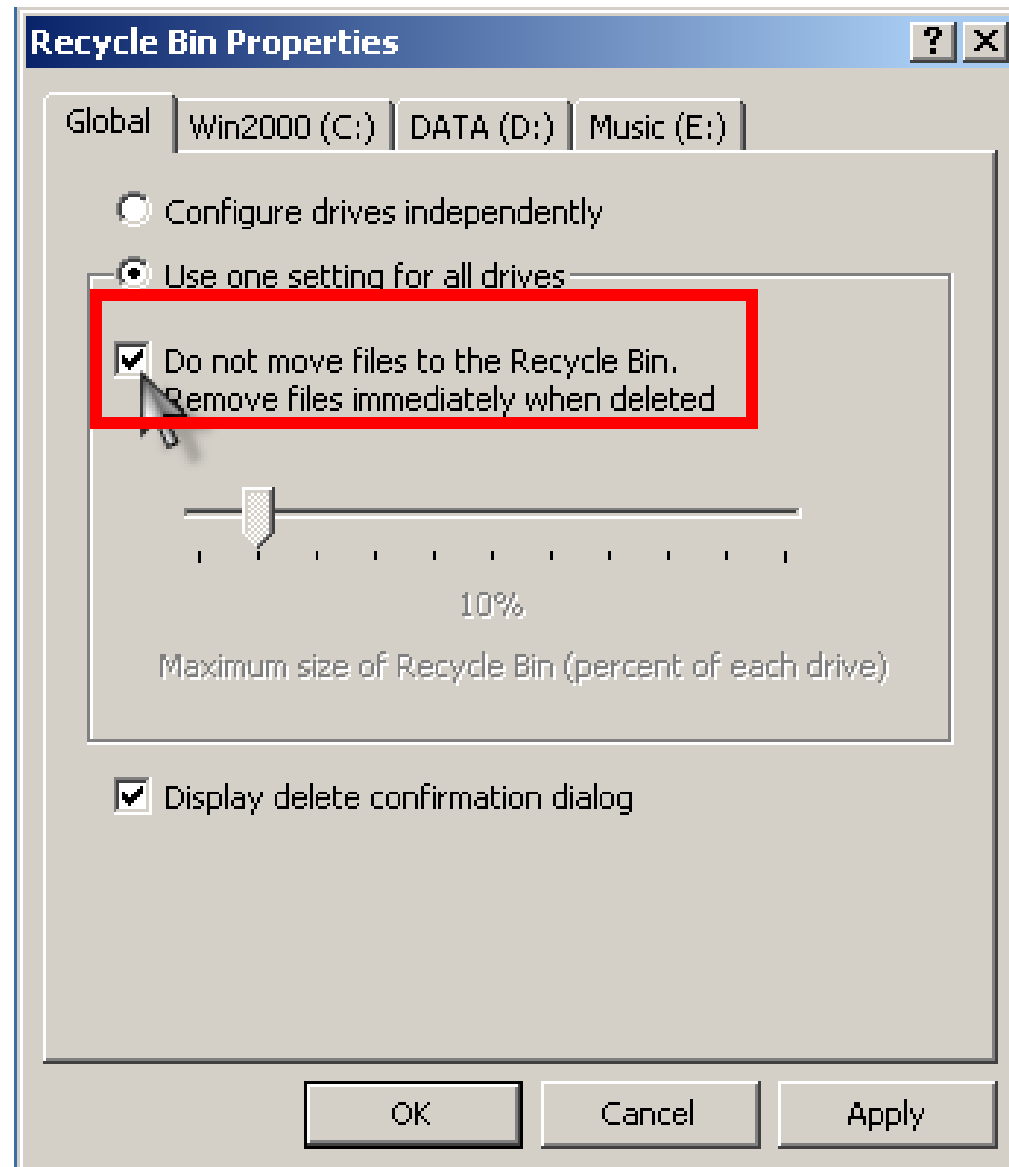


- Kích phải chuột trên **Recycle Bin**.
- Chọn **Properties**.
- Bỏ chọn mục: **Display delete confirmation dialog**.



5/ Không sử dụng thùng rác

- Kích phải chuột trên thùng rác.
- Chọn Properties.
- Đánh dấu mục: **Do not move files to the Recycle Bin. Remove files immediately when deleted.**



BÀI 6

CONTROL PANEL- BẢNG ĐIỀU KHIỂN MÁY TÍNH

I/ Giới thiệu

- ❑ **Control Panel** chứa các công cụ giúp người sử dụng thiết lập- cá nhân hóa PC theo ý mình.
- ❑ Mỗi thành phần gọi là **Applet**

II/ Khởi động Control Panel

- Start Menu – Settings – Control Panel (Hoặc phím Windows – S – C)
- Nhập **Control** vào cửa sổ Run (Windows + R). Nhấn Enter.
- My Computer. Nhấn đúp Control Panel.



1/ Xem thông tin hệ thống

➤ Nháy đúp vào biểu tượng System trong cửa sổ Control Panel (Hoặc có thể nháy phải chuột vào **My Computer**, chọn **Properties**)



System

- Chọn thẻ General:
- ✓ System: Hệ điều hành đang sử dụng
 - ✓ Computer: CPU loại gì, dung lượng RAM

2/ Thiết lập màn hình nền Desktop

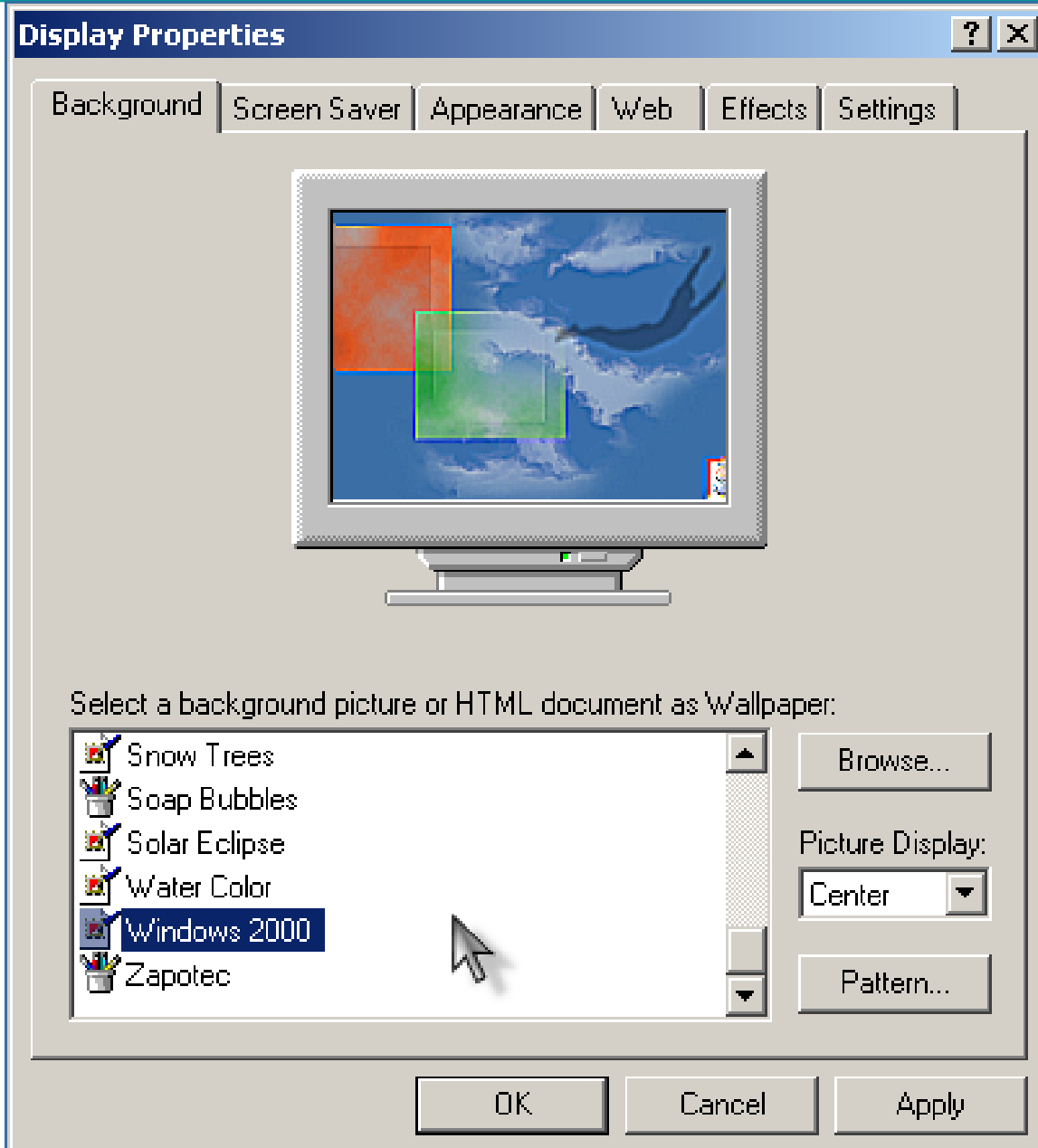
➤ Chạy Display



Display



➤ Có thể kích phải chuột trên Desktop, chọn Properties



3/ Thiết lập ngày/giờ hệ thống

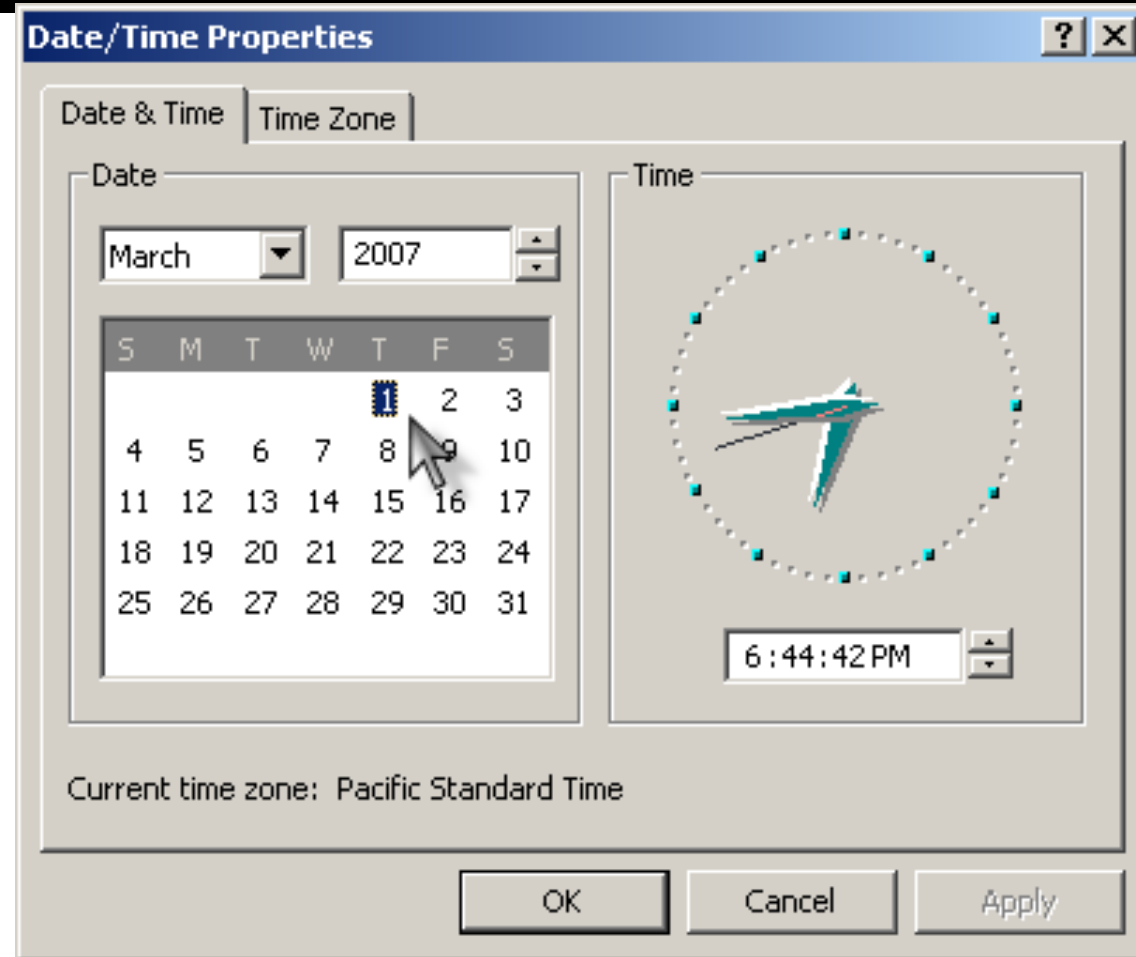
- Chạy **Date/Time** (Hoặc có thể nháy đúp chuột vào đồng hồ dưới khay hệ thống)



Date/Time



- Có thể thay đổi ngày, giờ, múi giờ phù hợp với hiện tại.



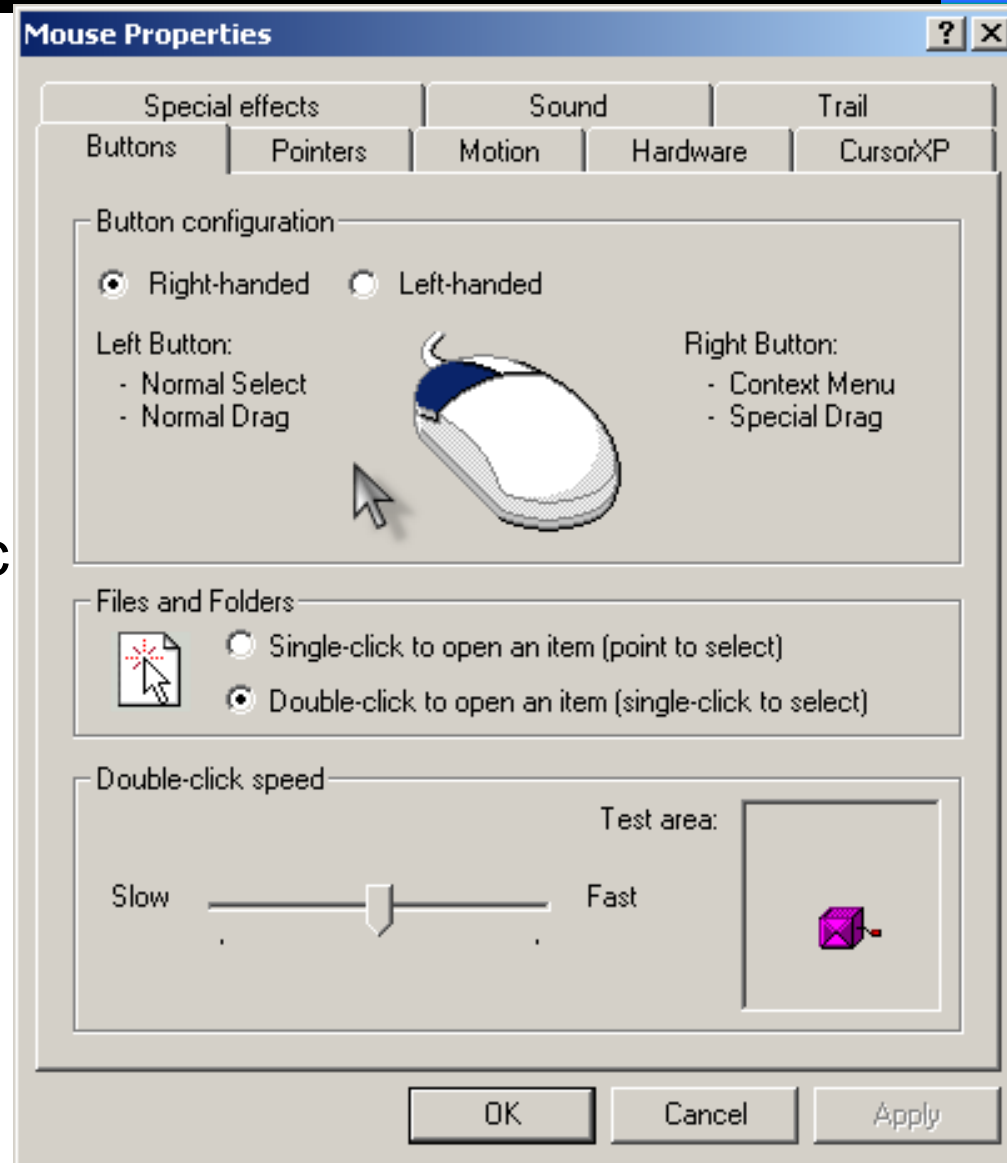
4/ Thiết lập chuột (Mouse)

☐ Chạy Mouse

- **Thẻ Buttons:** Thiết lập các nút nhấn
- **Thẻ Pointer:** Thiết lập các kiểu con trỏ chuột
- **Thẻ Motion:** Thiết lập tốc độ, hình dạng di chuyển của chuột.



Mouse



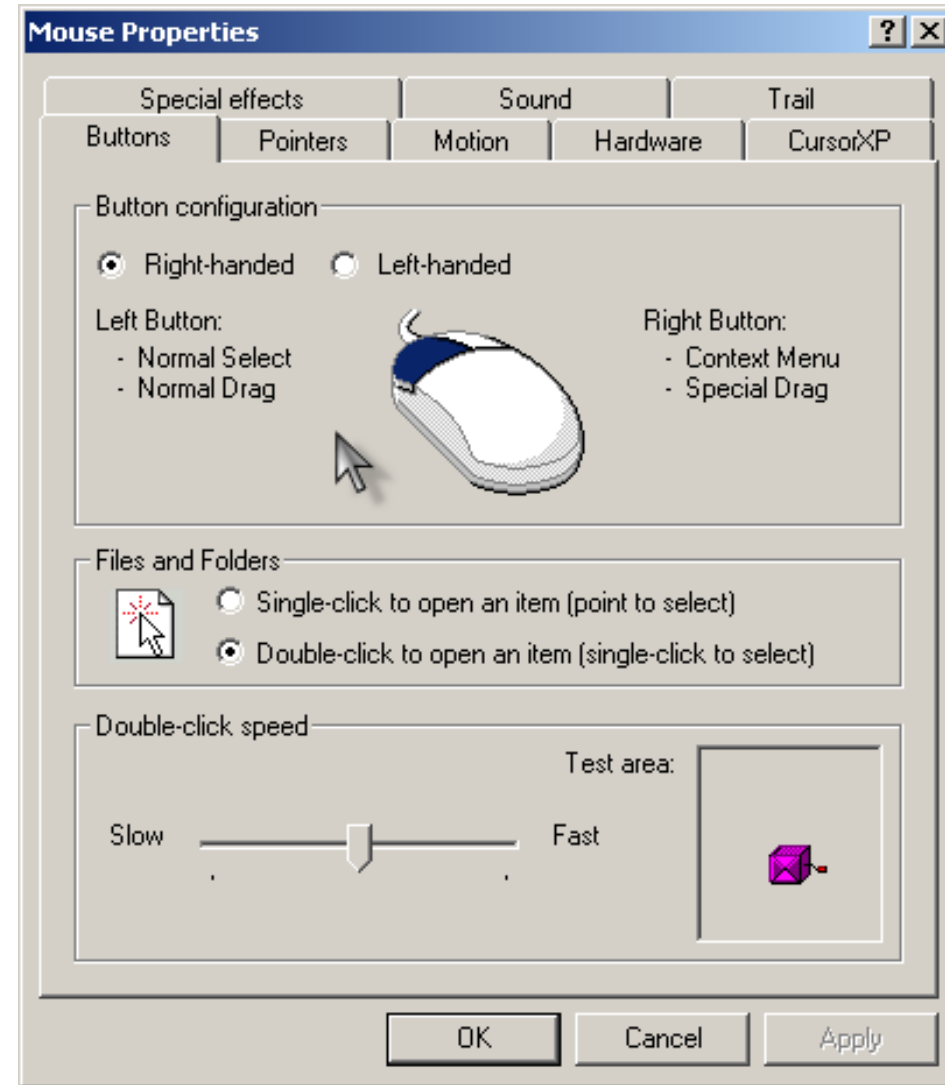
4.1/ Thiết lập nút nhấn

Button Configuration:

Thay đổi tay sử dụng chuột.

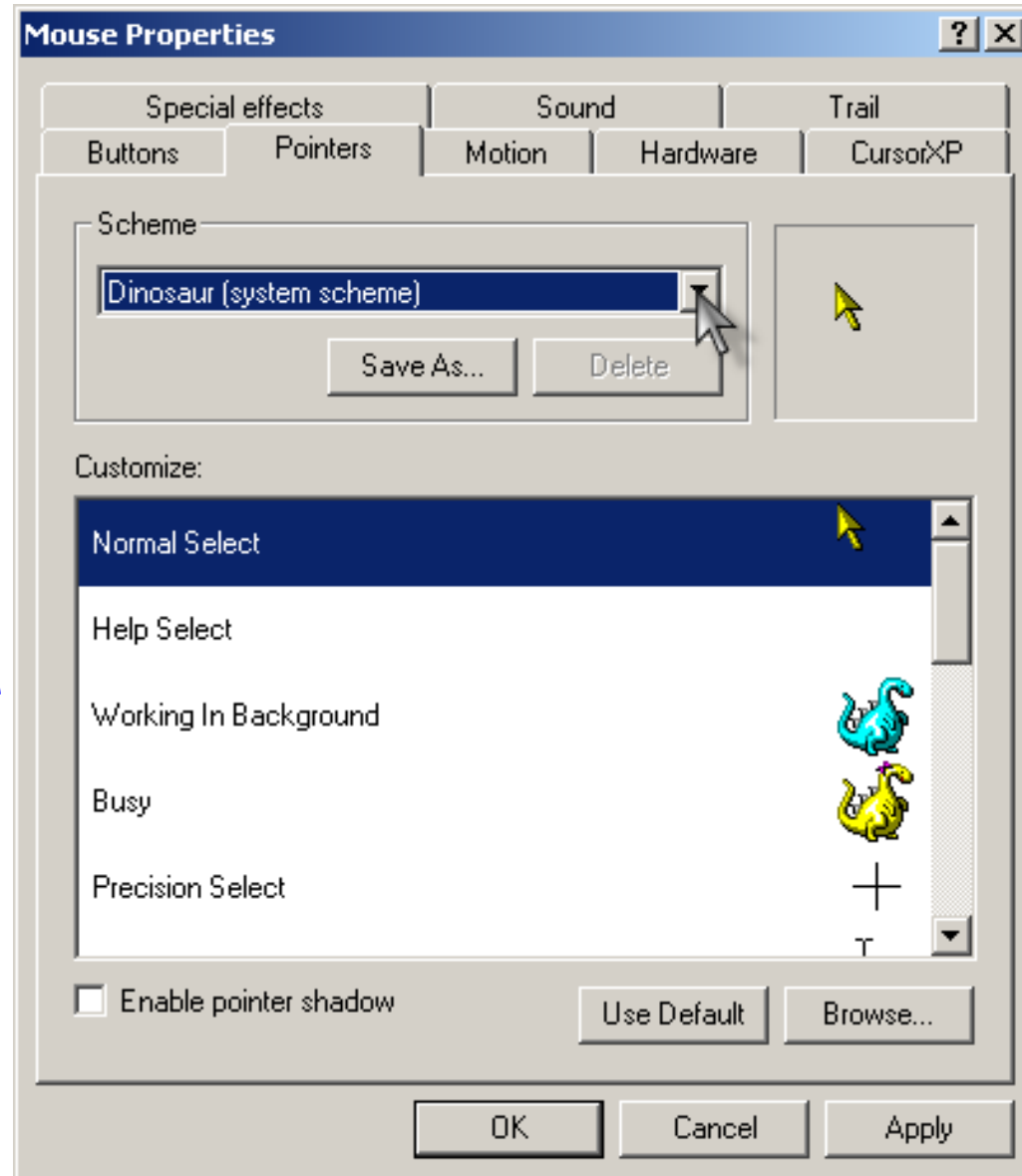
File or Folders: Thay đổi cách mở tệp tin, thư mục bằng chuột.

Double click Speed: Thay đổi tốc độ nháy đúp chuột.



4.2/ Thiết lập kiểu con trỏ

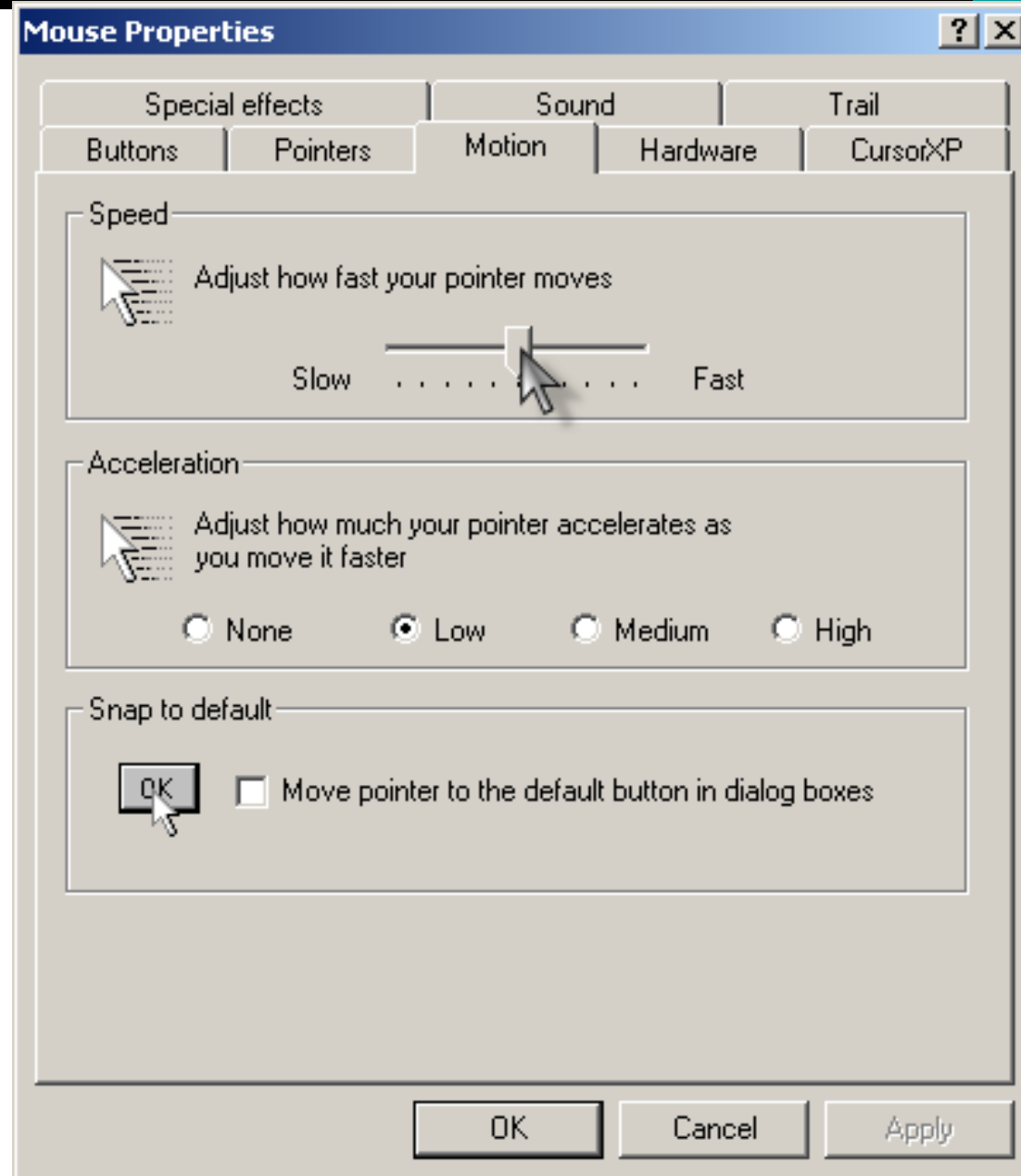
- ❑ Chọn một mẫu có sẵn trong mục **Scheme**.
- ❑ Kích chọn từng kiểu con trỏ. Nhấn nút **Browse** để tìm những hình dạng con trỏ khác.
- ❑ Dùng nút **Use Default** để trở về con trỏ mặc định.



4.3/ Thiết lập sự di chuyển

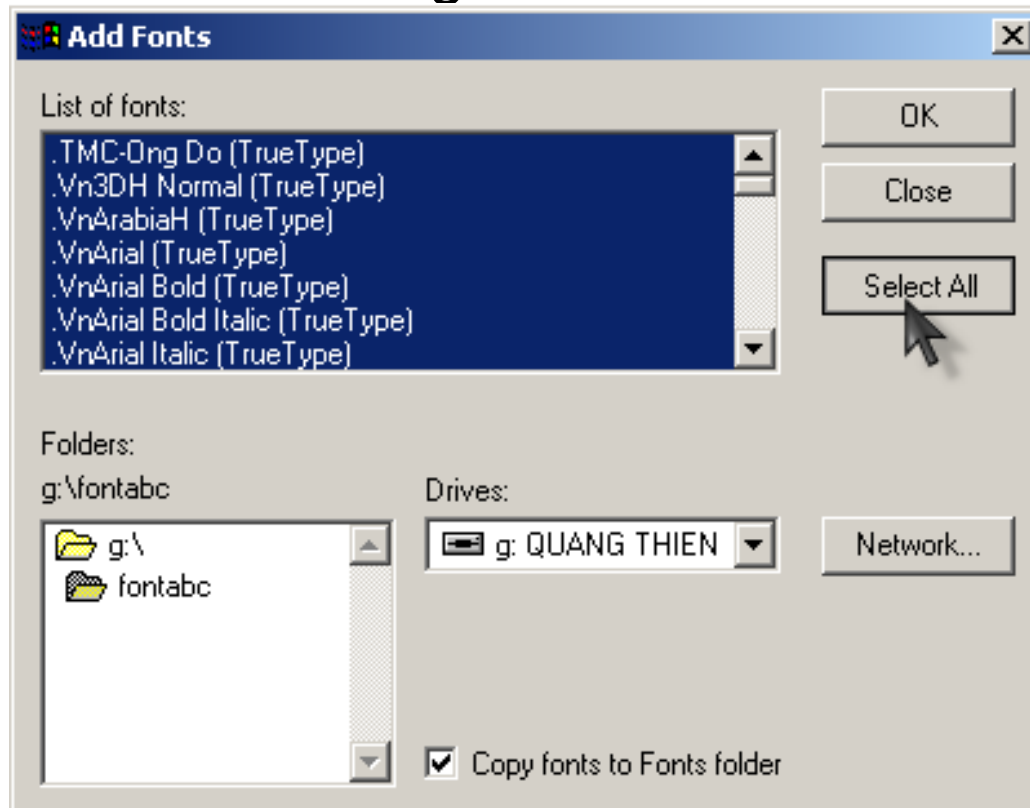
Nên dùng kiểu mặc định:

- **Speed:** Kéo về vị trí chính giữa.
- **Acceleration:** Chọn Low
- **Snap to Default:** Bỏ chọn



5/ Quản lý Font chữ

- ❑ Chạy Fonts
- ❑ Cửa sổ Font quản lý tất cả các font đang có trong máy.
- ❑ Cài thêm Font mới:
 - File-Install New Font.
 - Chọn ổ đĩa và thư mục có chứa font nguồn.
 - Chọn font cần cài trong danh sách.

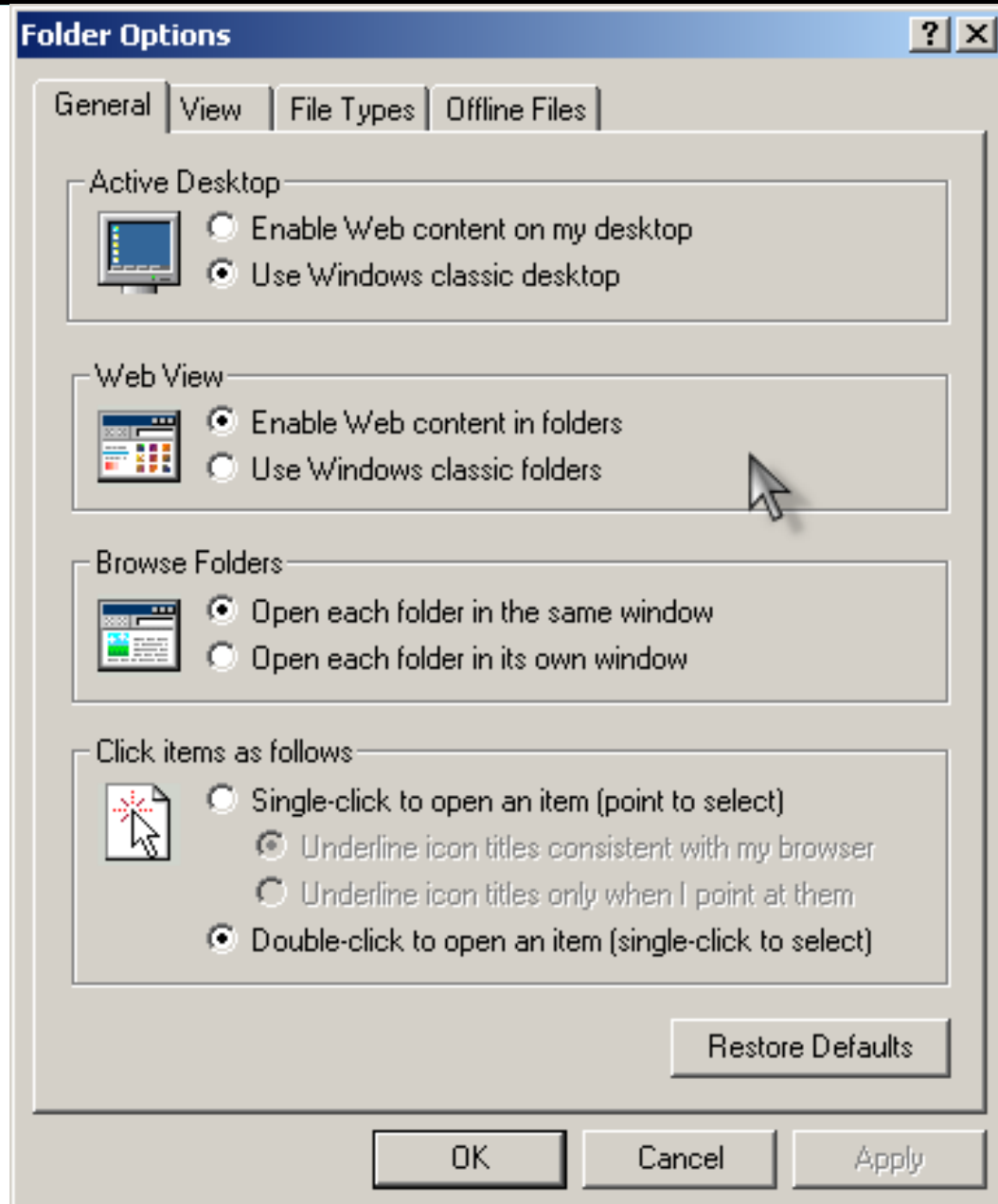


6/ Thiết lập cho tập tin & thư mục

- Chạy Folder Options.
- Hoặc menu Tools – Folders Options.



Folder Options

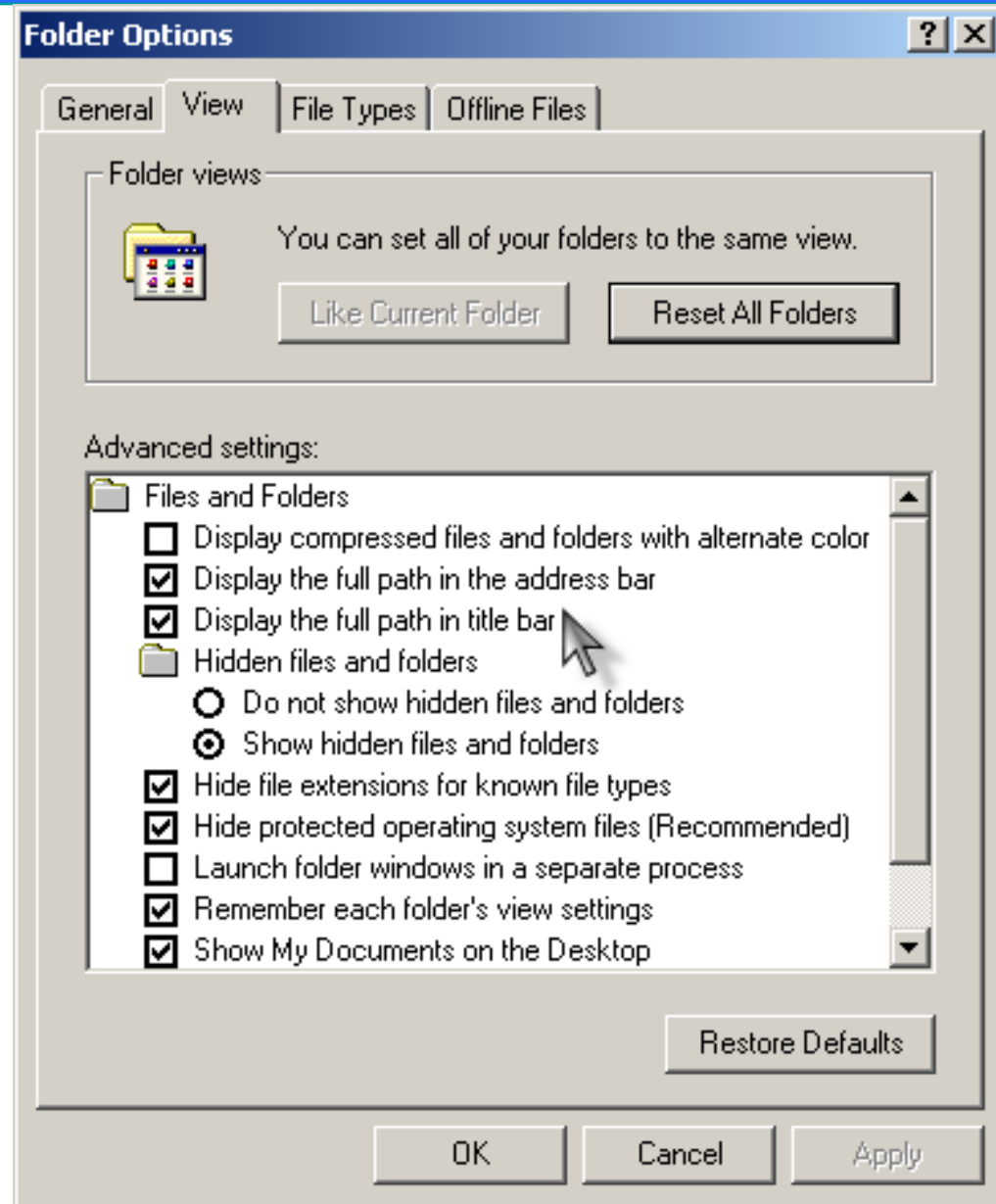


6.1/ Thiết lập chung (thẻ General)

- **Active Desktop**: Kiểu hiển thị của Desktop. Chọn **Use Windows classic Desktop**.
- **Web view**: Quy cách hiển thị cửa sổ thư mục. Chọn **Enable Web content in folders**.
- **Browse Folders**: Cách hiển thị cửa sổ khi truy cập thư mục. Chọn **Open each folder in the same window**.
- **Click items and follows**: Cách mở tập tin và thư mục. Chọn **Double - click**

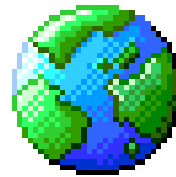
6.2/ Cách hiển thị tập tin & thư mục (thẻ View)

- **Display full path in the address bar:** hiển thị đầy đủ đường dẫn trên thanh địa chỉ.
- **Do not show hidden files and folders:** Không hiển thị các tập tin và thư mục bị ẩn.
- **Show all hidden files and folders:** Hiển thị tất cả các tập tin và thư mục bị ẩn.
- **Hide file extensions for known file types:** Ẩn phần mở rộng của các kiểu tập tin đã được nhận dạng.
- **Hide protected operating system files:** Ẩn những tập tin hệ thống của HĐH.



7/ Thiết lập theo vùng miền

- ❑ Chạy Regional Options.
- ❑ Bao gồm các thiết lập về ngôn ngữ sử dụng, hệ số, hệ tiền tệ, hệ ngày tháng, hệ thời gian.

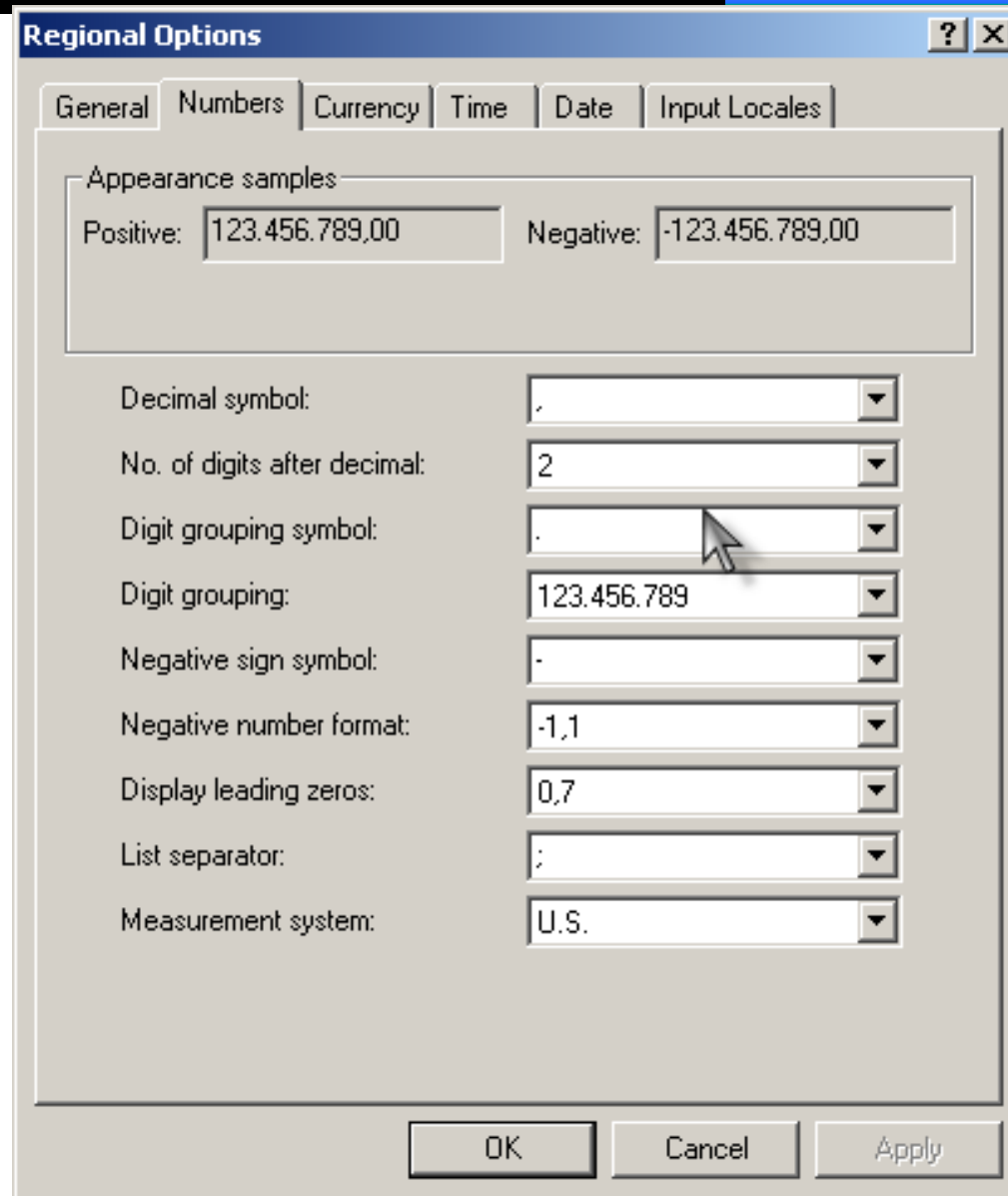


Regional
Options



7.1/ Thay đổi hệ số

- ❑ **Mục đích:** Thay đổi từ hệ số của **USA** thành hệ số của **VN**.
- ❑ **Thực hiện:**
 - Chọn thẻ **Numbers**.
 - **Decimal symbol** - Dấu phân cách phần nguyên và phần thập phân: Nhập dấu phẩy (,).
 - **Digit grouping symbol** - Dấu nhóm số: Nhập dấu chấm (.)



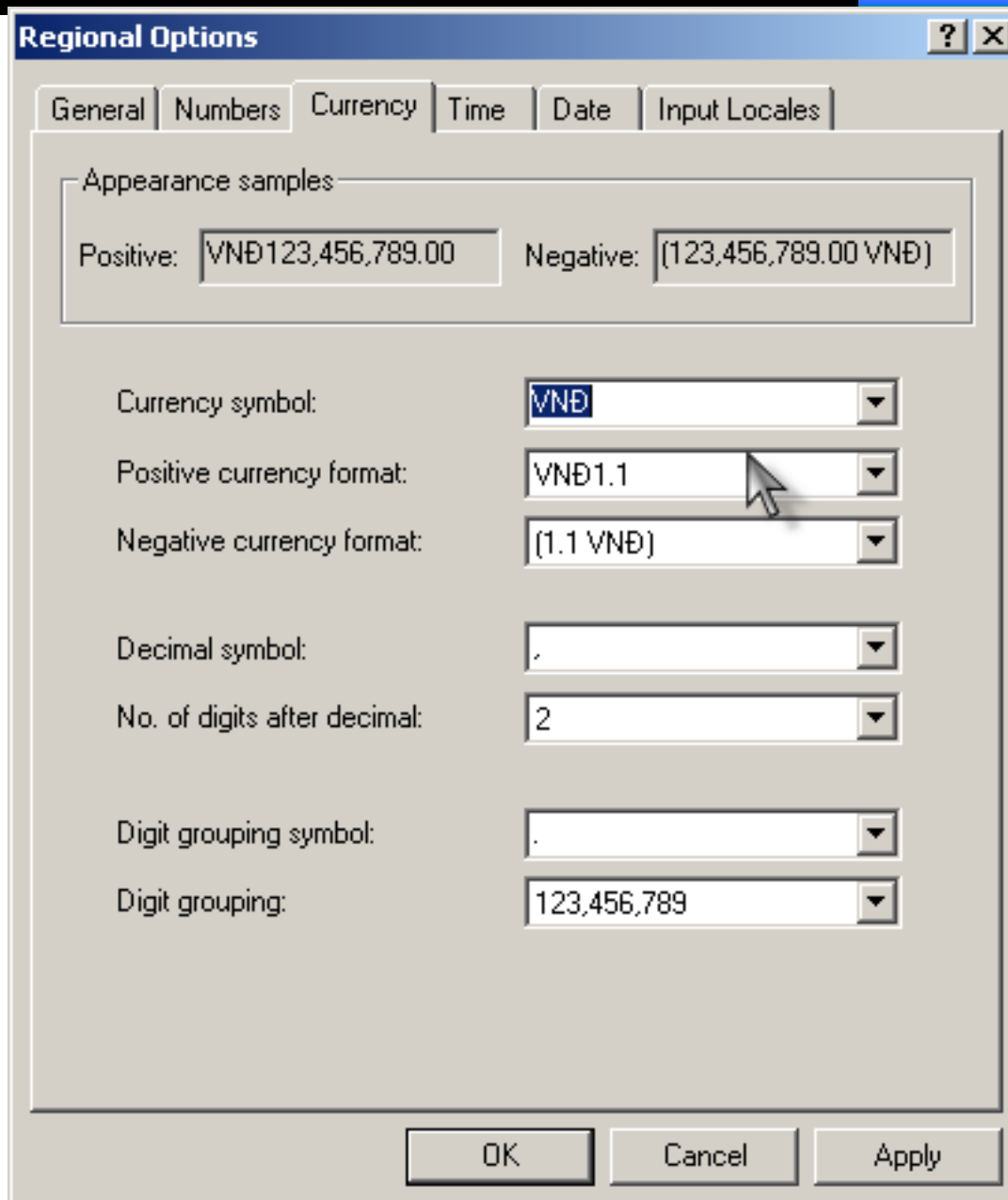
7.2/ Thay đổi hệ tiền tệ

Currency symbol:

Nhập ký tự đại diện tiền tệ. Ví dụ VNĐ.

Positive currency format:

Chọn kiểu hiển thị của ký tự tiền tệ trong một giá trị kiểu tiền tệ.



The screenshot shows the 'Regional Options' dialog box with the 'Currency' tab selected. The 'Appearance samples' section shows 'Positive: VNĐ123,456,789.00' and 'Negative: (123,456,789.00 VNĐ)'. The 'Currency symbol' dropdown is set to 'VNĐ'. The 'Positive currency format' dropdown is set to 'VNĐ1.1', and the 'Negative currency format' dropdown is set to '(1.1 VNĐ)'. The 'Decimal symbol' is set to '.', 'No. of digits after decimal' is set to '2', 'Digit grouping symbol' is set to '.', and 'Digit grouping' is set to '123,456,789'. The 'OK', 'Cancel', and 'Apply' buttons are visible at the bottom.

Regional Options

General | Numbers | Currency | Time | Date | Input Locales

Appearance samples

Positive: VNĐ123,456,789.00 Negative: (123,456,789.00 VNĐ)

Currency symbol: VNĐ

Positive currency format: VNĐ1.1

Negative currency format: (1.1 VNĐ)

Decimal symbol: .

No. of digits after decimal: 2

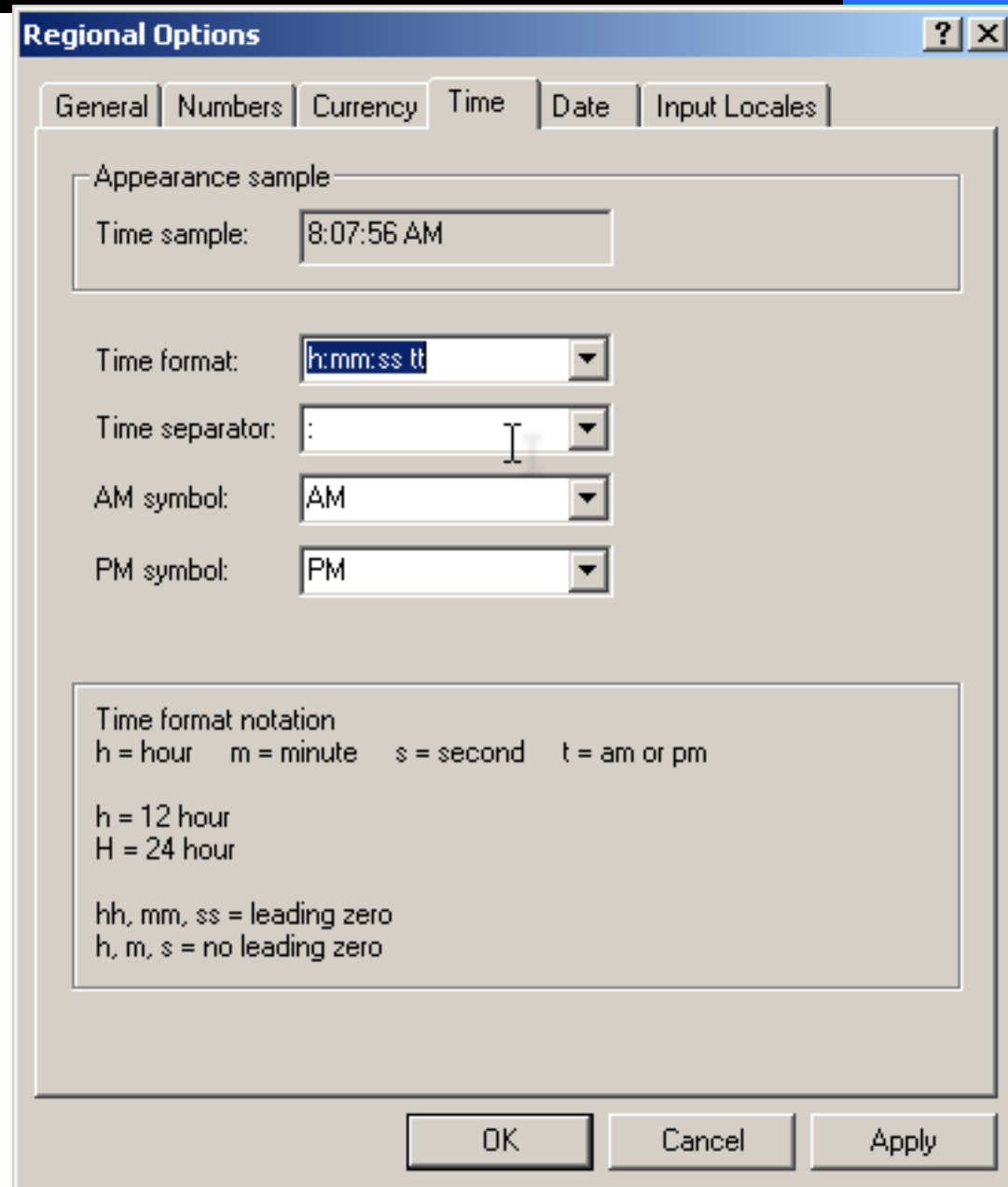
Digit grouping symbol: .

Digit grouping: 123,456,789

OK Cancel Apply

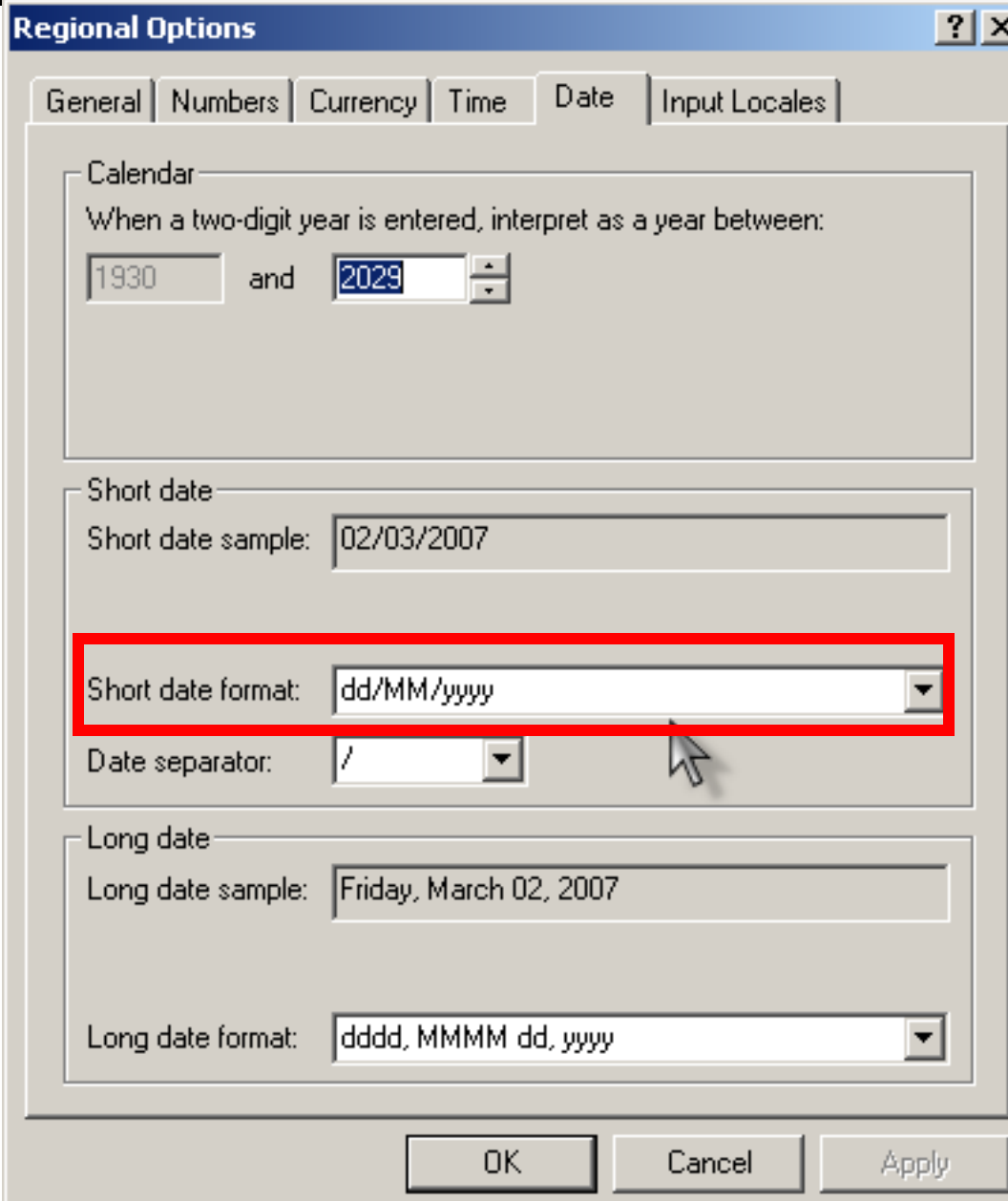
7.3/ Thiết lập thời gian

- Chọn một định dạng thời gian trong **Time Format**.
- Xem kết quả ở mục **Time Sample**.



7.4/ Thiết lập kiểu ngày tháng

- Nhập **dd/mm/yyyy** biểu thị cho kiểu ngày VN vào ô **Short Date Format**.



The image shows a screenshot of the Windows Regional Options dialog box, specifically the Date tab. The dialog box has a title bar with a question mark and a close button. Below the title bar are several tabs: General, Numbers, Currency, Time, Date, and Input Locales. The Date tab is selected. The dialog is divided into several sections:

- Calendar:** A section with the text "When a two-digit year is entered, interpret as a year between:" followed by two input boxes. The first box contains "1930" and the second box contains "2029".
- Short date:** A section with a label "Short date sample:" and a text box containing "02/03/2007".
- Short date format:** A section with a label "Short date format:" and a dropdown menu. The dropdown menu is highlighted with a red rectangular box and shows the selected format "dd/MM/yyyy".
- Date separator:** A section with a label "Date separator:" and a dropdown menu showing the separator character "/".
- Long date:** A section with a label "Long date sample:" and a text box containing "Friday, March 02, 2007".
- Long date format:** A section with a label "Long date format:" and a dropdown menu showing the selected format "dddd, MMMM dd, yyyy".

At the bottom of the dialog box are three buttons: "OK", "Cancel", and "Apply".

BÀI 7

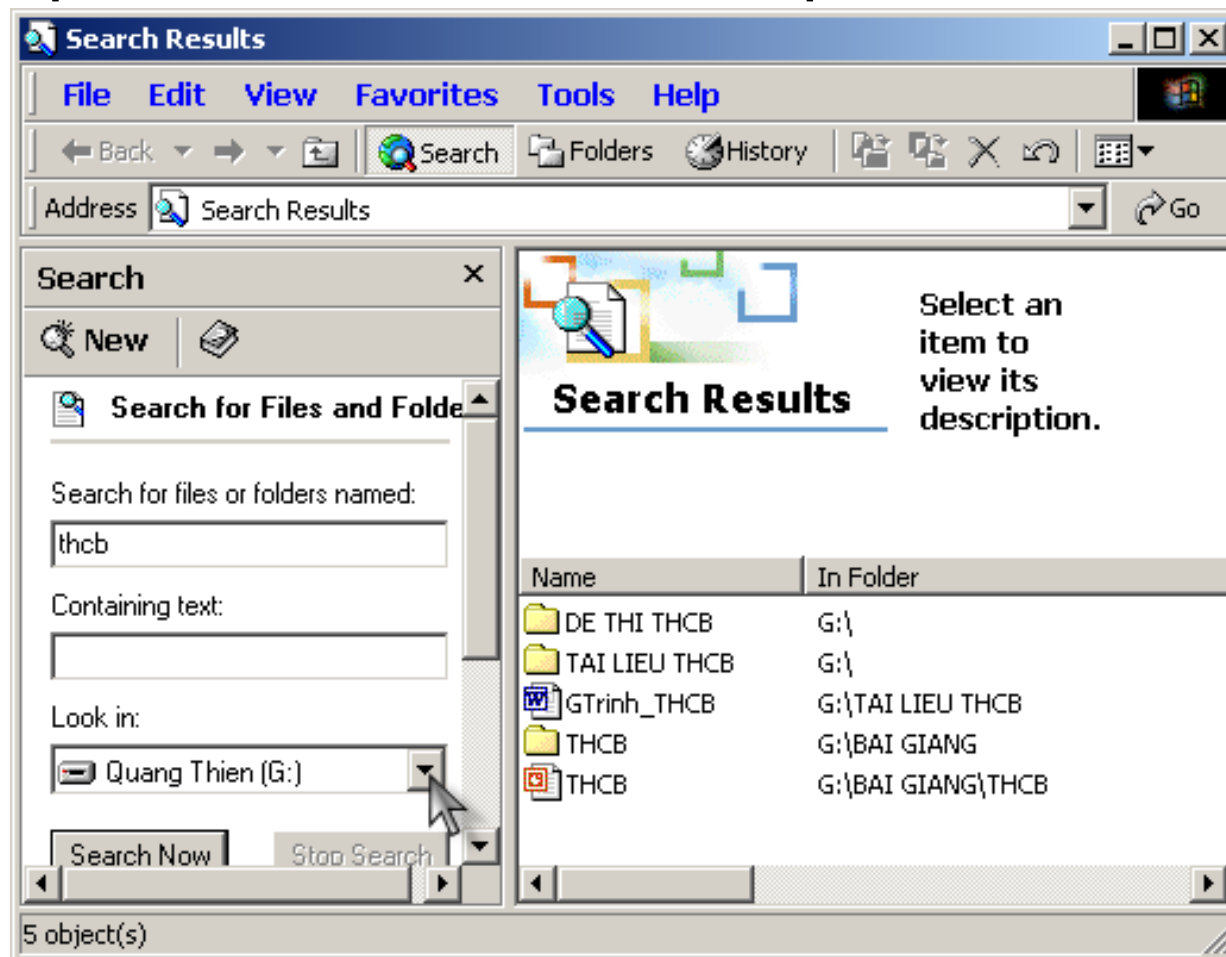
SỬ DỤNG CÁC TIỆN ÍCH CỦA WINDOWS

1/ Tìm kiếm

- **Mục đích:** Để tìm một tập tin, thư mục có tồn tại hay không trong máy.
- **Thực hiện:**
 - ✓ Start – Search – For Files or Folders. (hoặc phím Windows + F).
 - ✓ Nhấn F3 để tìm trong thư mục hiện tại.

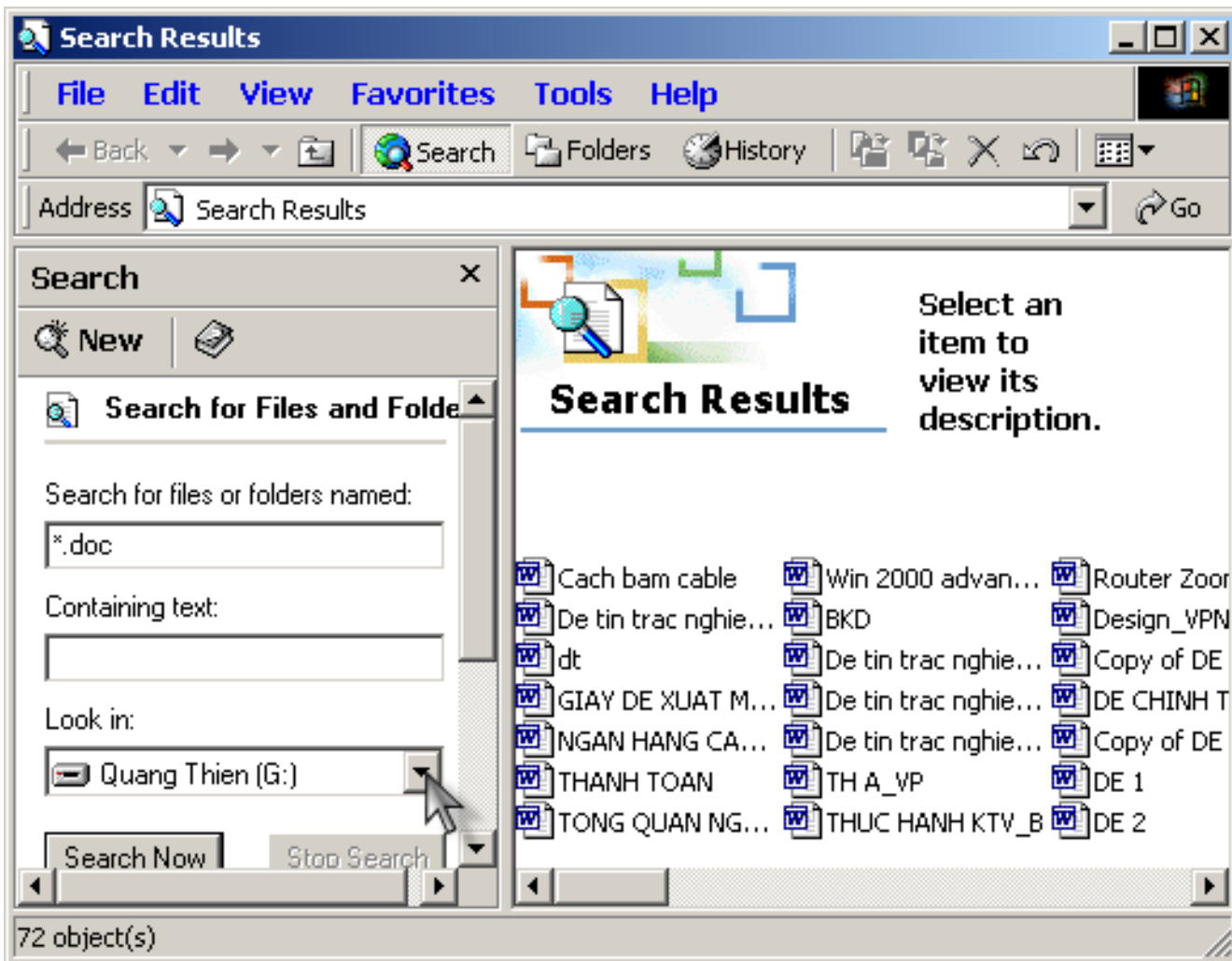
1.1/ Tìm kiếm đơn giản

- Nhập tên tập tin, thư mục cần tìm vào mục **Search for file or folders named**.
- Chọn nơi tìm trong mục **Look in** (nên chọn **My Computer**)
- Kích chuột vào nút **Search Now** hoặc nhấn **Enter**.



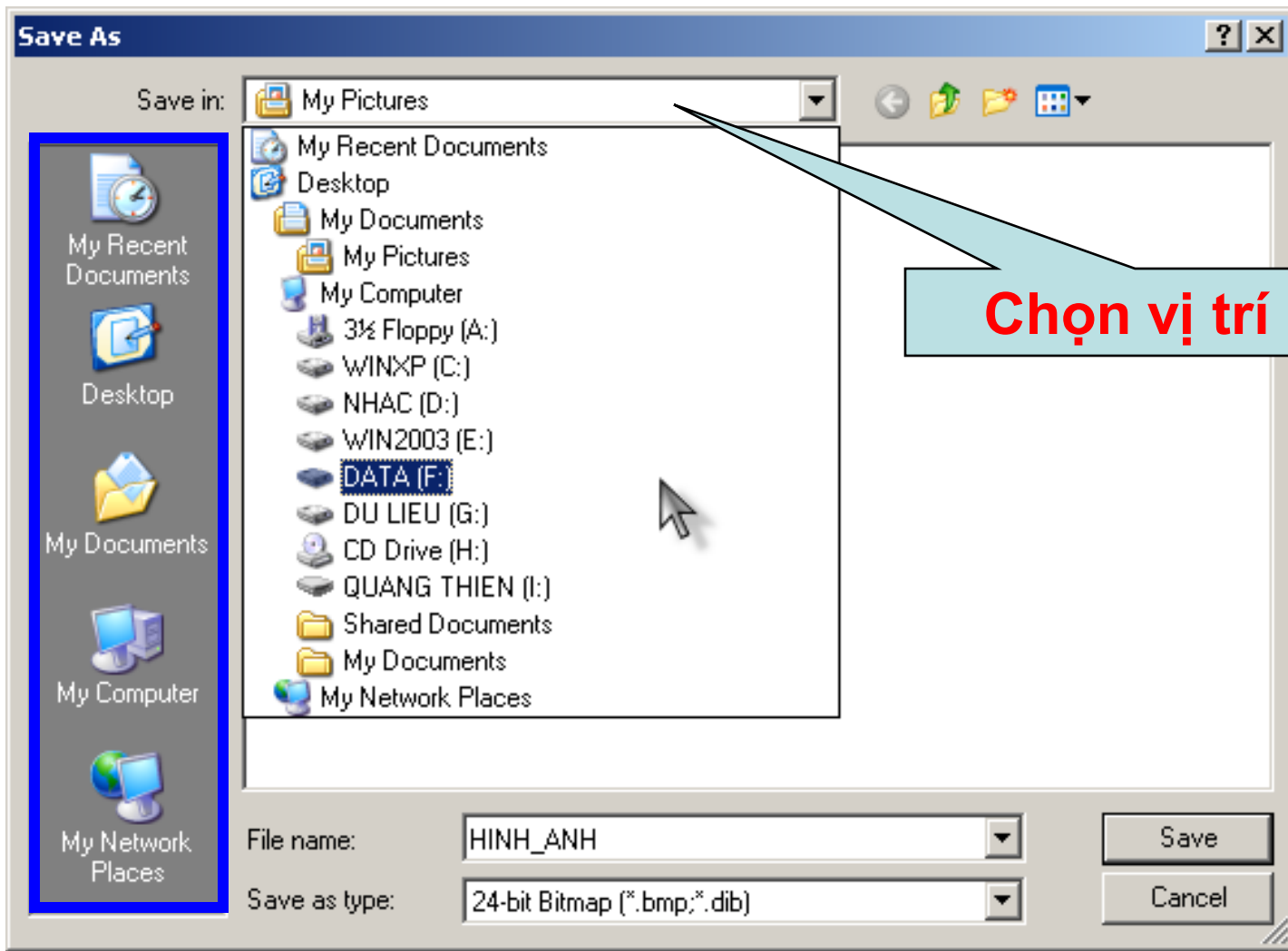
1.2/ Sử dụng ký tự thay thế

- Dùng dấu ? để đại diện cho 1 ký tự không xác định được.
- Dùng dấu * để đại diện cho 1 hoặc nhiều ký tự.
- Ví dụ: Nhập *.doc để tìm tất cả các tập tin văn bản word.



2/ Sử dụng MS Paint

- Gõ mspaint vào cửa sổ RUN.
- Vào **File – Save**. Chọn nơi lưu hình và đặt tên cho hình. Nhấn Save.



Module 3

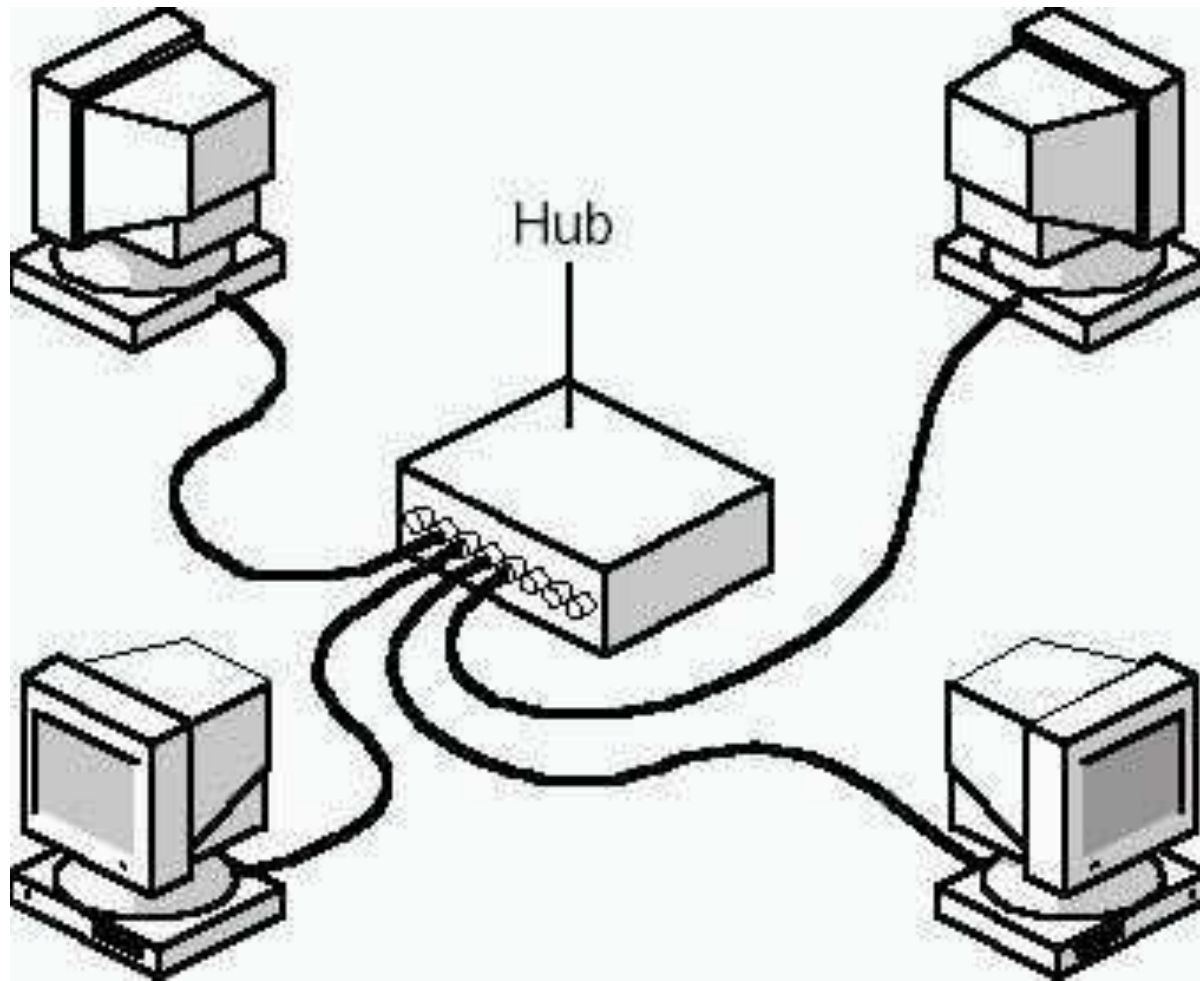
CÔNG NGHỆ INTERNET

I/ Khái niệm mạng máy tính

- Mạng máy tính là một tập hợp các máy tính được nối với nhau bởi môi trường truyền (đường truyền) theo một cấu trúc nào đó và thông qua đó các máy tính trao đổi thông tin qua lại cho nhau.
- Môi trường truyền là hệ thống các thiết bị truyền dẫn có dây hay không dây dùng để chuyển các tín hiệu điện tử từ máy tính này đến máy tính khác.

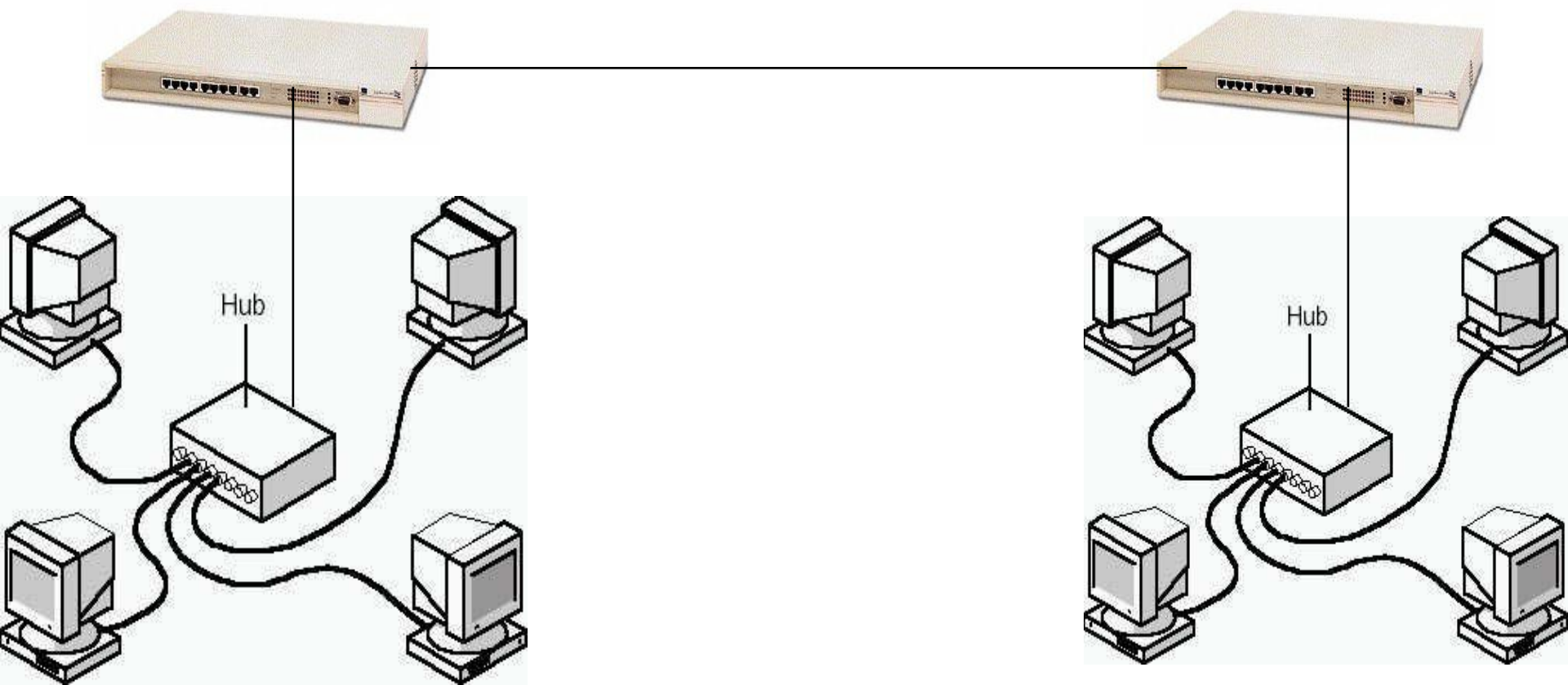
II/ Định nghĩa mạng LAN, WAN

- ❑ Mạng **LAN** (Local Area Network) là mạng nội bộ, kết nối các máy tính của một cơ quan, tổ chức trong phạm vi nhỏ, bán kính vài trăm mét.



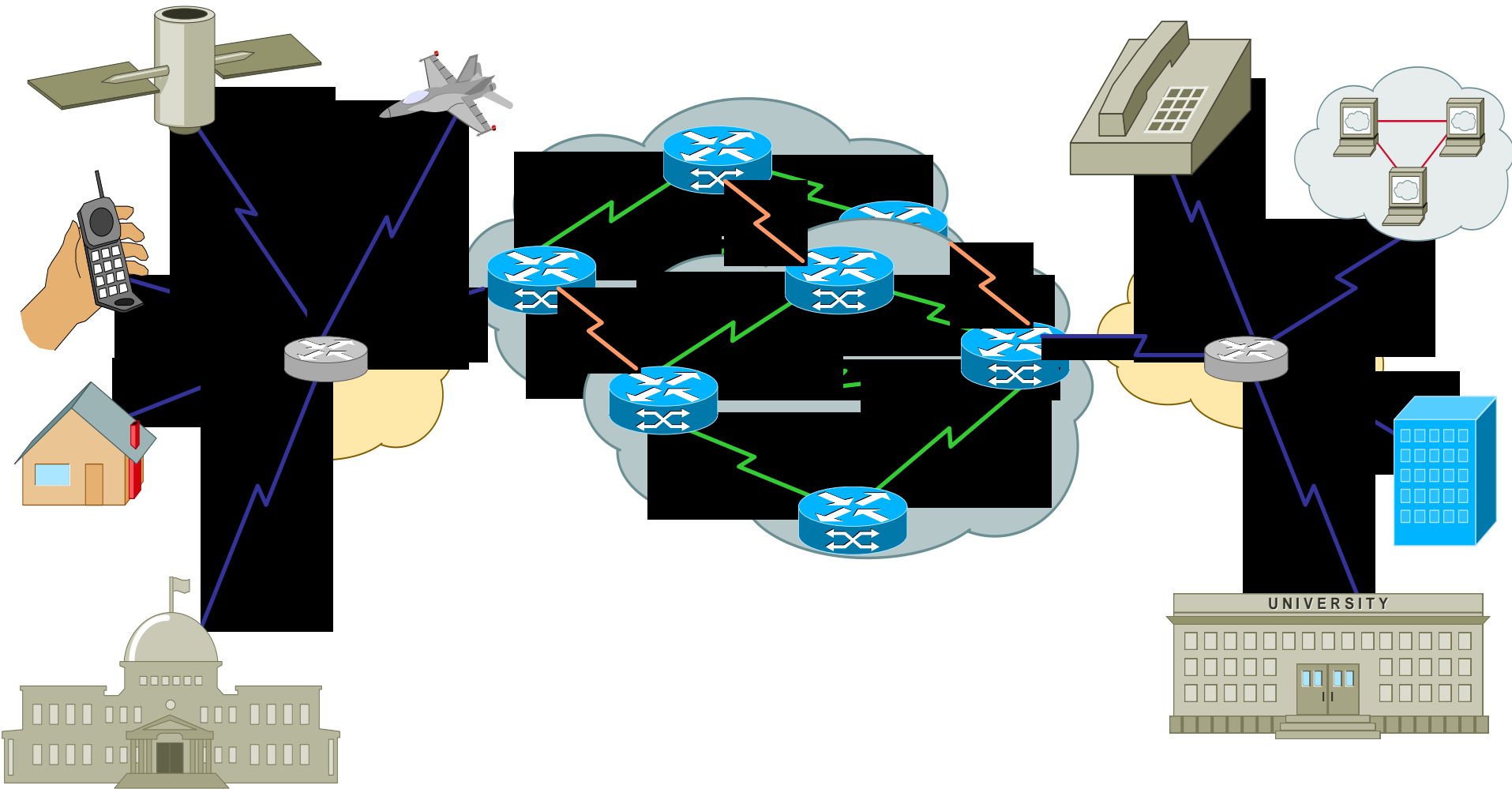
II/ Định nghĩa mạng LAN, WAN (tt)

- ❑ Mạng **WAN** (**W**ide **A**rea **N**etwork) là mạng diện rộng, kết nối các mạng LAN ở xa nhau để có một mạng duy nhất. Thông thường kết nối này được thực hiện thông qua mạng viễn thông hay vệ tinh.



III/ Internet là gì?

- ❑ Internet là liên mạng máy tính toàn cầu. Là sự kết hợp các mạng LAN, WAN trên toàn thế giới.
- ❑ Tháng 12/1997, Việt Nam kết nối Internet.



IV/ Một số địa chỉ Web

1/ Tin tức:

- ◆ Cổng Internet VN: home.vnn.vn
- ◆ Báo tuổi trẻ: www.tuoiitre.com.vn
- ◆ Báo TN: www.thanhvien.com.vn
- ◆ Báo CAND: www.cand.com.vn
- ◆ Báo CA TP HCM:
www.baocongantphcm.com.vn
- ◆ Báo Lao động: www.laodong.com.vn
- ◆ Báo Nhân dân: www.nhandan.com.vn
- ◆ Tin nhanh: www.vnexpress.net

IV/ Một số địa chỉ Web (tt)

2/ Giải trí:

- ◆ Nhạc việt: nhacviet.vietnamnet.vn
- ◆ VN Thư quán: vnthuquan.net
- ◆ Nhạc số: www.nhacso.net
- ◆ Nhạc QN: www.thienan.org
- ◆ Người mẫu: www.mm52.com,
www.24h.com.vn, www.ngoisao.net

IV/ Một số địa chỉ Web (tt)

3/ Tìm kiếm:

- ◆ Google: www.google.com.vn
- ◆ Vinaseek: www.vinaseek.com
- ◆ Panvietnam: www.panvn.com

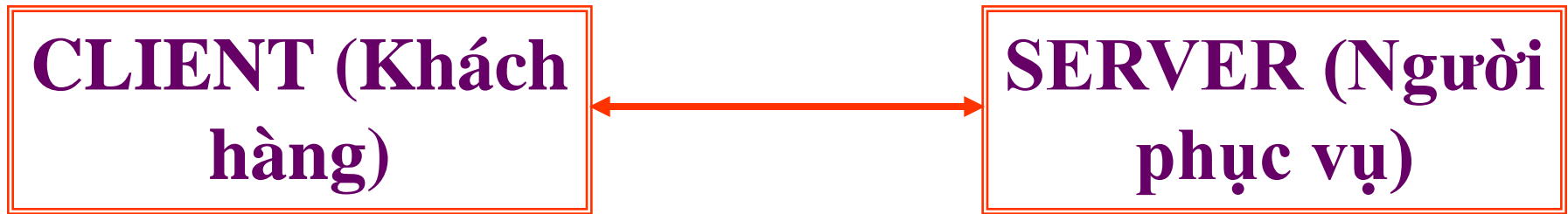
IV/ Một số địa chỉ Web (tt)

4/ Tin học:

- ◆ Tuần báo Echip: www.echip.com.vn
- ◆ Báo PC World: www.pcworld.com.vn
- ◆ Diễn đàn Mã nguồn: www.manguon.com
- ◆ Báo Tin học & Nhà trường: www.thnt.com.vn
- ◆ Quản trị mạng: www.quantrimang.com

V/ Mô hình khách hàng/người phục vụ

- ❑ **Mô hình khách hàng/người phục vụ:** Hệ thống Web được xây dựng theo mô hình khách hàng/người phục vụ (tiếng Anh gọi là **Client/Server**)



- ❑ Phải có tồn tại **các điểm phục vụ Web**, gọi là Web Server. Thì chúng ta mới có được những trang Web.

VI/ Qui trình kết nối và các dịch vụ trên Internet

❑ Qui trình kết nối Internet:

PC – MODEM – PHONE LINE - ISP - INTERNET

❑ Internet hoạt động theo mô hình **Client/Server** (khách/phục vụ) và cung cấp rất nhiều dịch vụ hữu ích.

- **www** (World Wide Web): dịch vụ trang tin toàn cầu
- **E-mail** (Electronic Mail): dịch vụ thư điện tử
- **FTP** (File Transfer Protocol): dịch vụ truyền dữ liệu trên Internet.

VII/ Phân biệt Internet, Intranet, Extranet

- ❑ Mạng LAN có triển khai các dịch vụ giống Internet như tư điện tử, trang tin nội bộ giống các dịch vụ của Internet mà không có kết nối với bên ngoài gọi là **Intranet**.
- ❑ Mạng LAN có triển khai các dịch vụ giống Internet và kết nối với Internet gọi là **Extranet**.





VIII/ Sử dụng trình duyệt Web Internet Explorer (IE)

- ❑ **Thanh công cụ (Standard Buttons):** Cho phép thực hiện nhanh các lệnh thay vì phải thực hiện từ Menu.
- ❑ **Thanh địa chỉ (Address Bar):** Muốn truy cập trang Web nào thì gõ địa chỉ của trang đó vào thanh địa chỉ và nhấn Enter.



VIII/ Sử dụng trình duyệt Web Internet Explorer (IE)

☐ Một số chức năng trên thanh công cụ:

- **Nút Back**  : Trở về trang trước trang hiện tại.
- **Nút Forward**  : Tiến tới trang trước khi quay về.
- **Nút Stop**  : Dừng tải nội dung trang hiện tại.
- **Nút Refresh**  : Nút làm tươi nội dung, tải lại nội dung trang web đang xem.



IX/ Những khái niệm cơ bản trên Internet

□ Trình duyệt web (Browser):

- Để sử dụng dịch vụ www (trang tin toàn cầu) máy bạn cần có ít nhất một trình duyệt web như Internet Explorer (IE), Mozilla Firefox, Netscape Navigator, Opera,...
- Trình duyệt web là một chương trình tải các trang tin (web) về máy của người sử dụng từ máy chủ phục vụ web.




IX/ Những khái niệm cơ bản trên Internet (tt)

□ Domain (Tên miền):

- Tên miền là tên của một máy chủ phục vụ web trên mạng.
- Tên miền do tổ chức Internet quốc tế quản lý.
- Tên miền là tên duy nhất trên thế giới.
- Cấu trúc tên miền gồm **WWW.<TÊN>.<MỞ RỘNG>**
- Ví dụ www.tuoitre.com.vn
- Một tên miền có thể có một hoặc nhiều phần mở rộng. Tên phần mở rộng cho biết lĩnh vực hoạt động hoặc quốc gia nơi nó đăng ký.
 - Quốc gia: .vn của VN, .uk của Anh, .us của Mỹ, .ca của Canada.
 - Lĩnh vực như: .net: liên mạng, .com là thương mại, .edu là giáo dục, .gov là chính phủ, .org là tổ chức.
- Tên miền là một hình thức thể hiện ngắn gọn của địa chỉ IP giúp người sử dụng truy cập một máy chủ phục vụ web không cần nhớ địa chỉ IP của máy đó.

IX/ Những khái niệm cơ bản trên Internet (tt)

- ❑ **HTTP**: Hyper Text Transfer Protocol – Phương thức truyền siêu văn bản trên Internet cho phép truyền nội dung những trang tin toàn cầu (WWW) từ máy phục vụ web về máy người sử dụng.
- ❑ **Hyperlink**-Siêu liên kết, là một tính chất quan trọng của trang Web. Tại vị trí nào đó của trang Web mà con trỏ chuột chuyển sang dạng **hình bàn tay trỏ**  thì ở vị đó có một siêu liên kết (Khi kích chuột sẽ liên kết (mở) đến một trang web mới).

IX/ Những khái niệm cơ bản trên Internet (tt)

- **ISP:** Nhà cung cấp dịch vụ Internet (Internet Service Providers): Nơi bạn đăng ký để gia nhập Internet và sử dụng các dịch vụ mà ISP cung cấp như Web, mail.
 - Một số ISP ở Việt Nam: Viettel, FPT, Saigonnet, VDC,...

X/ Tìm kiếm thông tin trên Internet

- Tìm kiếm qua các trang Web động gọi là máy tìm kiếm (**Search Engine**). Là một hệ thống cho phép tìm kiếm các thông tin trên Internet theo yêu cầu của người sử dụng.
- Ví dụ như trang Web: www.google.com là một trang Web hỗ trợ tìm kiếm cực mạnh, và được sử dụng phổ biến trên Internet hiện nay.



XI/ Phân biệt web tĩnh, web động

- ❑ Web tĩnh có nội dung được chuẩn bị sẵn, không thay đổi theo nhu cầu của người sử dụng. Ví dụ: những trang giới thiệu, CSDL luật,...
- ❑ Web động có nội dung thay đổi theo nhu cầu người sử dụng. Ví dụ các trang tin tức, tìm kiếm, bán hàng trên mạng,...

Module 4

**TRÌNH SOẠN THẢO VĂN BẢN
MICROSOFT WORD**

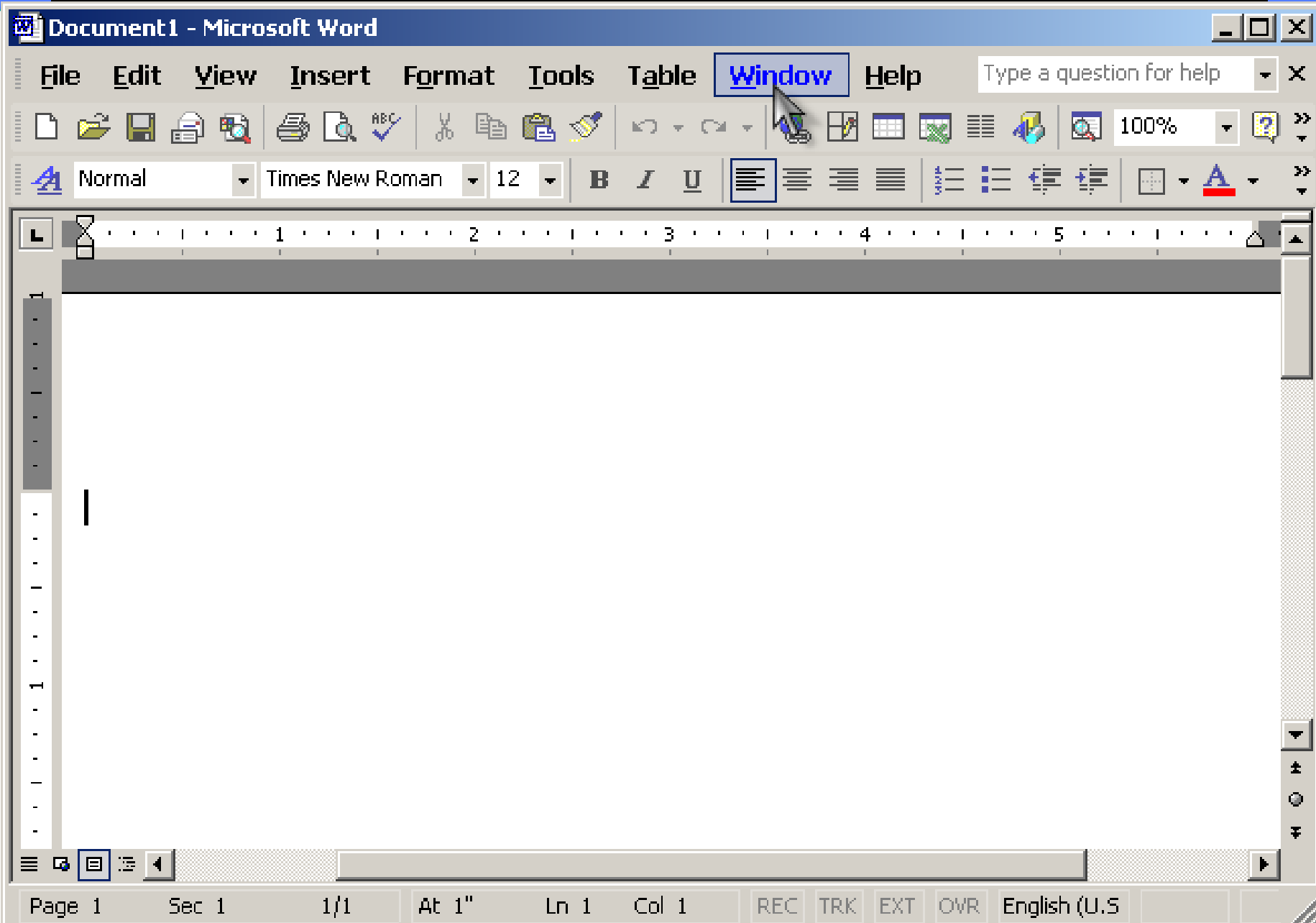
BÀI 1

TẠO VÀ LƯU TRỮ MỘT TÀI LIỆU MỚI

I/ Giới thiệu về Word

- ❑ Là 1 trong 6 phần mềm của bộ **Microsoft Office** của hãng **Microsoft** (Mỹ) sáng lập.
- ❑ Là phần mềm chuyên dùng xử lý văn bản trong công tác văn phòng.
- ❑ Tập tin thi hành của Word có tên: winword.exe
(Nằm trong đường dẫn: **C:\Program Files\Microsoft Office\Office\Winword.exe**)
- ❑ Tập tin mà Word sinh ra có phần mở rộng là: *.**doc**

II/ Màn hình làm việc của Word



III/ Tạo tệp tin văn bản mới

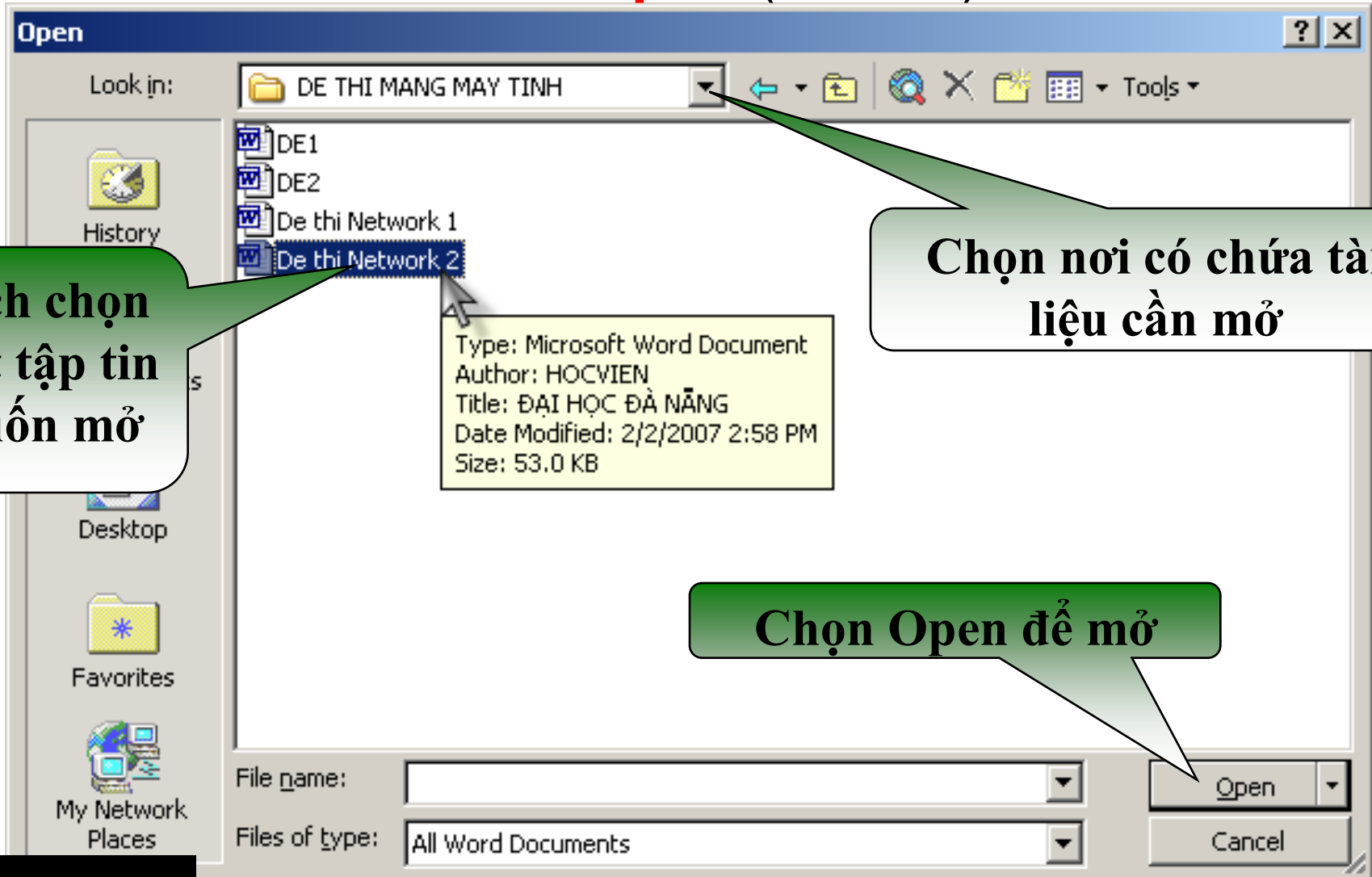
- Khi khởi động, MS-Word tự động tạo ra sẵn một tài liệu trắng với tên là **Document1**.
- Hoặc nhấn nút **New Blank Document** trên thanh công cụ.
- Hoặc nhấn tổ hợp phím **Ctrl+N**.
- Hoặc vào Menu **File**, chọn **New**.



New Blank Document

IV/ Mở tệp tin văn bản đã có

❑ Vào Menu **File**, chọn **Open** (**Ctrl+O**).



Kích chọn
một tệp tin
muốn mở

Chọn nơi có chứa tài
liệu cần mở

Chọn Open để mở

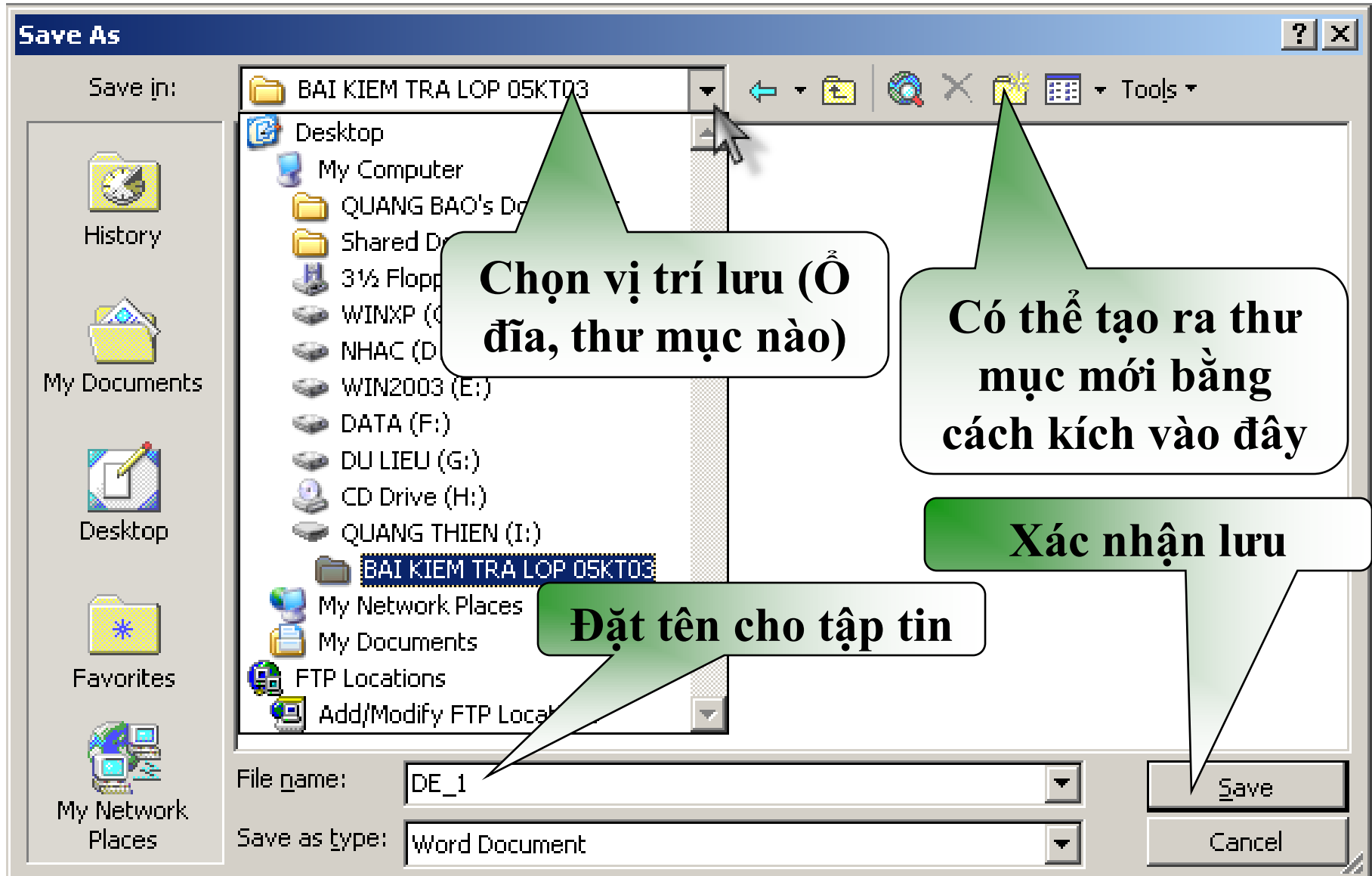


Chú ý:

Có thể chọn nhiều tệp tin để mở cùng một lúc.

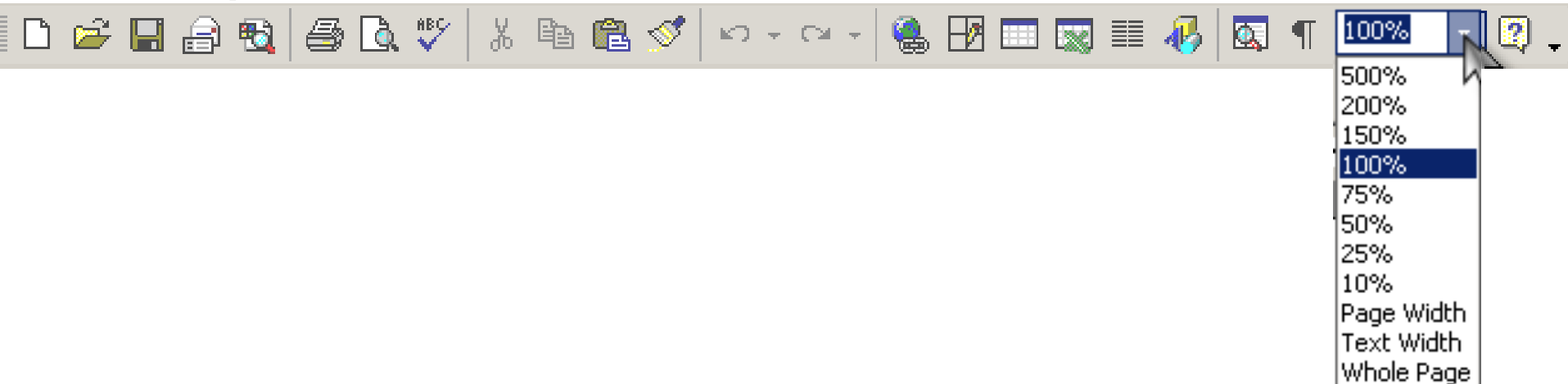
V/ Lưu văn bản

- ❑ Vào Menu **File**, chọn **Save** hoặc **Save As...**



VI/ Phóng to, thu nhỏ khung nhìn văn bản

- ❑ Mặc định khung nhìn tài liệu được đặt tỉ lệ 100%. Người dùng có thể tăng hoặc giảm tỉ lệ khung nhìn bằng công cụ **Zoom**.
- ❑ Thay đổi khung nhìn không làm thay đổi kích thước kí tự trên tài liệu khi in.
- ❑ **Cách thực hiện:**
 - Cách 1: Kích chuột vào menu **View**, chọn **Zoom**.
 - Cách 2: Kích chuột vào hộp **Zoom** trên thanh công cụ.



BÀI 2

CÁC THAO TÁC SOẠN THẢO VĂN BẢN

I/ Vấn đề gõ tiếng Việt trong văn bản

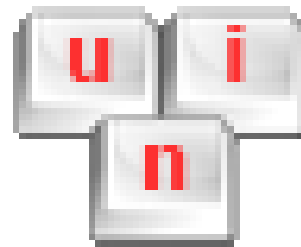
Để gõ được tiếng Việt trong trình soạn thảo văn bản Word và cũng như hầu hết tất cả các phần mềm chạy trên môi trường Windows ta phải cài đặt vào máy tính một chương trình phần mềm hỗ trợ gõ tiếng Việt.

Có rất nhiều phần mềm hỗ trợ gõ tiếng Việt khác nhau. Thường dùng nhất là 2 phần mềm: **VietKey** và **UniKey**.

Trong tất cả các bộ gõ tiếng Việt thì **bảng mã** và **Font chữ** đều phải phù hợp với nhau.



VietKeyNT



UniKeyNT

1/ Bảng mã và font chữ tiếng Việt

Bảng mã	Font
TCVN3 (Là bảng mã tiêu chuẩn quốc gia)	<ul style="list-style-type: none">- Font ABC- Tên các Font bắt đầu bằng dấu chấm và 2 chữ Vn- vd: .Vntime, .VnArial,...
VNI (do Vietnam International phát triển)	<ul style="list-style-type: none">- Font VNI- Tên các Font bắt đầu bằng chữ VNI.- vd: VNI-Times)
Bảng mã Unicode . Đây là bảng mã theo tiêu chuẩn quốc tế và được khuyến cáo nên dùng trong mọi phần mềm.	<ul style="list-style-type: none">- Các Font có sẵn trong mọi máy tính cài HĐH Windows- Vd: Times New Roman, Arial, Tahoma, Verdana,...

2/ Kiểu gõ tiếng Việt

□ Có 2 kiểu gõ phổ biến là **Telex** và **VNI**.

➤ Gõ tiếng việt theo kiểu **Telex**:

➤ Gõ tiếng việt theo kiểu **VNI**:

f	= huyền
s	= sắc
r	= hỏi
x	= ngã
j	= nặng
aa	= â
aw	= ă
oo	= ô
ow	= ơ
uw	= u
ee	= ê
dd	= đ

số 1	= dấu sắc
số 2	= dấu huyền
số 3	= dấu hỏi
số 4	= dấu ngã
số 5	= dấu nặng
số 6	= dấu mũ (â, ê)
số 7	= dấu râu (ơ, u)
số 8	= dấu trăng (ă)
số 9	= dấu ngang (đ)

II/ Các phím di chuyển con trỏ văn bản

- ✎ **Home:** Đưa con trỏ về đầu dòng hiện tại
- ✎ **End:** Đưa con trỏ về cuối dòng hiện tại
- ✎ **Ctrl + Home:** Đưa con trỏ đến vị trí đầu tiên của tài liệu
- ✎ **Ctrl + End:** Đưa con trỏ đến vị trí cuối cùng của tài liệu
- ✎ **Page Up:** Dịch chuyển con trỏ lên 1 trang màn hình
- ✎ **Page Down:** Dịch chuyển con trỏ xuống 1 trang màn hình

III/ Chế độ ghi chèn/ghi đè

- ❑ Chế độ ghi chèn: Ký tự nhập được chèn ngay tại vị trí con trỏ nhập, toàn bộ ký tự đằng sau con trỏ dịch chuyển sang phải.
- ❑ Chế độ ghi đè: Ký tự nhập được chèn ngay tại vị trí con trỏ nhập, con trỏ dịch chuyển sang phải và ghi đè lên ký tự liền sau nó.
- ❑ Phím dùng để chuyển đổi qua lại giữa 2 chế độ ghi chèn và ghi đè là phím **Insert**.

IV/ Chọn vùng làm việc

- ❑ **Chọn một câu:** Nhấn **Ctrl** và kích vào **vị trí bất kì** trong câu.
- ❑ **Chọn một dòng:** Nhấp vào **vị trí bên trái** của dòng.
- ❑ **Chọn một đoạn:** Nhấp đôi vào **khoảng trống bên trái của một dòng** trong đoạn.
- ❑ **Chọn một khối:** Nhấn **Shift** và **di chuyển chuột** từ **vị trí đầu đến cuối** đoạn.
- ❑ **Chọn một khối bất kì:** Nhấn giữ phím **Alt** và **kéo chuột**.
- ❑ **Chọn toàn bộ văn bản:** **Ctrl+A**



IV/ Chọn vùng làm việc (tt)

- Rê chuột từ điểm đầu đến điểm cuối khối văn bản cần chọn.
- Hoặc có thể để con nháy đầu khối văn bản cần chọn, bấm phím **Shift+Các phím mũi tên**.

V/ Sao chép và di chuyển văn bản

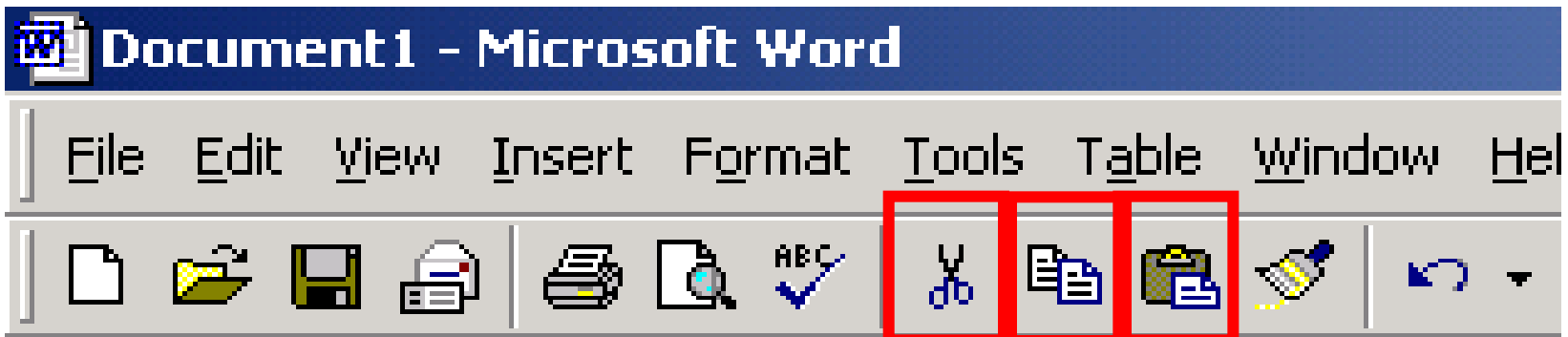
❖ Chọn vùng văn bản cần sao chép hay di chuyển.

☐ Sao chép:



- Kích chọn biểu tượng  hay vào **Edit**, chọn **Copy** (**Ctrl+C**).
- Di chuyển con trỏ văn bản đến vị trí cần dán.
- Kích chọn biểu tượng  hay vào **Edit**, chọn **Paste** (**Ctrl+V**).

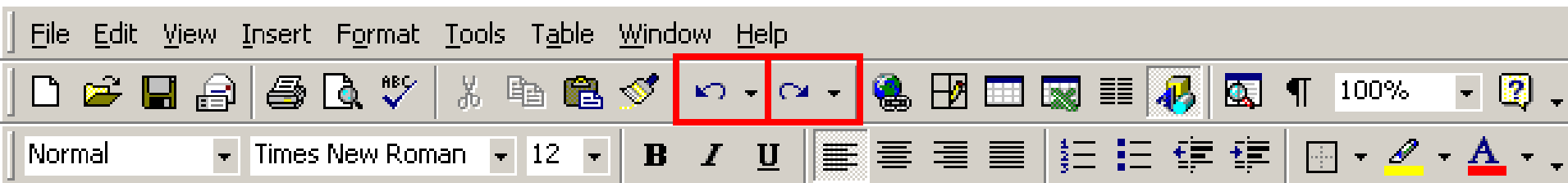
☐ Di chuyển:

- **Edit**, chọn **Cut** (**Ctrl+X**)
- Di chuyển con trỏ văn bản đến vị trí cần dán.
- **Edit**, chọn **Paste** (**Ctrl+V**).



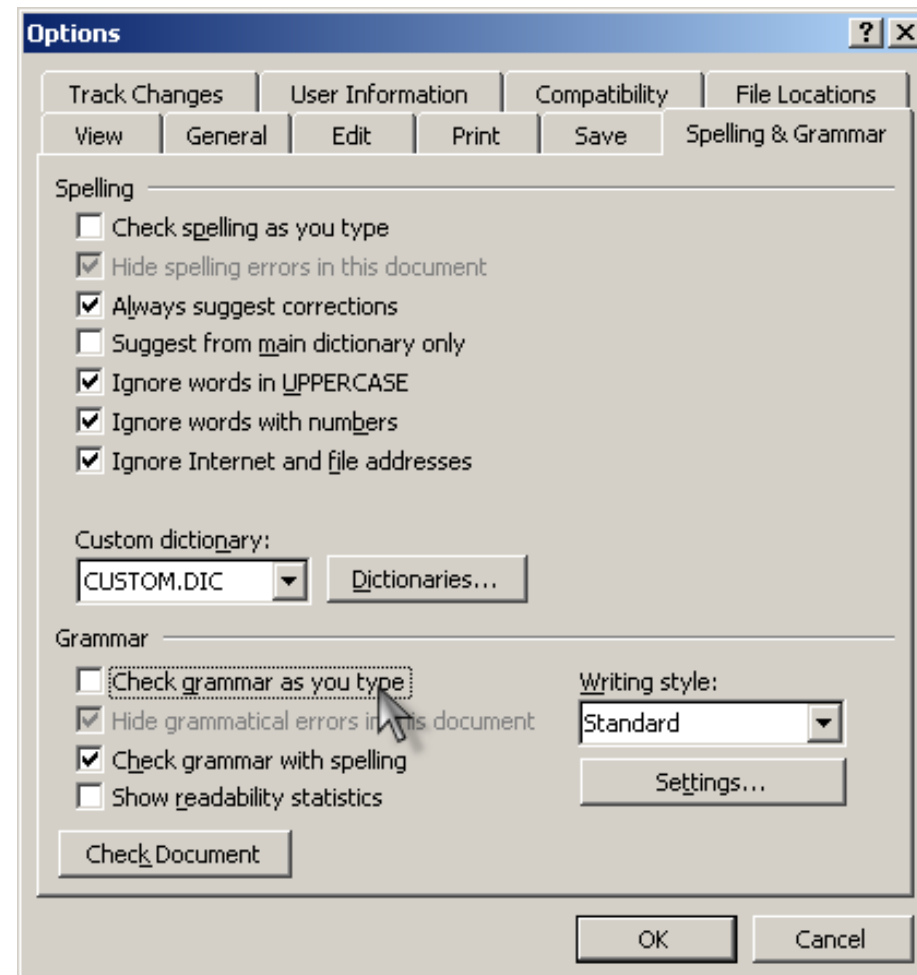
VI/ Sử dụng chức năng Undo và Redo

- ❑ Chức năng **Undo** dùng để quay về trước đó một thao tác.
 - Kích vào biểu tượng  hay nhấn tổ hợp phím **Ctrl+Z**
- ❑ Chức năng **Redo** ngược lại với chức năng Undo.
 - Kích vào biểu tượng  hay nhấn tổ hợp phím **Ctrl+Y**



VII/ Công cụ kiểm tra chính tả, sửa lỗi ngữ pháp tiếng Anh

- ❑ **Bật / tắt công cụ kiểm tra chính tả và ngữ pháp:**
Vào **Tools - Options** -> Chọn **Spelling & Grammar**.
Bỏ n **Check spelling as you type** và **Check Grammar as you type**.



VIII/ Định dạng trang in

- ❑ Thao tác này cho phép xác định chính xác vị trí của nội dung văn bản trong tờ giấy in.
- ❑ Vào menu File -> Page setup.

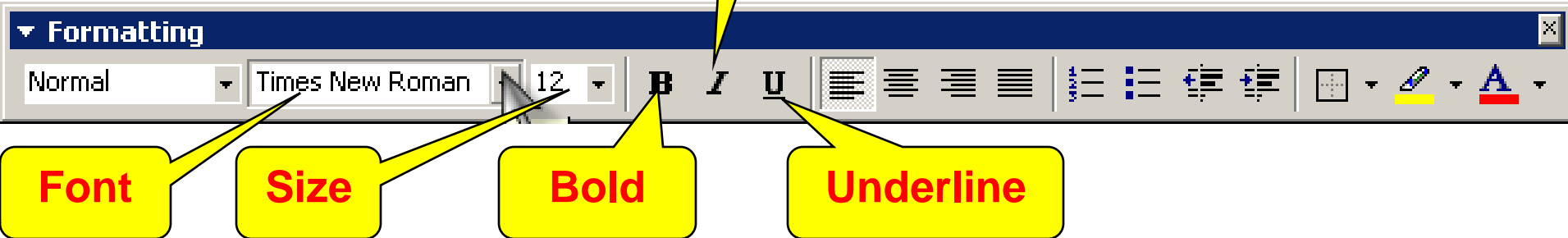


BÀI 3

CÁC THAO TÁC ĐỊNH DẠNG VĂN BẢN

I/ Định dạng ký tự

1/ Dùng thanh công cụ:




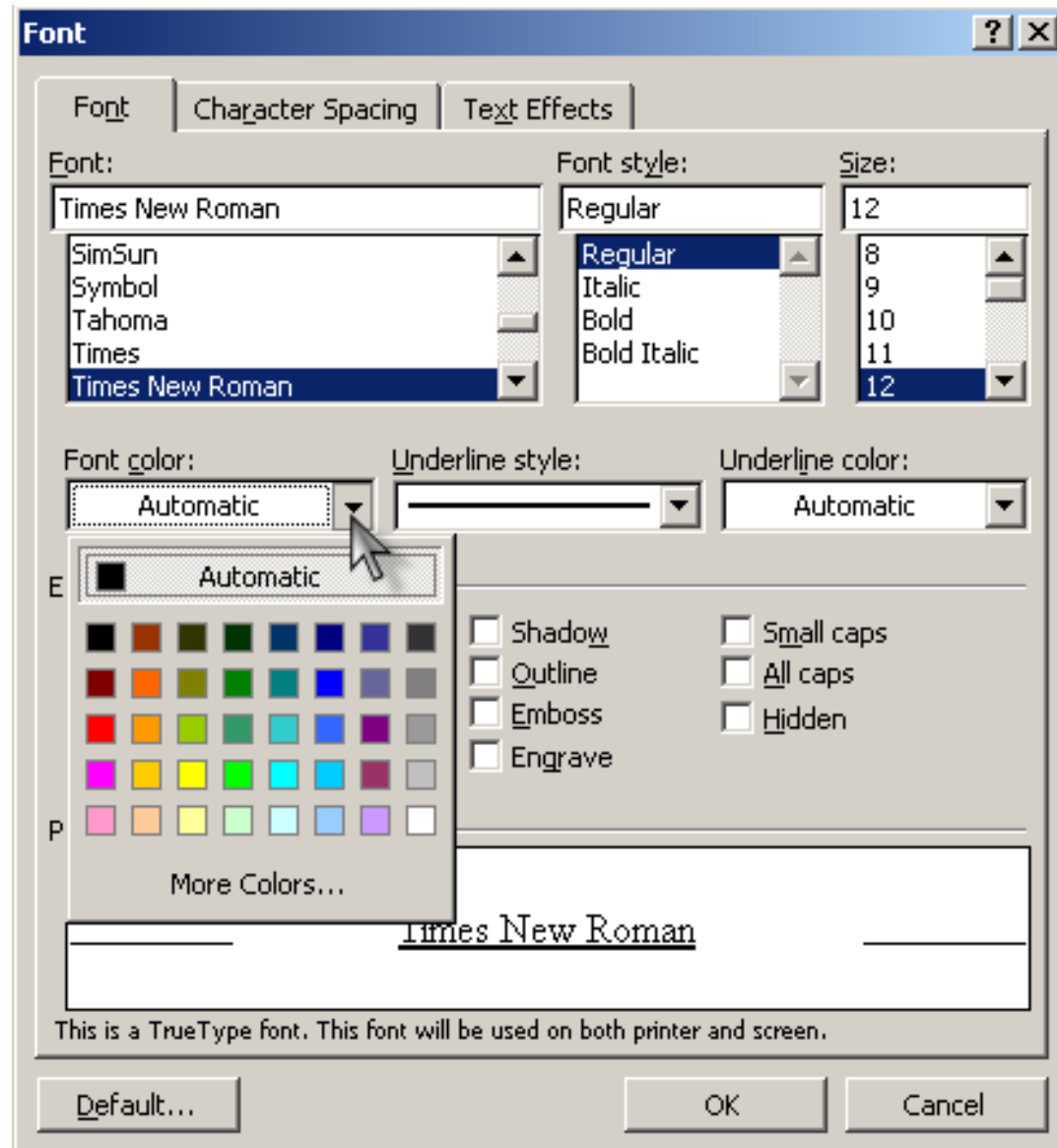
Chọn khối văn bản cần định dạng.

- ✓ **Font:** Font chữ
- ✓ **Size:** Kích cỡ chữ (**Ctrl+]**: Tăng size; **Ctrl+[**: Giảm size)
- ✓ **Bold:** In đậm (**Ctrl+B**)
- ✓ **Italic:** In nghiêng (**Ctrl+I**)
- ✓ **Underline:** Gạch dưới (**Ctrl+U**)

I/ Định dạng ký tự (tt)

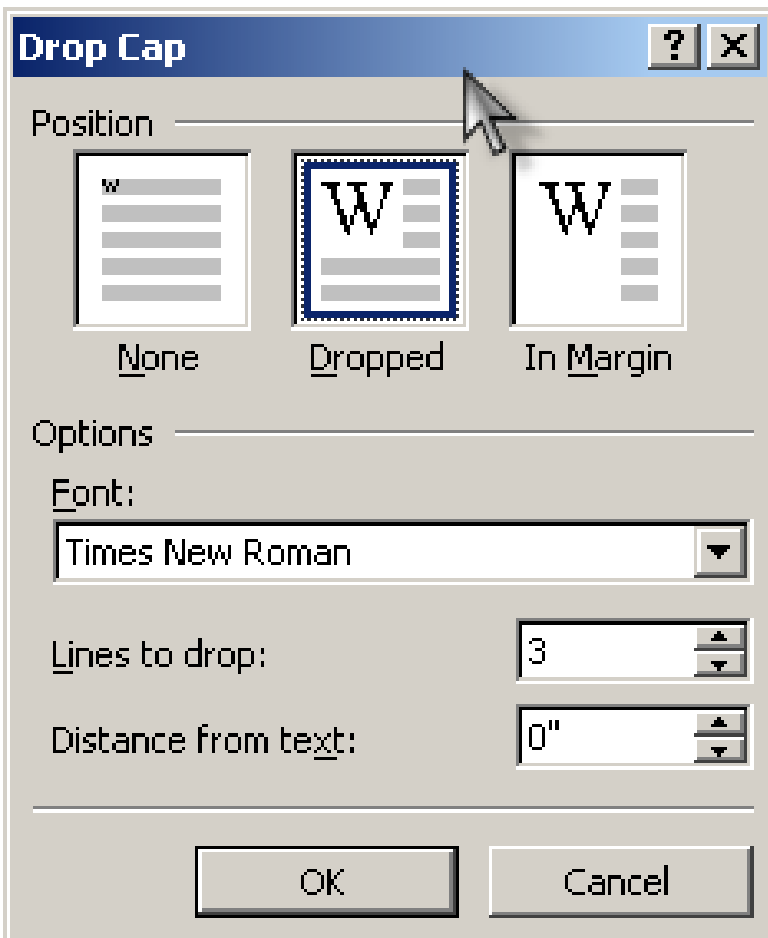
2/ Dùng menu **Format \ Font...**

 Ngoài ra có thể chọn hiển thị văn bản lên màn hình “sinh động” bằng cách chọn Tab **Text Effects**.



II/ Áp dụng chữ to đầu đoạn (Drop Cap)

- Đặt con trỏ trong đoạn văn.
- Mở menu **Format**, chọn **Drop Cap**



Cuối cùng anh cũng đã nhận được thiệp mời cưới của em, thật sự trong anh có điều gì đó thật đau đớn. Mình xa nhau được 8 tháng rồi em nhỉ, 4 năm yêu nhau quá ngắn để níu bước em đi nhưng quá dài để anh có thể quên đi hình bóng của em, thật lòng anh cầu chúc cho em được hạnh phúc trên con đường mà em đã chọn.

Mong rằng người đó sẽ mang cho em thật nhiều hạnh phúc - điều mà anh không làm được khi em ở bên anh. Anh biết rằng anh sẽ nhớ em nhiều lắm đấy Vịt Con của anh ạ, hãy để nụ cười luôn luôn ngự trị trên môi em bởi vì ở một nơi nào đó vẫn có một người luôn nhớ đến em và luôn mong em được hạnh phúc. Tạm biệt em! Vịt Con Xấu Xí...

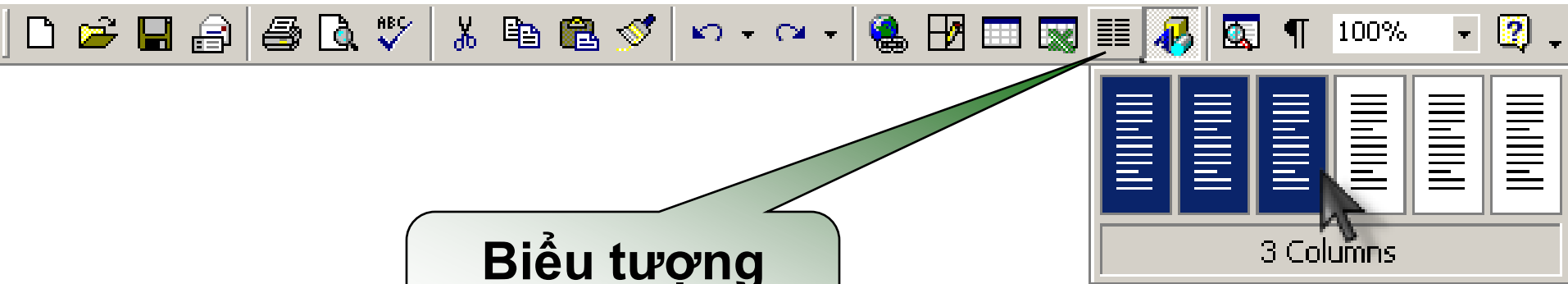
II/ Áp dụng chữ to đầu đoạn (tt)

- Chọn kiểu chữ:
 - **Dropped:** Chữ cái được căn lề giống các từ còn lại của văn bản.
 - **In Margin:** Chữ cái nằm ngoài lề đoạn văn bản.
- Chọn font chữ: Nhấn vào hộp **Font** rồi chọn font cho chữ cái đầu đoạn văn.
- Chọn độ lớn của kí tự đầu đoạn: Nhập số chỉ dòng vào ô **Lines to drop.**
- Nhập khoảng cách từ đầu đoạn đến phần còn lại của đoạn vào hộp **Distance from text.**

III/ Tạo cột văn bản

1) Bảng thanh công cụ Standar:

- Chọn một hoặc nhiều đoạn văn bản cần chia cột.
- Bấm chọn biểu tượng lệnh **Column** trên thanh công cụ, chuyển chuột sang phải để chọn số cột cần tạo rồi thả chuột.



III/ Tạo cột văn bản (tt)

2) Bằng Menu: Vào menu Format – Columns.

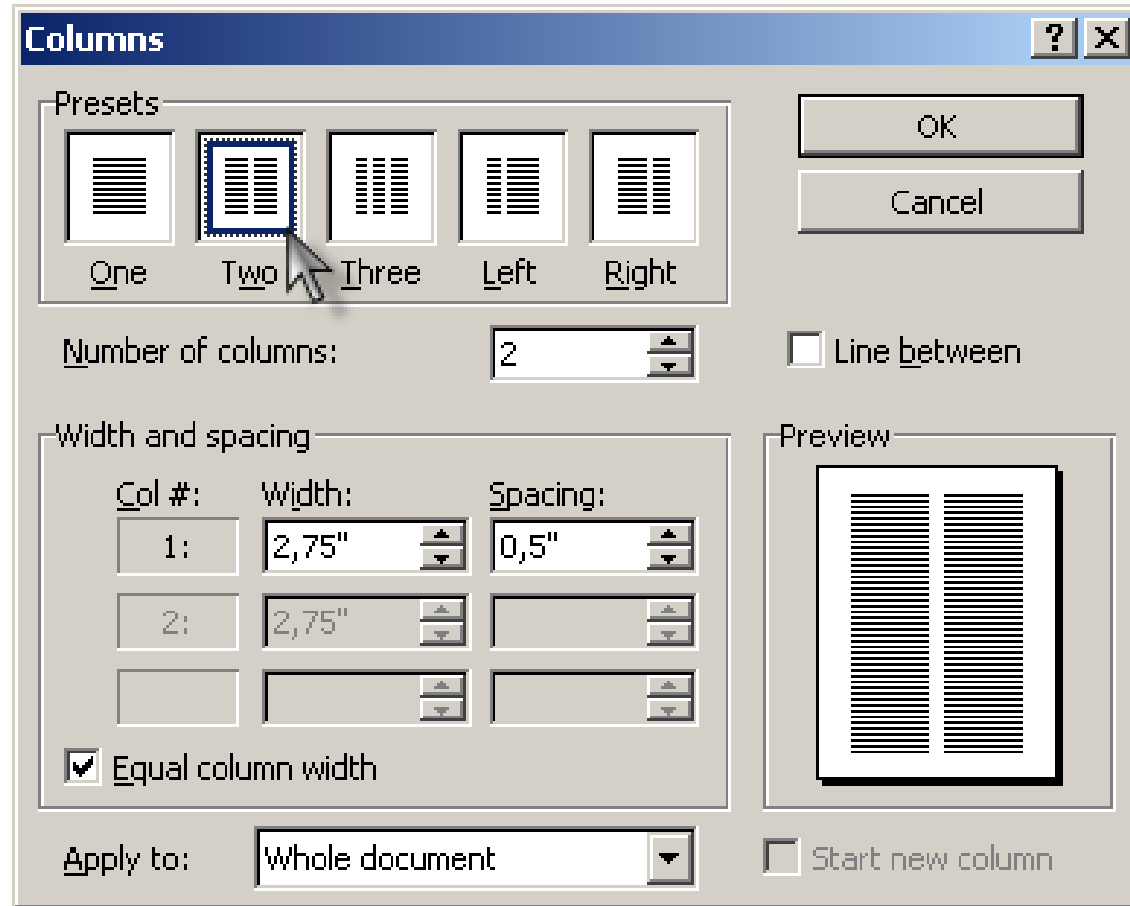
➤ Trong khung **Presets** chọn một trong các kiểu cột có sẵn.

➤ Hoặc có thể gõ trực tiếp số cột trong ô

Number of columns.

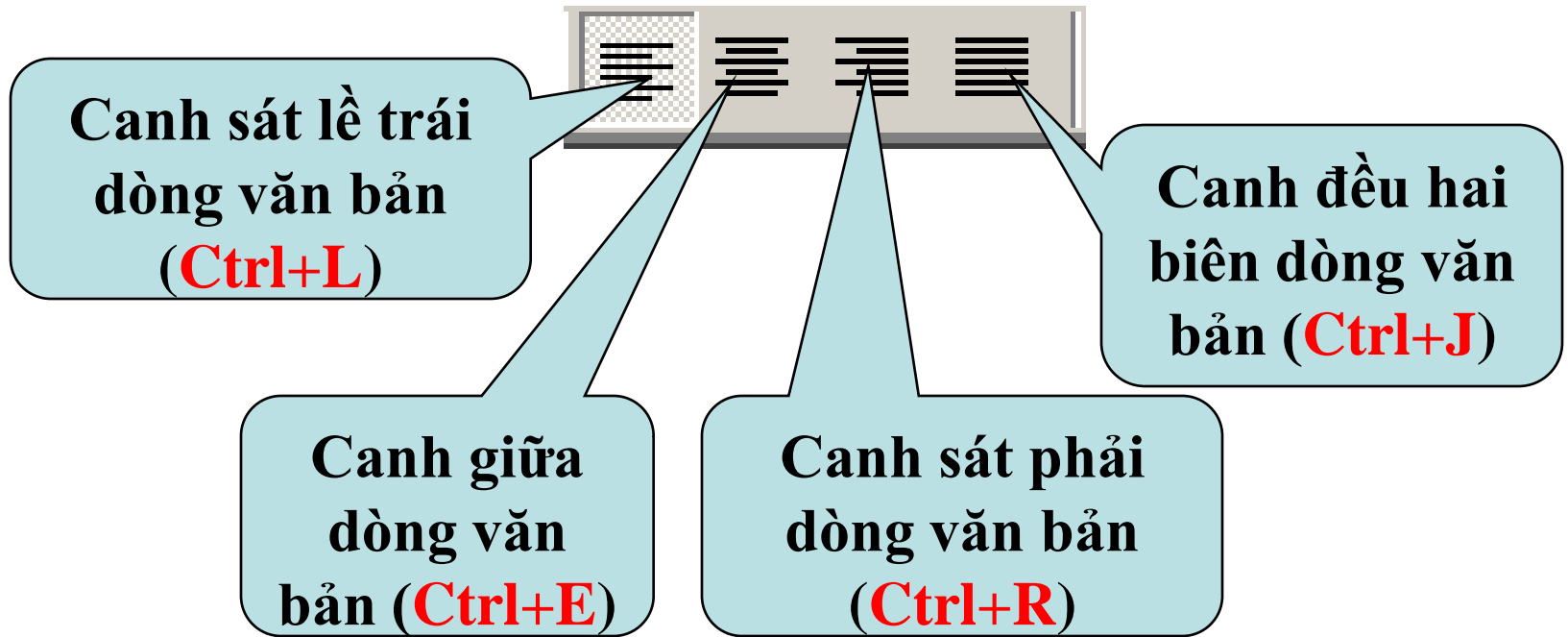
➤ Có thể chọn độ rộng và khoảng cách giữa các cột trong mục **Width and spacing.**

➤ Nếu muốn giữa các cột có đường kẻ phân cách thì đánh dấu vào Checkbox **Line between.**



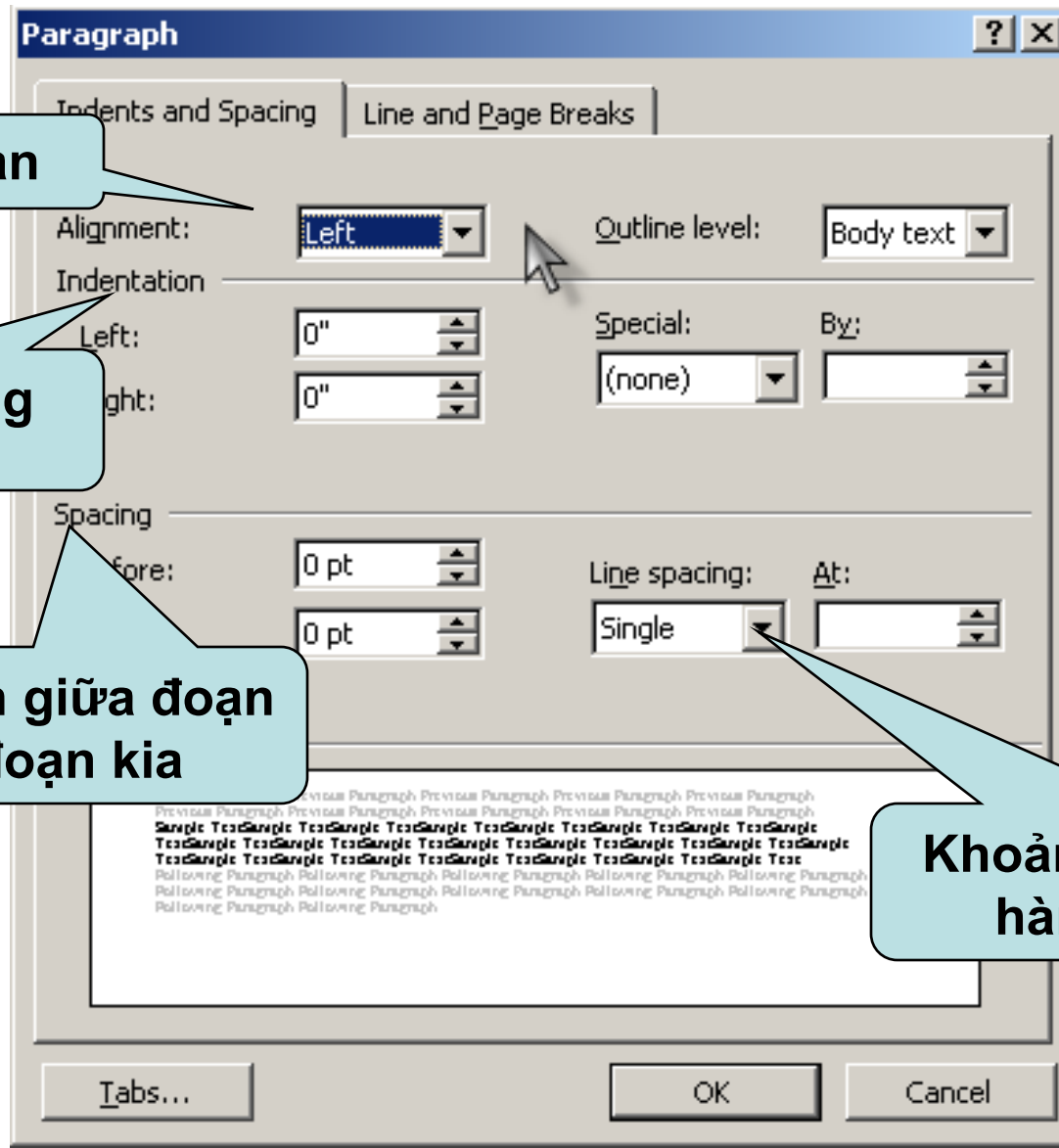
IV/ Định dạng đoạn văn bản

- ❑ Các đoạn văn trong tài liệu phân biệt bởi ký hiệu kết thúc đoạn (Khi nhấn **Enter**).
- ❑ Đặt con trỏ trong đoạn cần định dạng.
 - **Sử dụng thanh công cụ:**



IV/ Định dạng đoạn văn bản (tt)

Sử dụng Paragraph: Format \ Paragraph



Canh lề đoạn

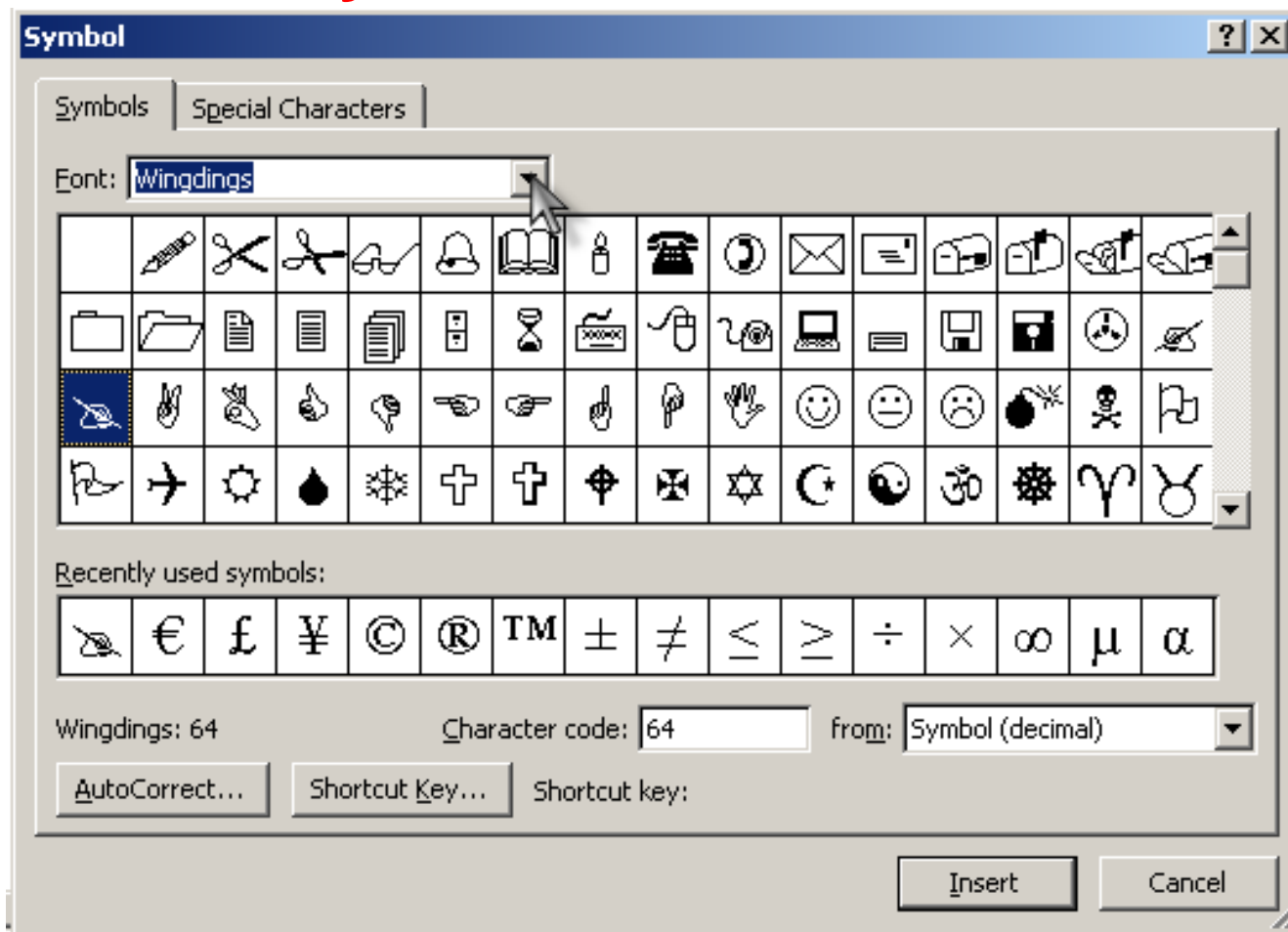
Định khoảng cách lề

Khoảng cách giữa đoạn này với đoạn kia

Khoảng cách giữa các hàng trong đoạn

V/ Chèn ký tự đặc biệt

❑ Vào **Insert** → **Symbol**



❑ **Font:** Danh sách các nhóm ký hiệu.

❑ **Shortcut Key:** Định nghĩa tổ hợp phím tắt cho ký hiệu đã chọn.

VI/ Định mốc dừng (Tab Stop)

☐ Ví dụ về sử dụng Tab Stop:

VĂN PHÒNG UBND
26 HÙNG VƯƠNG
====ooo====

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc
soos□soos

GIẤY MỜI

Kính gửi:

.....

Địa chỉ:

.....

.....

Đúng vào lúc:giờ, ngàythángnăm 2005. Mời

ông(bà) về tại văn phòng UBND tỉnh để dự cuộc họp.

....., ngày 05 tháng 06 năm 2005.

VI/ Định mốc dừng (tt)

- Dấu Tab được đặt ở đầu mút trái trên thanh thước kẻ ngang.



Kí hiệu biểu diễn dấu Tab trên thanh thước kẻ ngang

- Các loại Tab thường dùng:

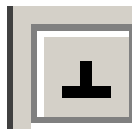
➤ Tab trái:



➤ Tab phải:



➤ Tab giữa:

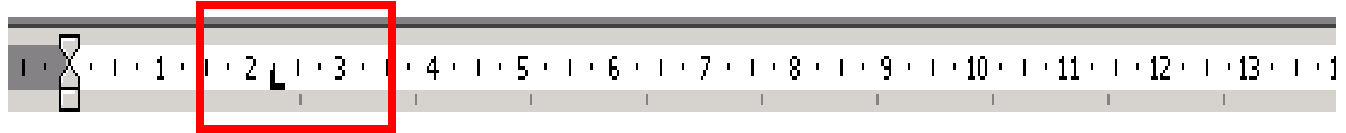


VI/ Định mốc dừng (tt)

□ Cách đặt mốc dừng:

➤ Nhấn vào **biểu tượng dấu Tab** cho đến khi hiện biểu tượng dấu **Tab** thích hợp.

➤ **Nhấn chuột** vào vị trí đặt Tab **trên thanh thước ngang**



➤ Nhấn vào **phím Tab trên bàn phím** để con nháy chuột nhảy đến vị trí đặt Tab trên vùng soạn thảo.

➤ Gõ văn bản vào, văn bản sẽ được canh theo Tab.

VI/ Định mức dừng (tt)

Chúng tôi yêu nhau một cách tự
nhiên như vốn dĩ nó phải thế,
mặc dù ngay từ đầu chúng tôi đã cố
gắng kiềm chế bởi hơn ai
hết cả 2 chúng tôi đều hiểu

Công ty TNHH ABC
Địa chỉ: 234 Ngô Gia Tự - Thành Phố Huế
Email: abc@yahoo.com
--o0o--

Cộng hòa xã hội chủ nghĩa Việt Nam
Độc lập - Tự do - Hạnh phúc
Đơn xin nghỉ học
Kính gửi ông hiệu trưởng trường Đại Học

VI/ Định mốc dừng (tt)

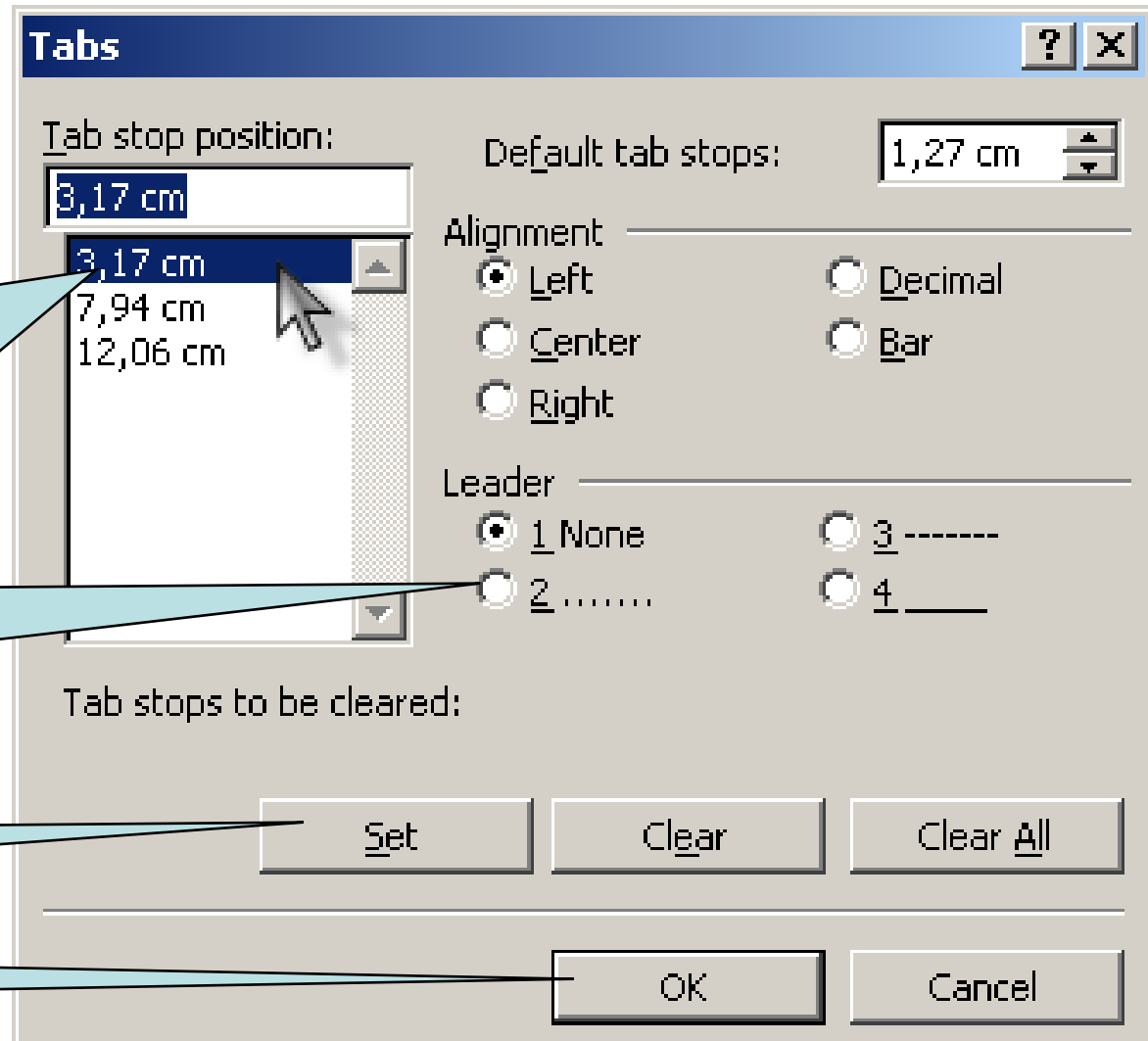
- ❑ Chèn ký hiệu phía trước dấu Tab: Vào menu **Format**, chọn **Tabs**.

Chọn Tab muốn làm xuất hiện ký hiệu

Chọn kiểu ký hiệu để xuất hiện

Chọn Set

Chọn OK

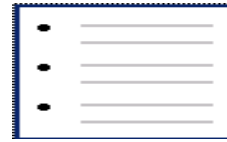


VII/ Danh sách liệt kê

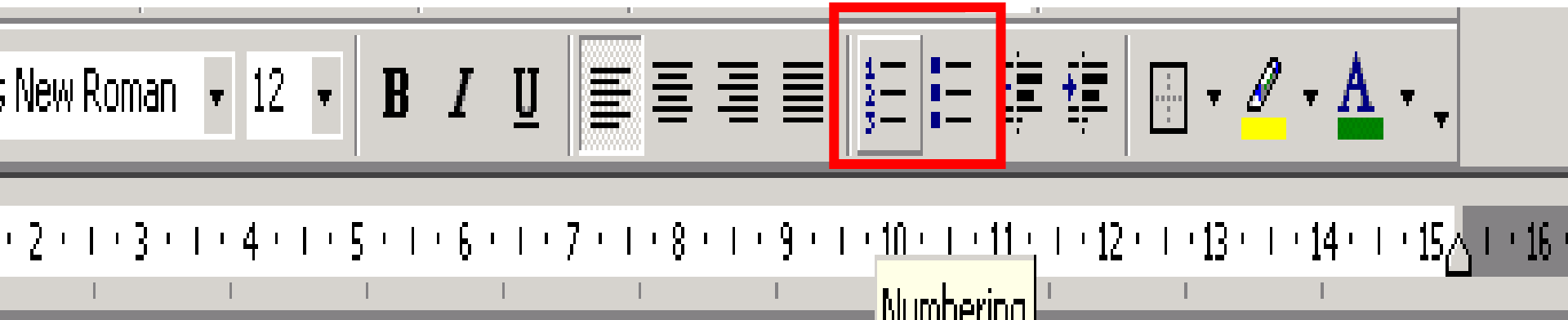
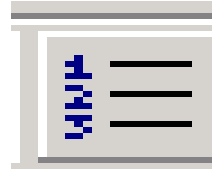
□ Danh sách liệt kê là một dãy liên tiếp các đoạn văn bản được đánh dấu bằng các kí hiệu đặt đầu dòng.

□ Có nhiều cách để tạo:

➤ Sử dụng nút **Bullets**



➤ Hay nút **Numbering**



➤ Hoặc từ lệnh **Format → Bullets and Numbering.**

VII/ Danh sách liệt kê (tt)

□ Tùy chỉnh danh sách liệt kê: Kích vào nút **Customize**

Đánh số thứ tự từ số mấy.

Thay đổi định dạng của số

Kiểu chỉ mục thứ tự

Đổi cách canh lề

Khoảng cách từ lề giấy đến đoạn văn

Khoảng cách từ kí hiệu đến đoạn văn

The image shows a 'Customize Numbered List' dialog box with several fields and callouts:

- Number format:** A text box containing '3/' and a 'Font...' button. Callout: 'Thay đổi định dạng của số'.
- Number style:** A dropdown menu showing '1, 2, 3, ...'. Callout: 'Kiểu chỉ mục thứ tự'.
- Start at:** A spinner box set to '3'. Callout: 'Đánh số thứ tự từ số mấy.'.
- Number position:** A dropdown menu set to 'Left'. Callout: 'Đổi cách canh lề'.
- Aligned at:** A spinner box set to '1,1 cm'. Callout: 'Khoảng cách từ kí hiệu đến đoạn văn'.
- Text position:** A spinner box set to '1,27 cm'. Callout: 'Khoảng cách từ lề giấy đến đoạn văn'.
- Buttons:** 'OK' and 'Cancel' buttons.
- Preview:** A preview window showing a list with three items: '3/ _____', '4/ _____', and '5/ _____'.

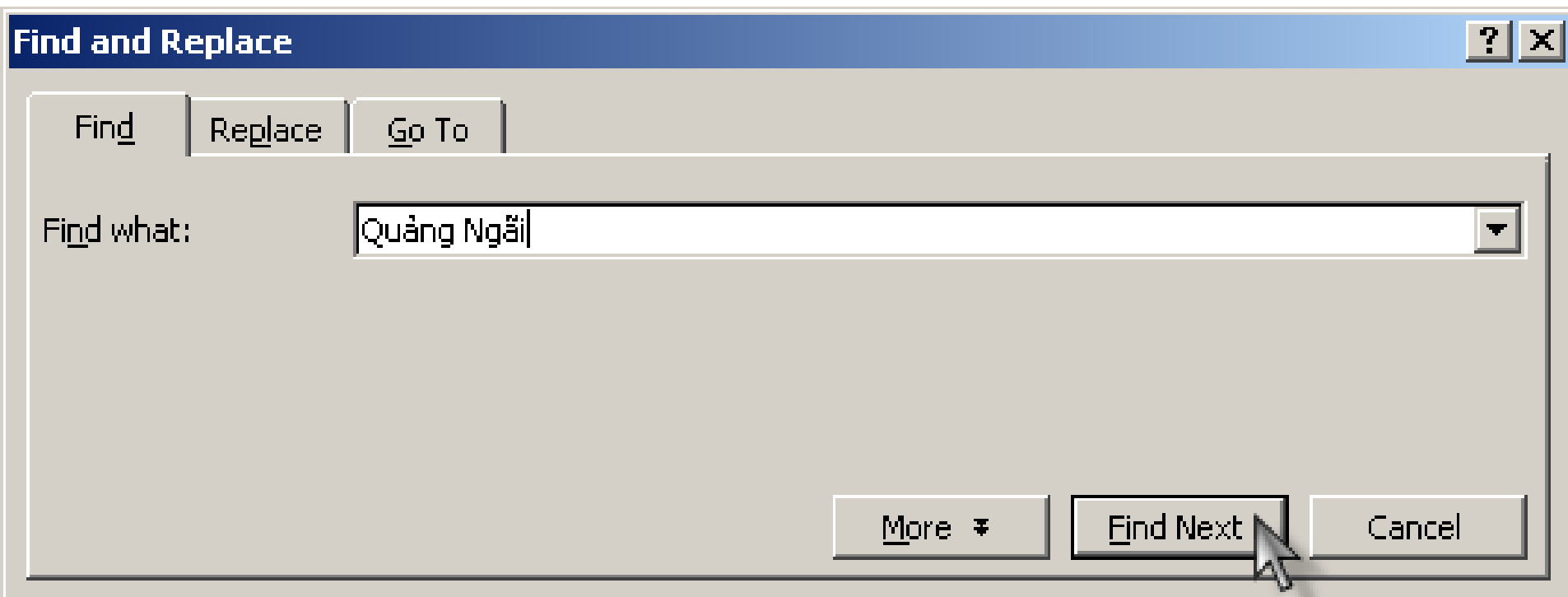
BÀI 4

SỬ DỤNG CÁC CÔNG CỤ HIỆU CHỈNH VĂN BẢN

I/ Tìm kiếm và thay thế

1/ Tìm kiếm:

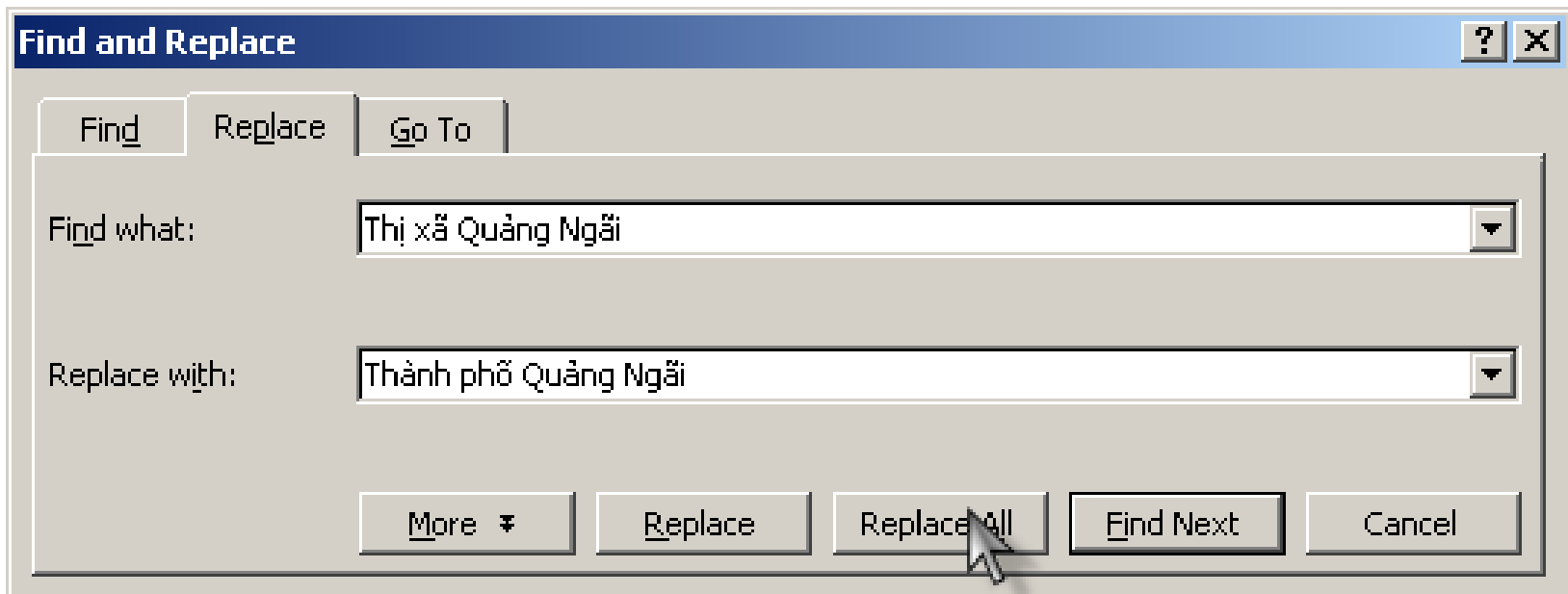
- Vào menu **Edit**, chọn **Find...**
- Trong khung **Find what** gõ vào đoạn văn muốn tìm.
- Kích nút **Find next** để tìm tiếp. Khi tìm thấy thì đoạn văn đó sẽ được bôi đen.



I/ Tìm kiếm và thay thế (tt)

2/ Thay thế:

- Vào menu **Edit**, chọn **Replace...**
- Nhập văn bản cần thay thế vào khung **Find what**
- Nhập văn bản thay vào khung **Replace with**
 - **Replace:** Thay từng phần tử được tìm thấy.
 - **Replace all:** Thay thế tất cả.
 - **Find next:** Không thay, tìm tiếp.



II/ Tiện ích AutoCorrect

1/ Tạo một mục Autocorrect: Vào menu **Tools**, chọn **AutoCorrect...**

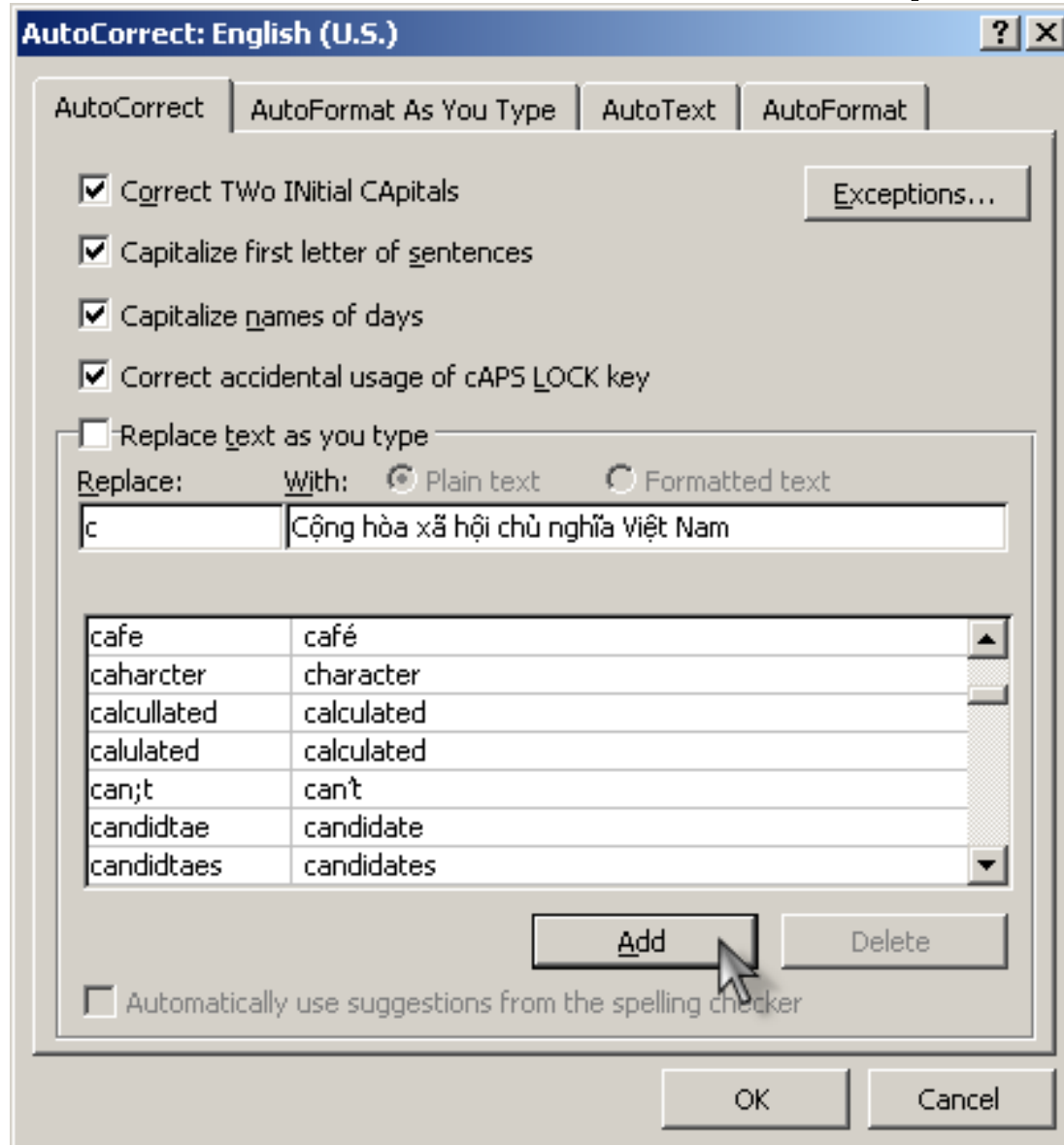
➤ Đánh dấu kiểm vào hộp chọn: **Replace text as you type**

➤ Nhập tên mục **AutoCorrect** vào khung **Replace**

➤ Nhập nội dung vào khung **With**

➤ Nhấn **Add**

➤ Nhấn **OK**



II/ Tiện ích AutoCorrect (tt)

- 2/ Sử dụng Autocorrect:** Trong khi soạn thảo, nhập tên mục **Autocorrect**, nhấn **Spacebar**.
- 3/ Xóa một Autocorrect:** Vào menu **Tolls**, chọn **Autocorrect**. Chọn tên **Autocorrect** cần xóa, nhấn nút **Delete**.

III/ Tiện ích Autotext

1/ Tạo mới một mục Autotext:

- Chọn nội dung muốn lưu trữ (bao gồm văn bản hay đồ họa).
- Vào menu **Insert**, chọn **Autotext...**
- Gõ tên Autotext vào mục **Enter Autotext Entries here**. Nhấn **Add**.

2/ Sử dụng Autotext:

- Đưa con trỏ văn bản đến vị trí cần chèn **Autotext**, gõ ký tự đã định nghĩa ở trên và nhấn phím **F3**.

IV/ Tiêu đề đầu trang, tiêu đề chân trang

- ❖ Vào menu **View**, chọn **Header and Footer...**

Header

Giáo trình Cấu trúc dữ liệu và Giải thuật

▼ Header and Footer

Insert AutoText ▼



Close

Số thứ tự trang

Tổng số trang

Chuyển đổi giữa tiêu đề đầu trang với tiêu đề chân trang

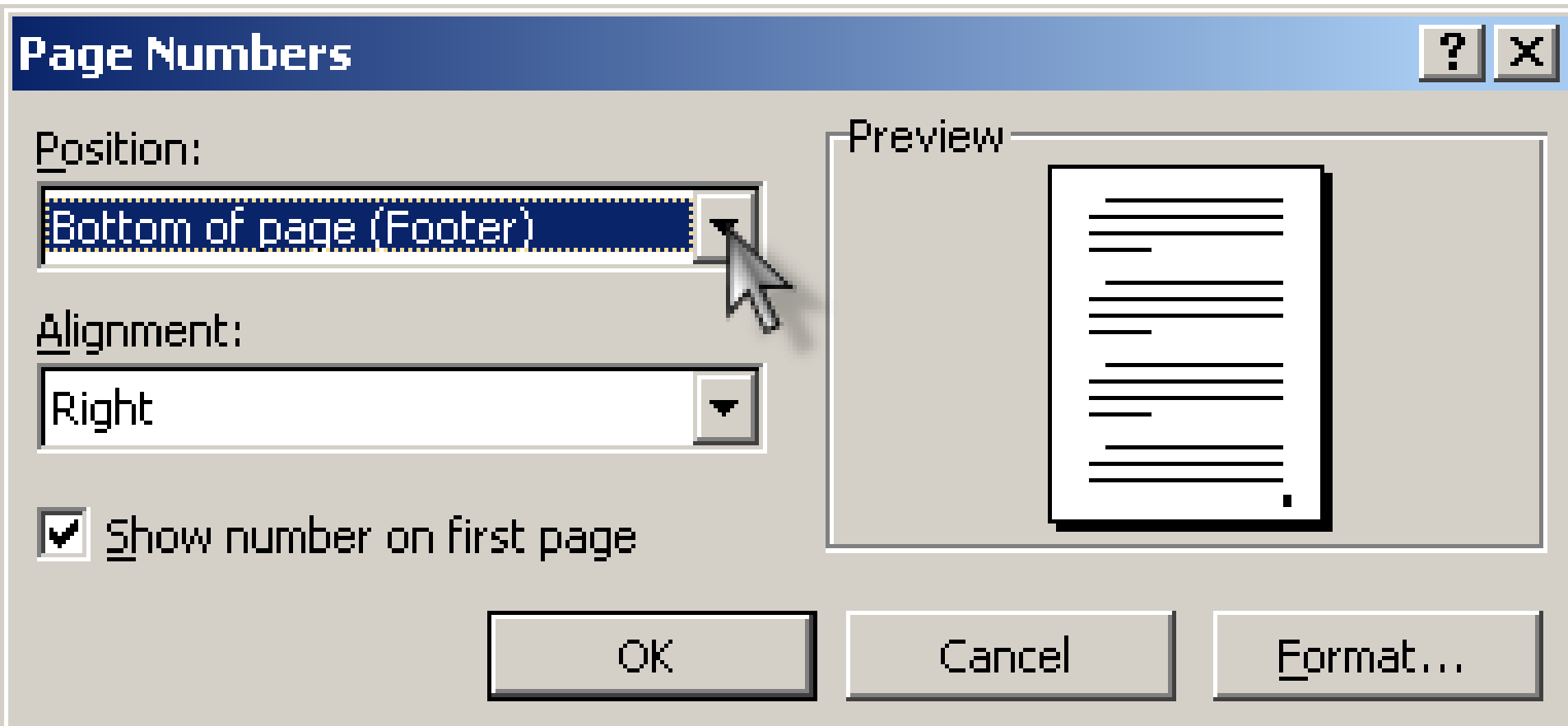
Footer

Khoa CNTT - Đại học Bách khoa Tp.Hồ Chí Minh

Trang 1

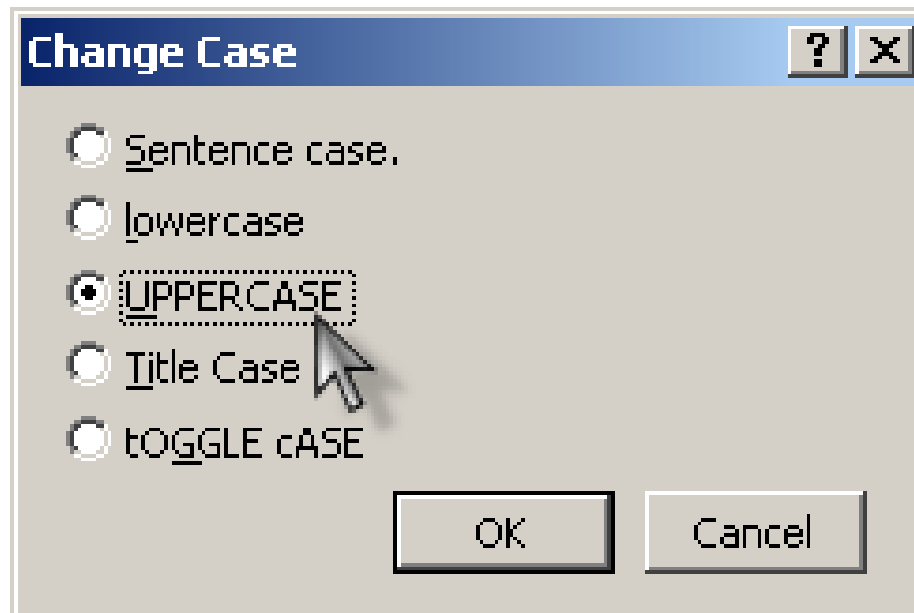
V/ Đánh số trang

- ❖ Vào menu **Insert**, chọn **Page numbers...**
- ❖ Chọn vị trí trong khung **Position**
- ❖ Chọn cách canh trong khung **Alignment**



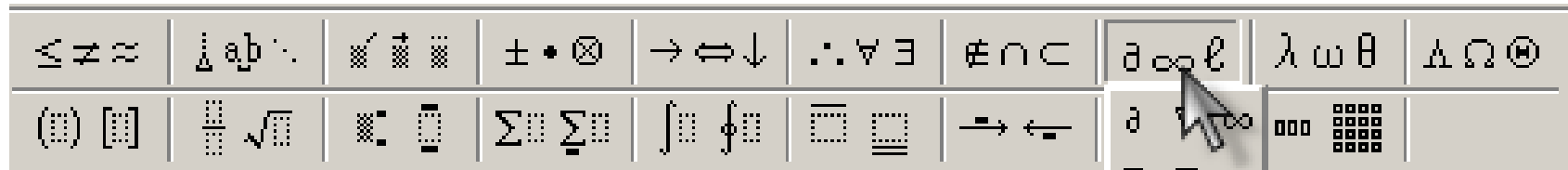
VII/ Sử dụng tiện ích Change case

- Chọn đoạn văn bản cần **Change case**.
- Vào menu **Format**, chọn **Change case...**
 - **Sentence case**: Viết hoa kí tự đầu câu.
 - **Lower case**: Tất cả là chữ thường.
 - **Upper case**: Tất cả đều là chữ in.
 - **Title case**: Viết hoa ký tự đầu mỗi từ.
 - **Toggle case**: Ngược với Title case.



VII/ Nhập công thức toán học

- ❑ Sử dụng công cụ Microsoft Equation 3.0
- ❑ Vào **Insert - Object... - Microsoft Equation 3.0 - OK**
Xuất hiện một thanh c sau:



$$J = \int \frac{1}{\sqrt{2} \sqrt[3]{x(e^x - e^{-x})}} dx$$

$$\frac{\partial^2 z}{\partial x^2} - a^2 \frac{\partial^2 z}{\partial y^2} = 0$$

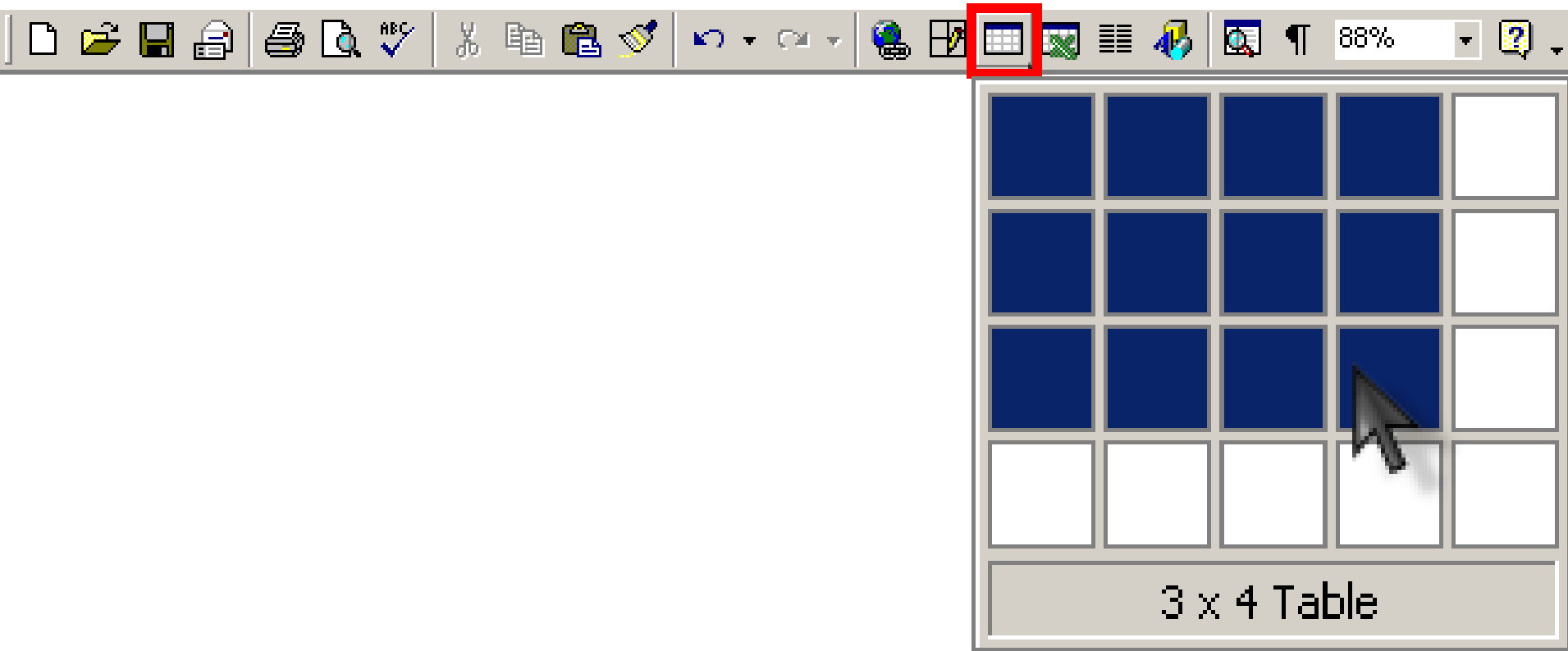
BÀI 5

BẢNG BIỂU VÀ HÌNH ẢNH

I/ Tạo mới một bảng biểu

1/ Sử dụng thanh công cụ:

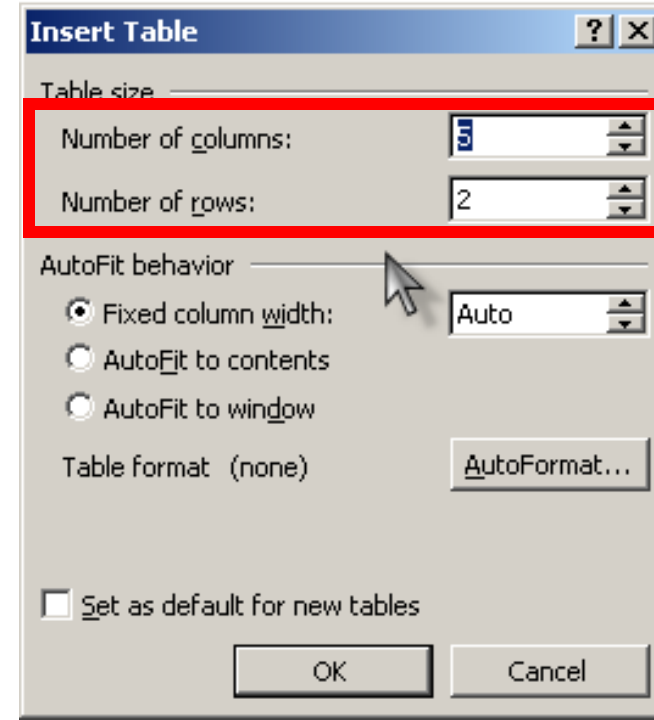
- Dời con trỏ đến nơi cần tạo bảng biểu.
- Bấm chọn biểu tượng **Insert Table** – Rê chuột chọn số hàng cột cần thiết.



I/ Tạo mới một bảng biểu (tt)

2/ Dùng menu:

- Dời con trỏ đến nơi cần tạo bảng biểu.
- Vào menu **Table**, chọn **Insert\ Table...**
 - Nhập số cột vào khung **Number of Columns**
 - Nhập số dòng vào khung **Number of Rows**
 - Trong khung **Column width** nhập vào độ rộng cột - thường chọn **Auto**. Nhấn OK



II/ Di chuyển con nháy trong bảng

- **Tab:** Dời con trỏ sang ô bên phải so với ô hiện tại.
- **Shift+Tab:** Dời con trỏ sang ô bên trái so với ô hiện tại.
- **Các phím mũi tên:** Di chuyển con trỏ theo chiều tương ứng.
- **Alt+Home:** Về ô đầu tiên trong hàng.
- **Alt+End:** Về ô cuối cùng trong hàng.
- **Alt+Page Up:** Về ô đầu tiên trong cột.
- **Alt+Page Down:** Về ô cuối cùng trong cột.
- **Ctrl+Tab:** Đưa con trỏ đến vị trí Tab Stop đã định.

III/ Chọn một dòng, cột hay ô

- Nhấn chuột vào *lề trái* của một dòng để chọn được cả dòng.
- Di chuyển con trỏ chuột vào đến *mép lề trên đỉnh cột*, khi con trỏ chuột có dạng *mũi tên màu đen* thì nhấn chuột *chọn cả cột*.
- Di chuyển con trỏ chuột đến *lề trái* của ô đến khi thành *mũi tên màu đen* thì nhấn chuột để *chọn một ô*. *Nếu chọn nhiều ô thì nhấn chuột kéo và thả*.
- Ta có thể chọn nhiều dòng hay nhiều cột bằng cách kích chuột *kéo và thả*

IV/ Thay đổi cấu trúc của bảng

- 1) Thêm dòng:** Vào **Table – Insert - Rows Above** hay **Rows Below** (Dòng mới thêm vào sẽ chèn vào **phía trên** hay **phía dưới** của dòng đang chứa con trỏ).
- 2) Thêm cột:** Vào **Table – Insert - Columns to the left** hay **Columns to the right** (Cột mới thêm vào sẽ chèn vào **bên trái** hay **bên phải** của cột có chứa con trỏ).
- 3) Thêm ô:** Vào **Table - Insert - Cells -**

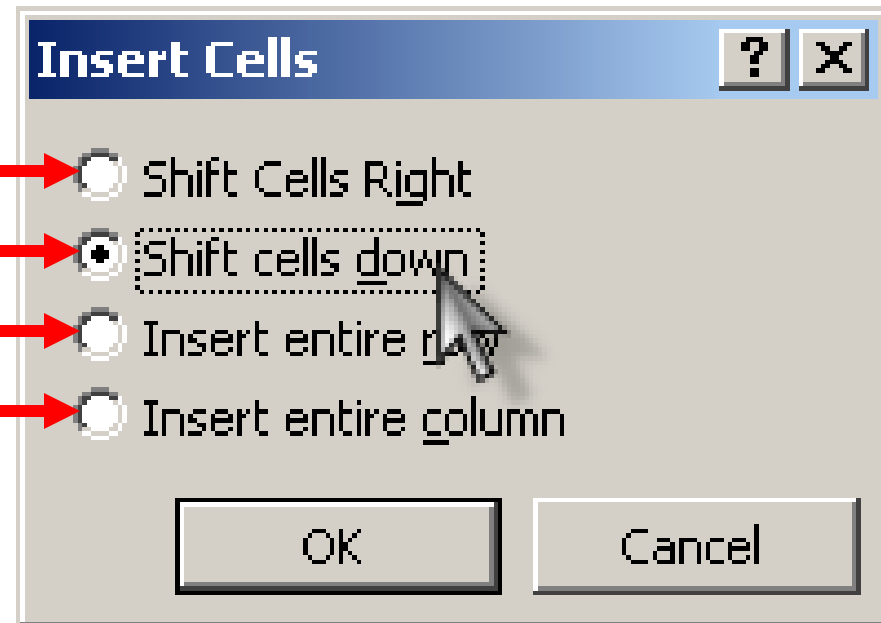
:

Dịch các ô về phía phải

Dịch các ô xuống dưới

Chèn cả hàng

Chèn cả cột



IV/ Thay đổi cấu trúc của bảng (tt)

4) Xóa dòng, cột hay cả bảng:

- Chọn các dòng, cột hay bảng cần xóa.
- Vào **Table - Delete - Columns/Rows/Table** (Xóa cột / dòng / bảng).

5) Xóa ô:

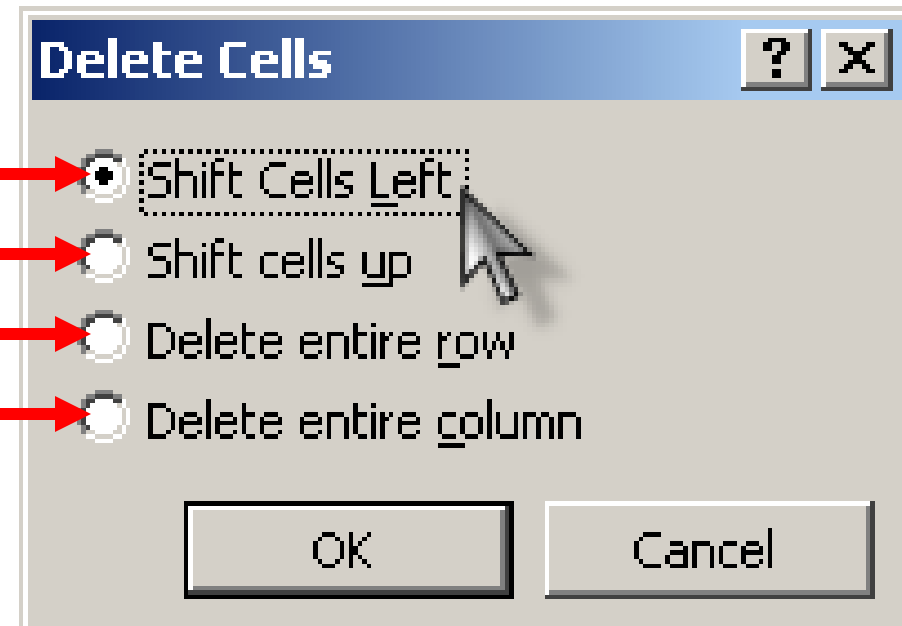
- Chọn các ô muốn xóa.
- Vào **Table - Delete - Cells -** :

Dịch các ô còn lại về phía trái

Dịch các ô còn lại xuống dưới

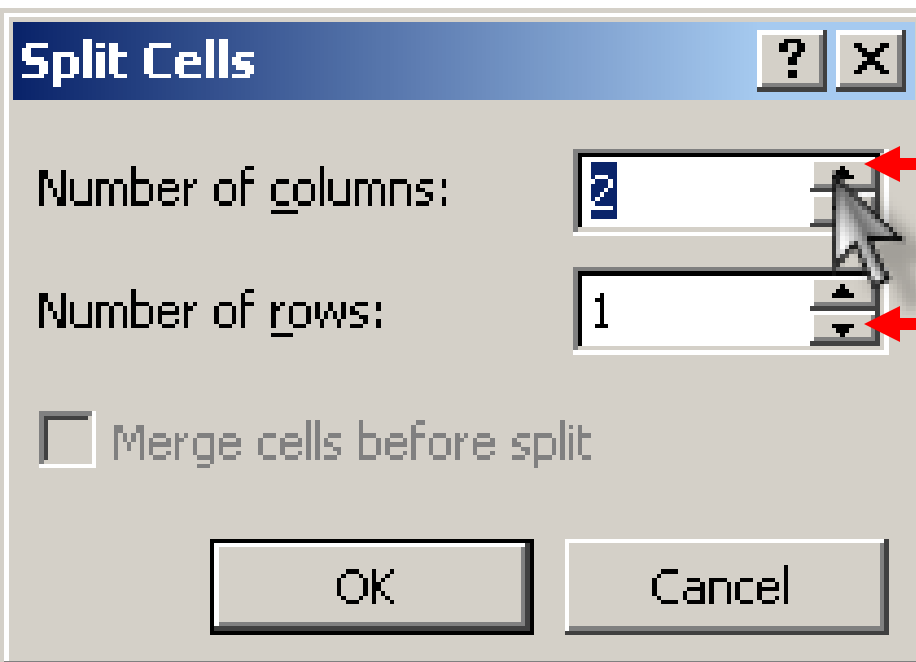
Xóa cả hàng

Xóa cả cột



IV/ Thay đổi cấu trúc của bảng (tt)

- 6) **Trộn nhiều ô thành một ô:** Chọn các ô muốn trộn. Vào menu **Table – Merge Cells**.
- 7) **Chia một ô thành nhiều ô:** Chọn ô muốn chia. Vào menu **Table – Split Cells**. Xuất hiện hộp thoại:



Nhập vào số cột cần chia ra từ ô đã chọn

Nhập vào số hàng cần chia ra từ ô đã chọn

IV/ Thay đổi cấu trúc của bảng (tt)

8) Chia một bảng thành nhiều bảng:

- Đặt con trỏ trong hàng mà ta muốn là hàng đầu tiên của bảng mới.
- Vào menu **Table – Split Table**.

9) Thay đổi độ rộng của cột, chiều cao của hàng:

- Di chuyển con trỏ chuột đến đường biên của cột hoặc hàng cần thay đổi, khi con trỏ chuột biến thành mũi màu đen 2 chiều thì kéo và thả chuột.

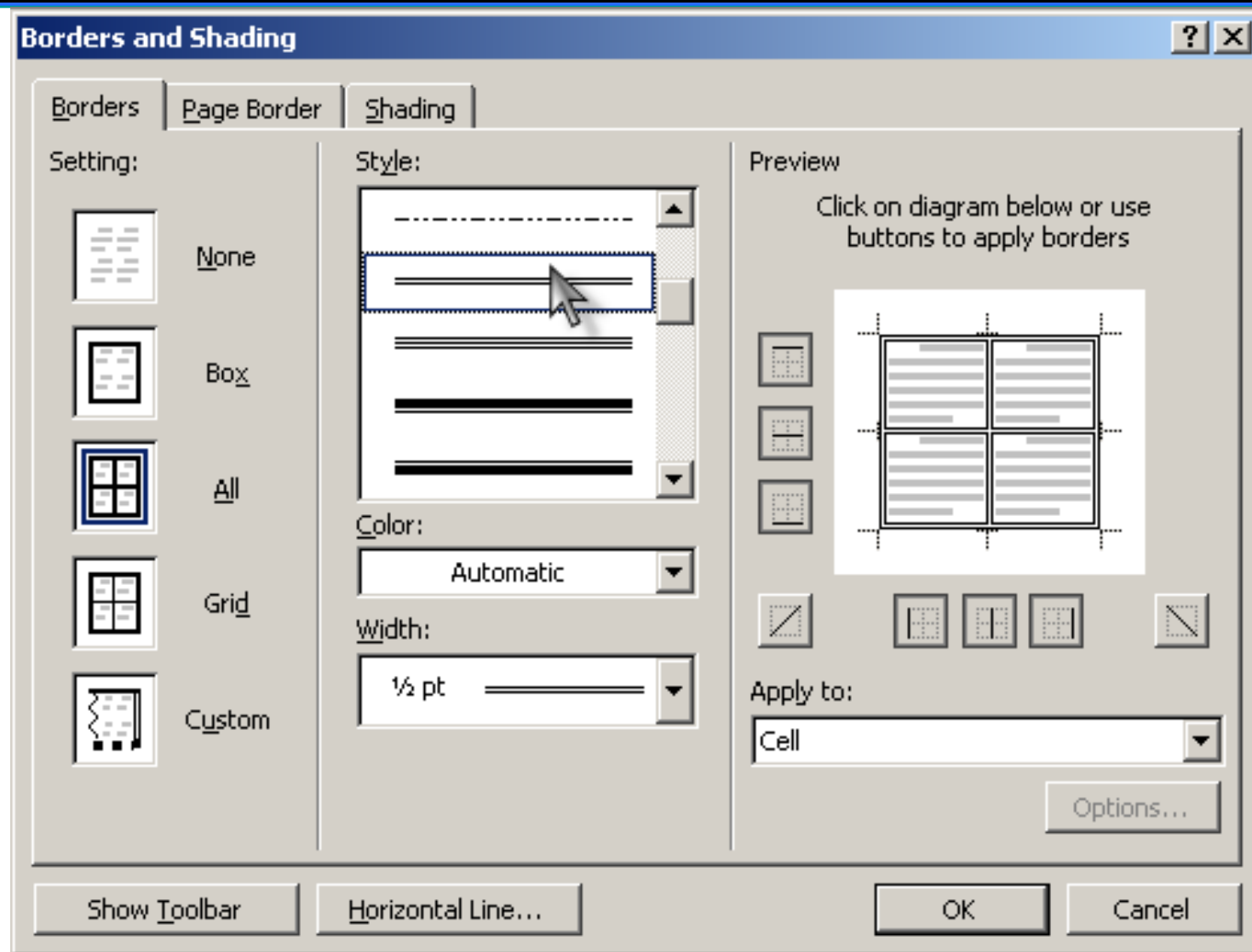
IV/ Thay đổi cấu trúc của bảng (tt)

10) Đóng khung, tô bóng mờ: Nếu muốn áp dụng thay đổi cho toàn bảng thì đặt con trỏ vào một vị trí bất kỳ trong bảng.

Đóng khung:

- Chọn khối ô muốn đóng khung.
- Vào menu **Format – Borders and Shading**, Chọn **Boder tab**. Xuất hiện hộp thoại:

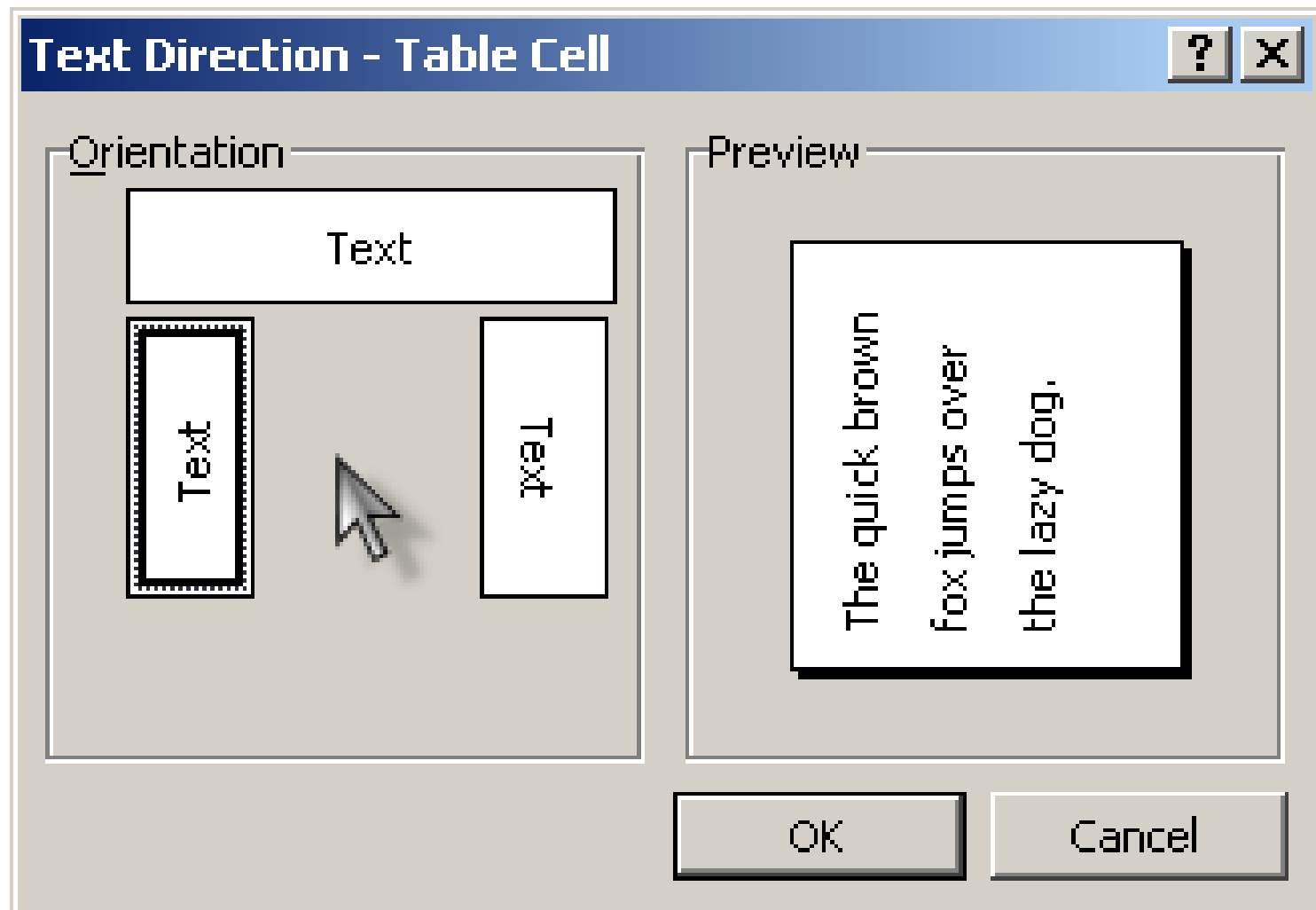
IV/ Thay đổi cấu trúc của bảng (tt)



□ Tô bóng mờ: Cũng trong cửa sổ **Borders and Shading**, chọn qua **tab Shading**.

V/ Thay đổi hướng văn bản trong ô

- ❑ Chọn khối ô cần thay đổi hướng.
- ❑ Vào **Format -> Text Direction**



VI/ Thêm hình ảnh vào tài liệu

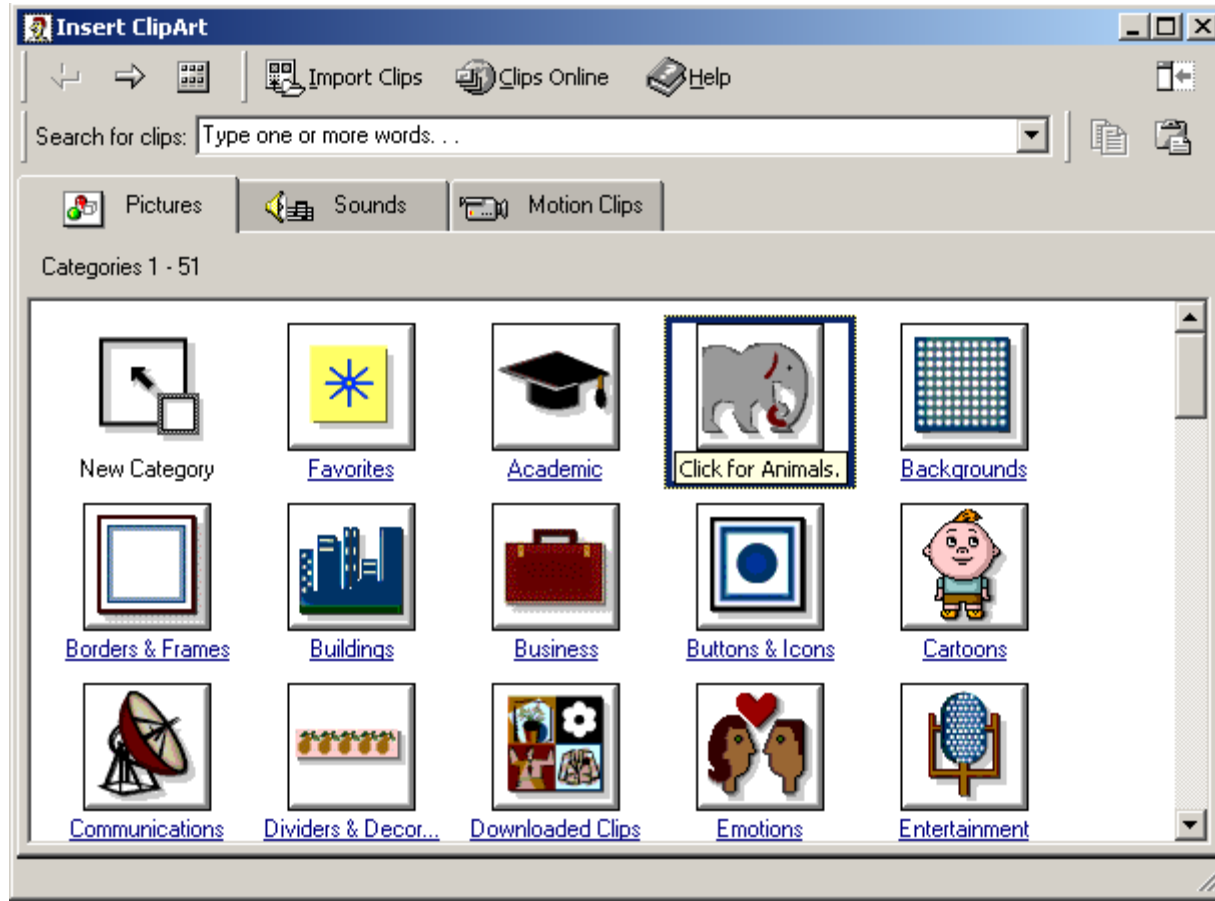
- ❑ Vào menu **Insert – Picture – Clip Art**
- ❑ Chọn một hình ảnh từ danh mục hình ảnh có sẵn trong bộ **Office**. Kích chuột phải và chọn **Insert** hay kích chuột trái và chọn vào biểu tượng **Insert Clip**.

📁 Nếu muốn chèn hình ảnh từ bất kỳ một tệp tin nào ảnh nào đó trong máy thì vào:

Insert – Picture – From File

📁 **Định dạng lại hình:**

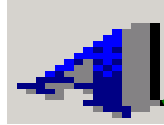
Nháy phải chuột vào hình cần định dạng, chọn **Format Picture**.



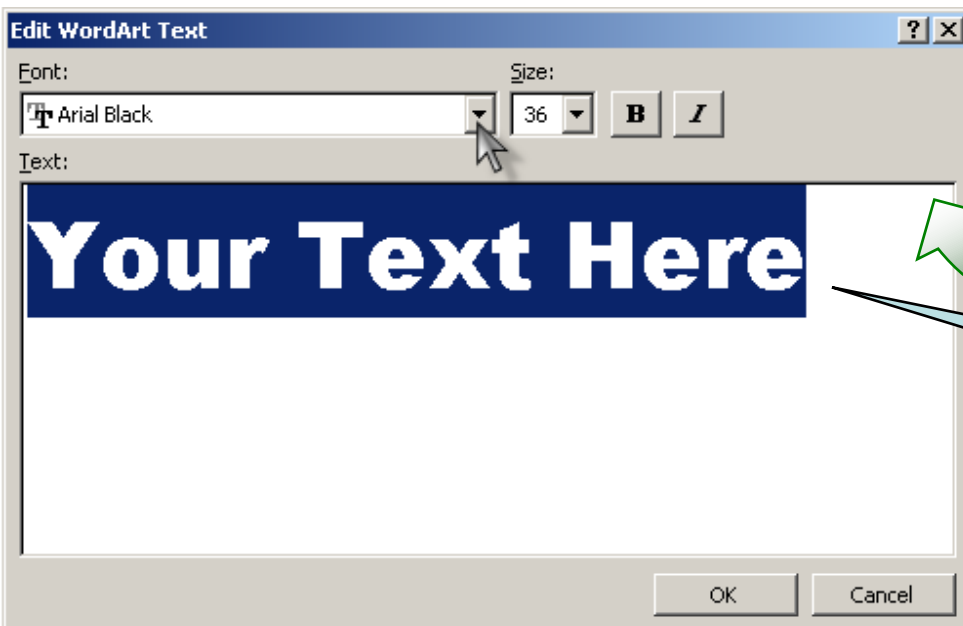
VII/ Tạo chữ nghệ thuật

o Insert - Picture - WordArt

ng Word Art trên thanh Drawing.

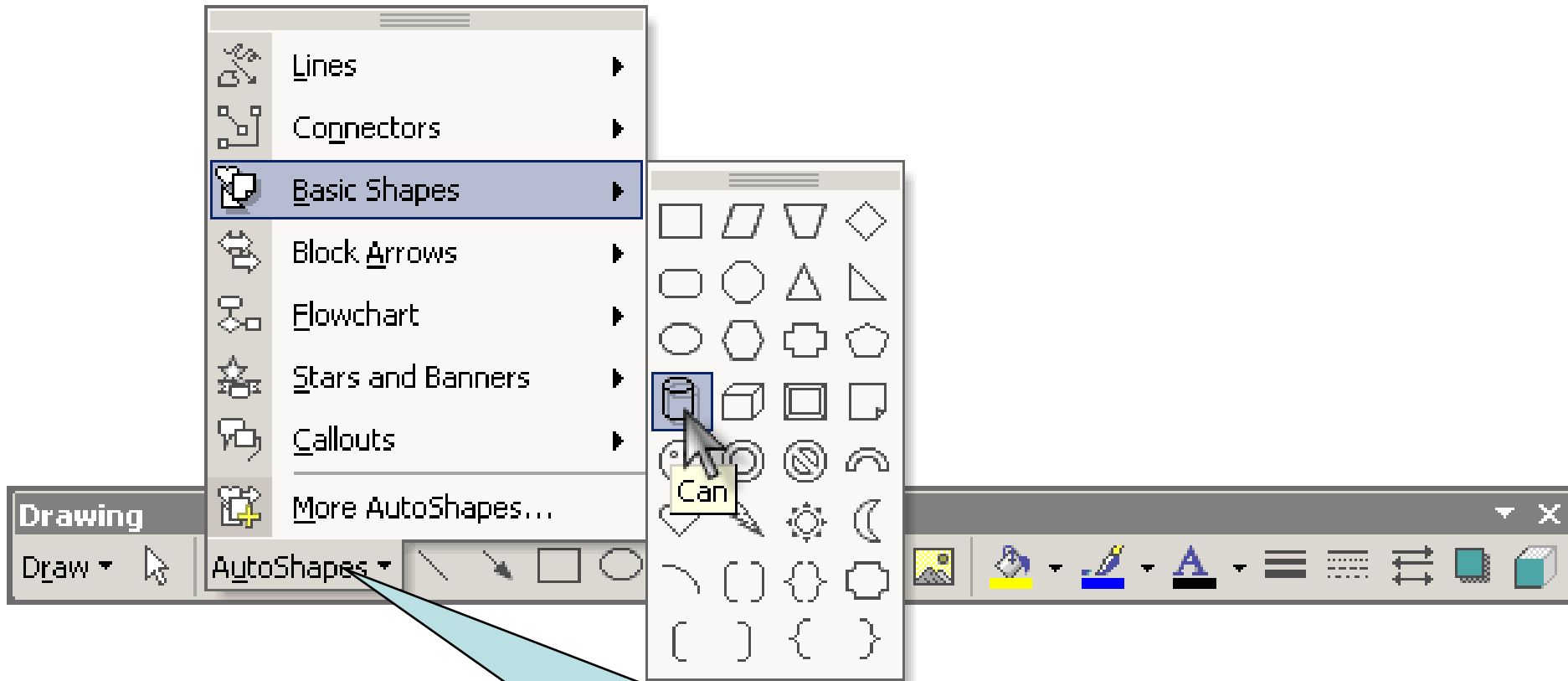


n OK



Chọn lại Font và i
d n.

VIII/ Thêm các khối hình học vào tài liệu



t khối

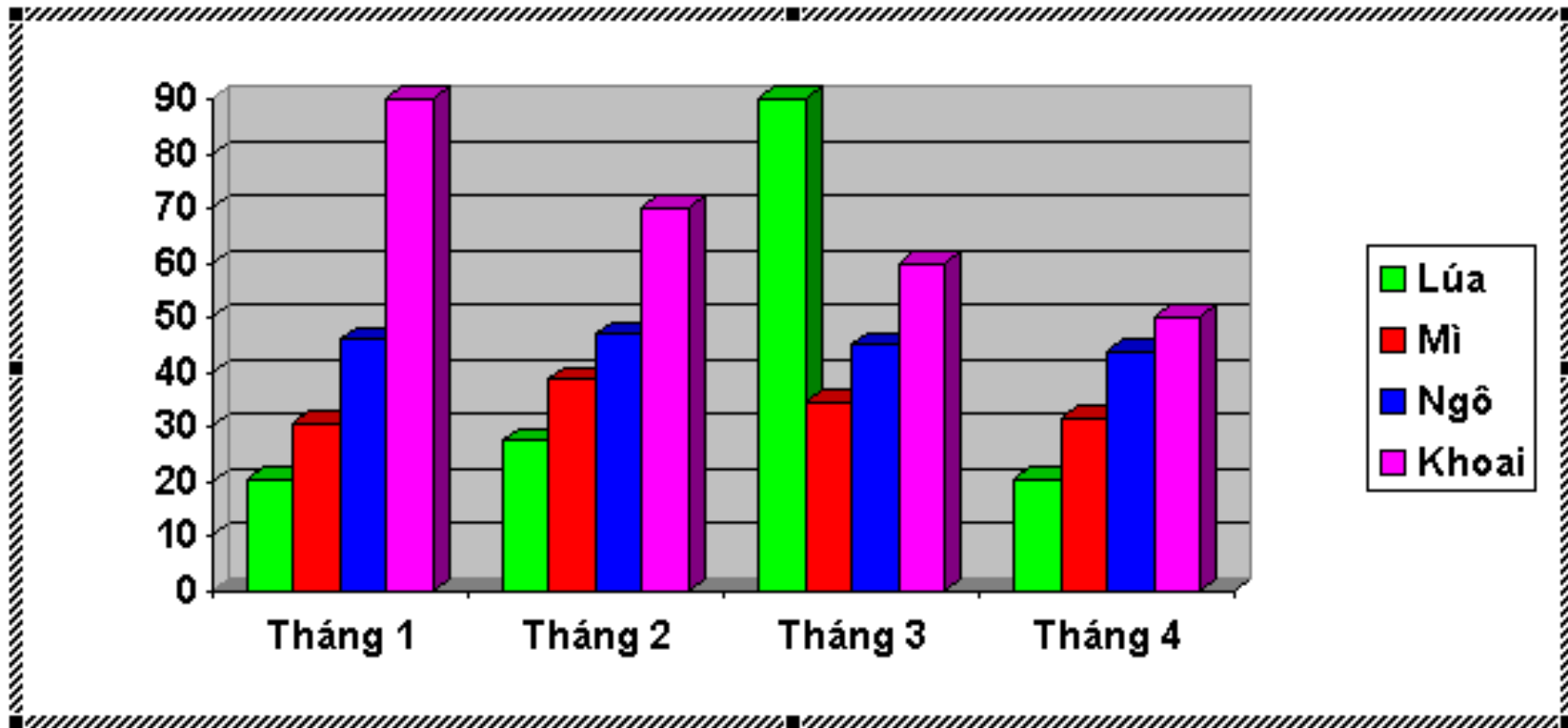
y,

t

o

IX/ Chèn biểu đồ vào tài liệu

- Vào menu **Insert - Picture - Chart**



Document 1 - Datasheet

		A	B	C	D	E
		Tháng 1	Tháng 2	Tháng 3	Tháng 4	
1	Lúa	20.4	27.4	90	20.4	
2	Mì	30.6	38.6	34.6	31.6	
3	Ngô	45.9	46.9	45	43.9	
4	Khoai	90	70	60	50	

BÀI 6

TIỆN ÍCH TRỘN THU

I/ Ý nghĩa của việc trộn thư


- ❑ Đây là công việc hay làm trong việc tạo ra các giấy mời, thư từ, phiếu báo điểm,... cùng một nội dung cho nhiều người, nhiều cơ quan,...
- ❑ Thao tác này cho phép chúng ta trộn 2 văn bản đã soạn thảo thành một văn bản “kết quả” theo một thứ tự cho trước.

II/ Chuẩn bị văn bản trộn

- ❑ Tạo văn bản thứ nhất có nội dung sẽ cố định trong văn bản “kết quả”, giả sử văn bản được lưu với tên: **Phieu_bao_diem.doc**
- ❑ Tạo văn bản thứ 2 có nội dung sẽ là các mục được trộn vào văn bản thứ nhất, giả sử văn bản được lưu với tên: **Danh_sach_diem.doc**. Với văn bản thứ 2 này, nội dung của chúng là một bảng biểu với các cột tương ứng là các trường cần trộn và các dòng là nội dung tương ứng cho từng trường.

II/ Chuẩn bị văn bản trộn (tt)

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC BÁCH KHOA
TP. HỒ CHÍ MINH

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc


PHIẾU BÁO ĐIỂM

Báo cho thí sinh:

Sinh ngày:

Địa chỉ:

Đã đạt kết quả trong kỳ thi tuyển sinh Đại học năm 2006 với số điểm như sau:

Môn Toán: Môn Lý: Môn Hóa:

Tổng điểm:

Kết quả:

Mọi thắc mắc hoặc đơn xin phúc khảo thí sinh gửi đến phòng Đào tạo của trường trước ngày 30/08/2006.

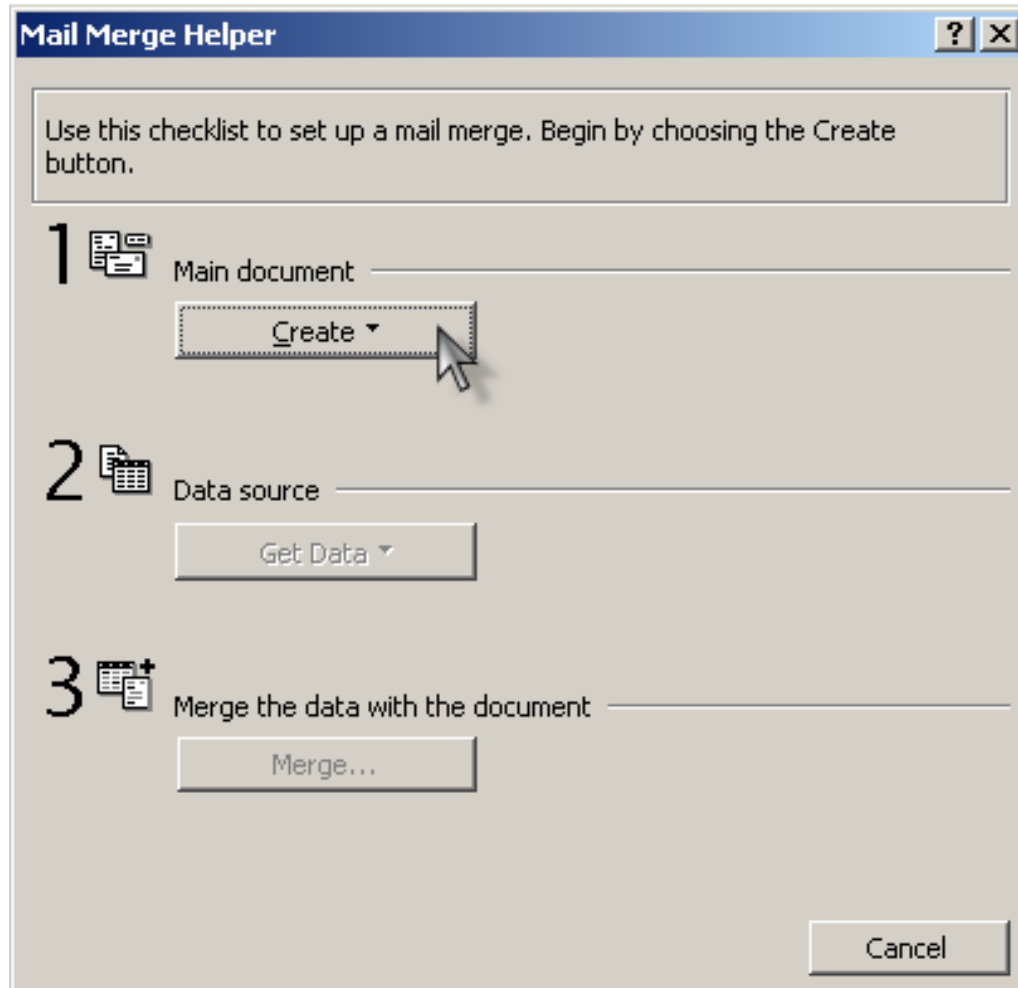
Tp.HCM, Ngày 10/08/2006

Hiệu trưởng

HOTEN	NGAYSINH	DIACHI	TOAN	LY	HOA	TONGDIEM
Đoàn Ngọc Tấn	26/05/1983	Quảng Ngãi	6	5	7	18
Nguyễn Thị Phương Thư	26/03/1983	Huế	6	2	1	9
Trần Văn Tâm	02/01/1982	Tp.HCM	3	6	8	17
Trần Tây Ninh	03/06/1981	Hà Nội	4	7	3	14

III/ Thao tác trộn

- ❑ Mở văn bản thứ nhất (Phieu_bao_diem.doc)
- ❑ Vào menu **Tools – Mail Merge**. Xuất hiện hộp thoại Mail Merger Helper với nội dung như sau:

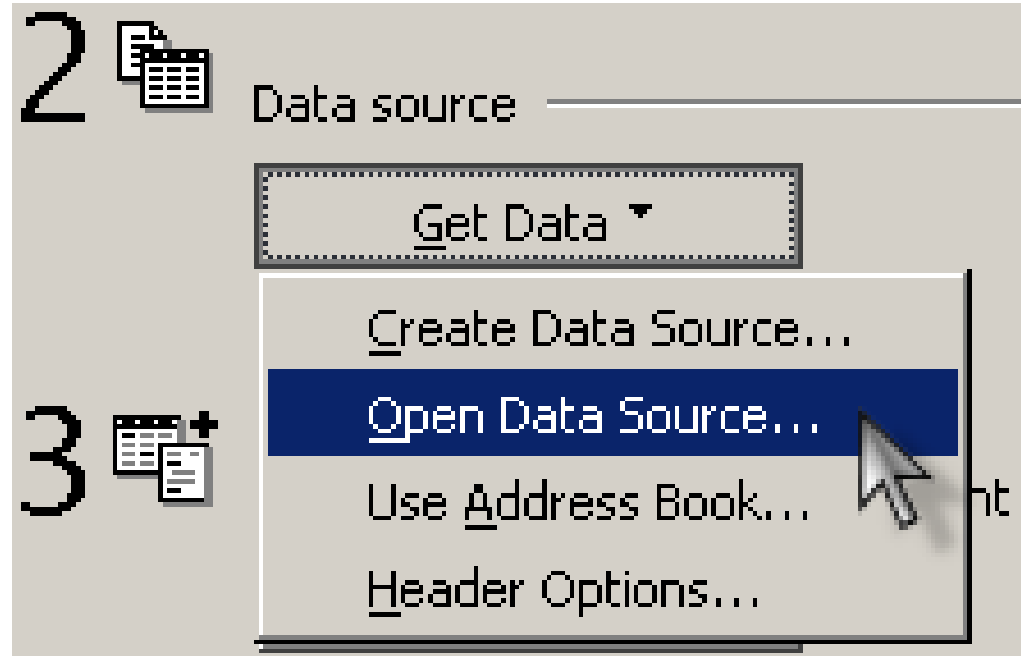
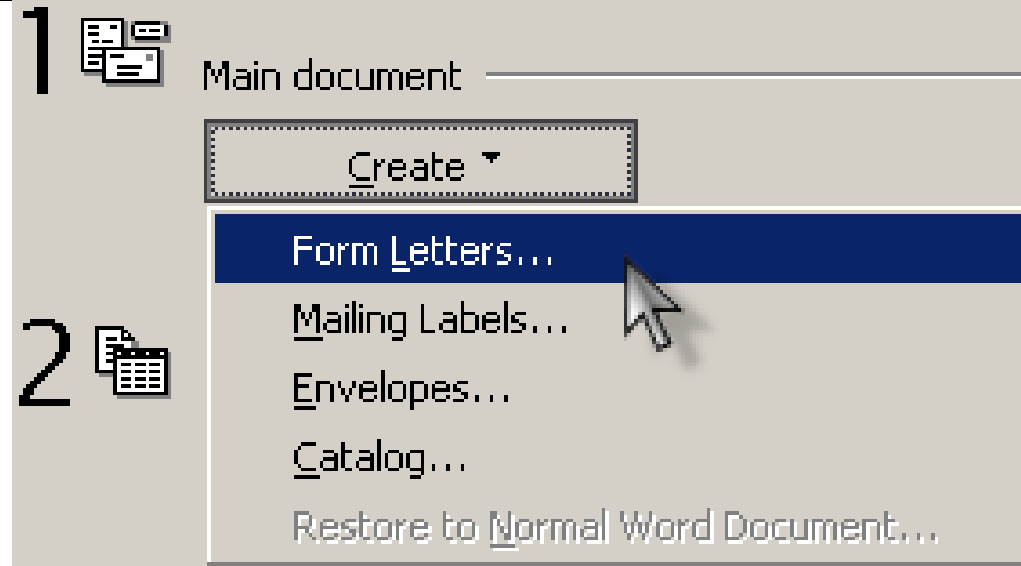


III/ Thao tác trộn (tt)

- Mục **Main document**: Cho phép chúng ta chọn tài liệu chính cần trộn (tài liệu thứ nhất). Với cách thực hiện này hệ thống sẽ ngầm hiểu tài liệu chính là tài liệu đang mở.
- Mục **Data source**: Cho phép chúng ta chọn tài liệu phụ cần trộn vào tài liệu chính (tài liệu thứ 2).
- Mục **Merge the data with the document**: Chấp nhận trộn nội dung của 2 tài liệu đã chọn.

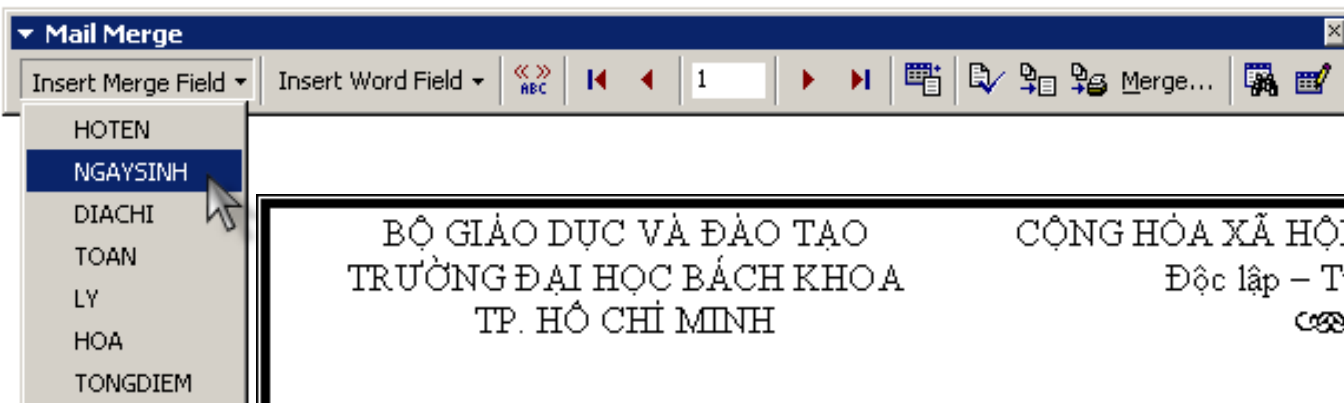
III/ Thao tác trộn (tt)


- Từ nút lệnh **Create** -> **Form letter** -> **Chọn Active Windows** khi hộp thoại Microsoft Windows xuất hiện.
- Từ nút lệnh **Get data** -> **Open data source** -> Chọn đường dẫn và xác định cho hệ thống biết địa chỉ của tệp tin thứ 2 cần trộn -> **Chọn Edit main document** khi hộp thoại Microsoft Windows xuất hiện.



III/ Thao tác trộn (tt)

Lúc này hệ thống sẽ tự động đưa ra thành công cụ **Mail Merge**:



BỘ GIÁO DỤC VÀ ĐÀO TẠO TRƯỜNG ĐẠI HỌC BÁCH KHOA TP. HỒ CHÍ MINH	CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM Độc lập – Tự do – Hạnh phúc 
---	--

PHIẾU BÁO ĐIỂM

Báo cho thí sinh: «HOTEN».....

Sinh ngày:

Địa chỉ:

Đã đạt kết quả trong kỳ thi tuyển sinh Đại học năm 2006 với số điểm như sau:

Môn Toán: Môn Lý: Môn Hóa:

Tổng điểm:

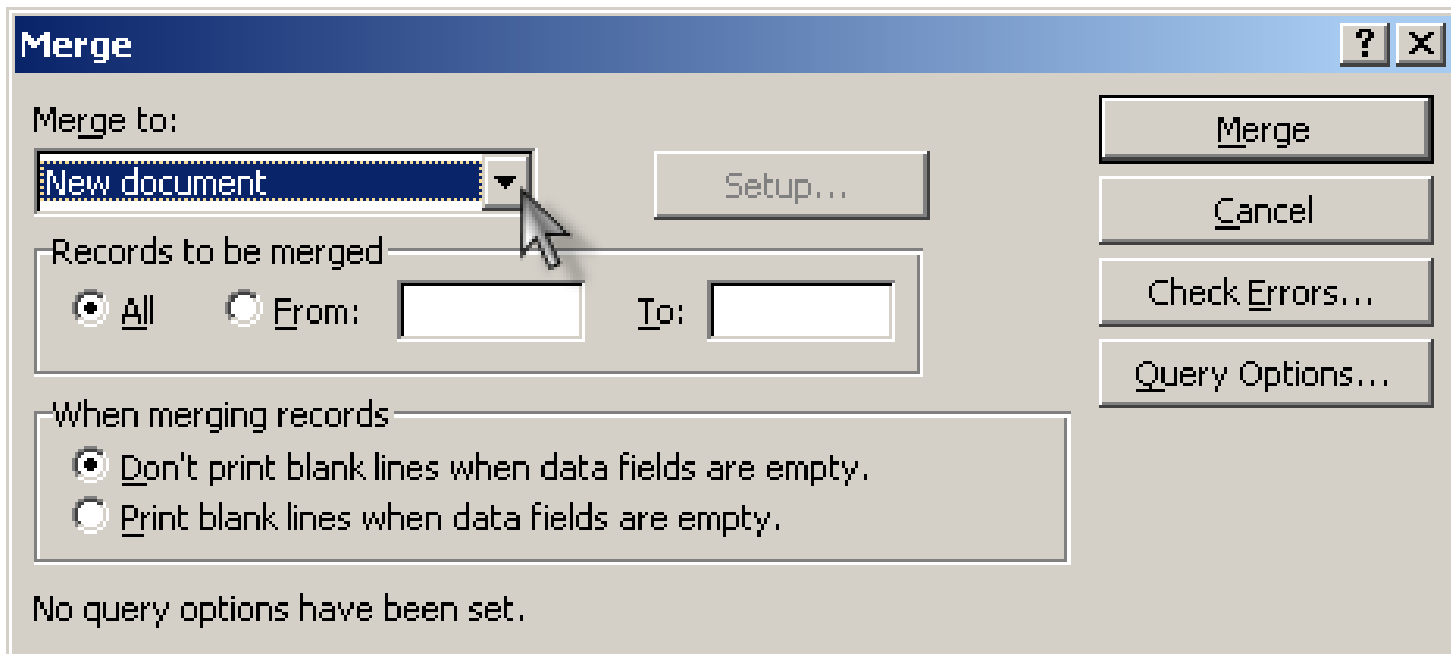
Kết quả:

Mọi thắc mắc hoặc đơn xin phúc khảo thí sinh gửi đến phòng Đào tạo của trường trước ngày 30/08/2006.

Tp.HCM, Ngày 10/08/2006
Hiệu trưởng

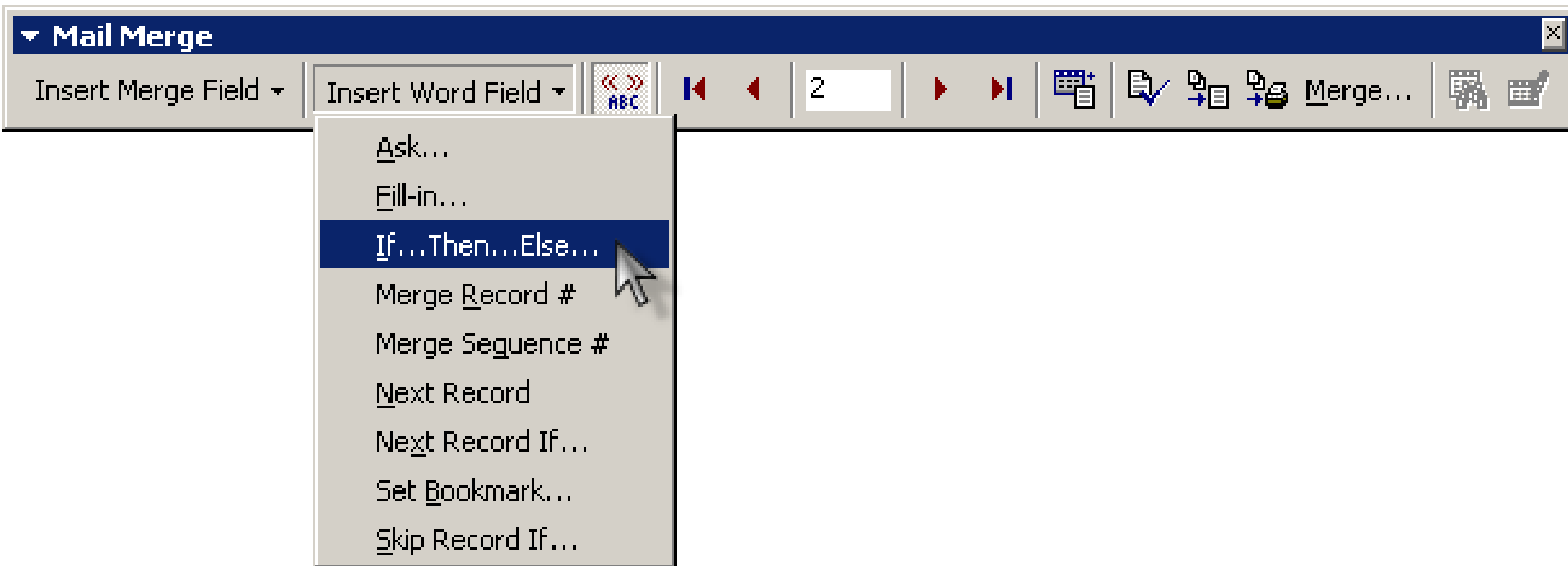
III/ Thao tác trộn (tt)

- Menu **Insert Merge Field**: Danh sách các trường cần trộn tương ứng trong tệp tin thứ 2.
- Lần lượt chèn tên các trường vào vị trí cần trộn.
- Nút lệnh **Merge...** trên thanh công cụ cho phép bắt đầu công việc trộn với các chức năng sau:
 - ✓ **Merge to**: Chọn nơi sẽ xuất hiện văn bản kết quả.
 - ✓ **Record to be merged**: Chọn số lượng bản ghi sẽ trộn.



IV/ Nâng cao chức năng trộn thư

- Trong phiếu báo điểm trong danh sách trên còn có một dòng “Kết quả:...”. Giả sử kết quả được tính với yêu cầu như sau: Nếu Tổng điểm ≥ 15 thì đậu, ngược lại thì Rớt.
- Với yêu cầu như vậy thì ta sử dụng thêm chức năng sau: Vào menu **Insert Word Field -> If...Then...Else...**



- Xuất hiện hộp thoại sau:

IV/ Nâng cao chức năng trộn thư

Insert Word Field: IF

IF

Field name: TONGDIEM

Comparison: Greater than or equal

Compare to: 15

Insert this text: Đậu

Otherwise insert this text: Rớt

OK Cancel

- **If** <Điều kiện> **Then** <Câu lệnh 1> **Else** <Câu lệnh 2>:
Nếu Điều kiện đúng thì thực hiện Câu lệnh 1, ngược lại Điều kiện sai thì thực hiện Câu lệnh 2.

IV/ Nâng cao chức năng trộn thư (tt)

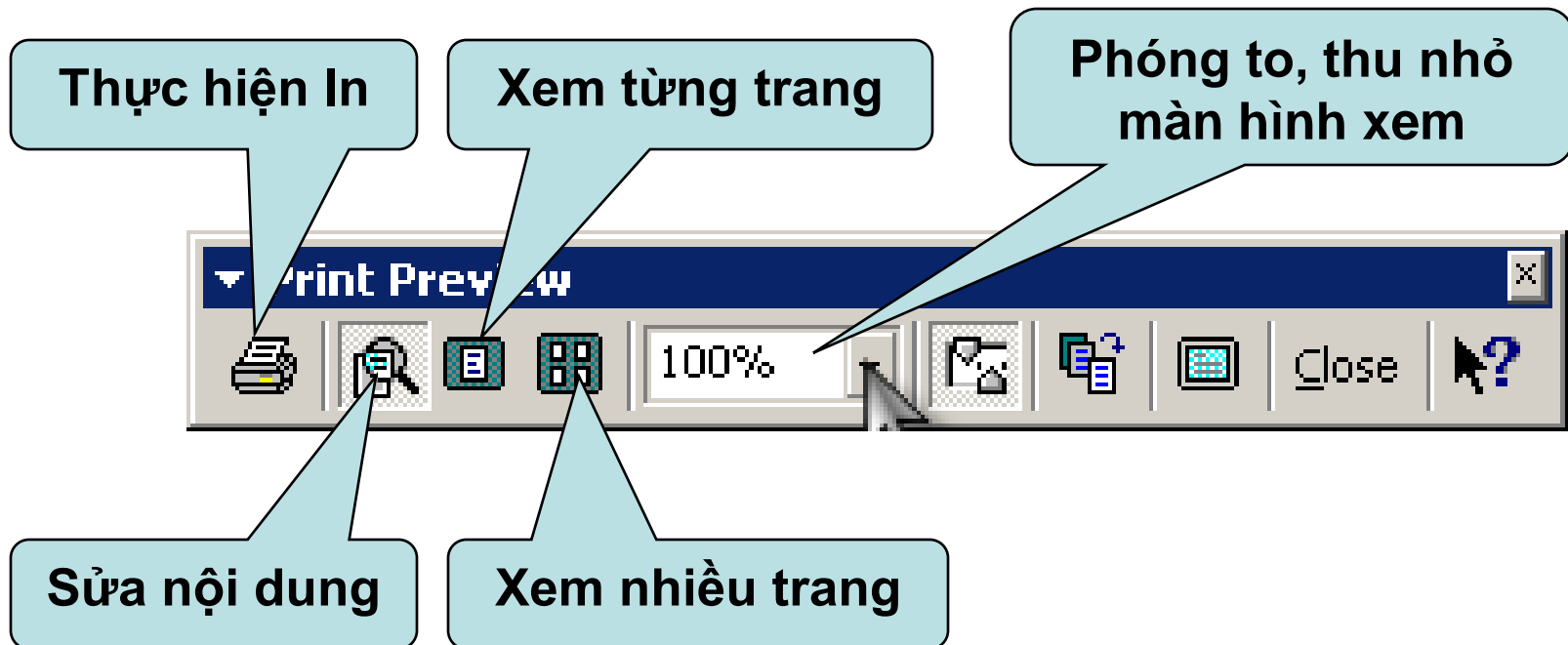
- ❑ **Field name:** Trường cần so sánh.
- ❑ **Comparison:** Biểu thức so sánh:
 - **Equal to:** Bằng
 - **Not equal to:** Không bằng
 - **Less than:** Nhỏ hơn
 - **Greater than:** Lớn hơn
 - **Less than or equal:** Nhỏ hơn hoặc bằng
 - **Greater than or equal:** Lớn hơn hoặc bằng
 - **Is blank:** Là rỗng
 - **Is not blank:** Không là rỗng
- ❑ **Compare to:** Giá trị so sánh.
- ❑ **Insert this text:** Giá trị khi biểu thức so sánh đúng.
- ❑ **Otherwise insert this text:** Giá trị khi biểu thức so sánh sai.

BÀI 7

IN ẤN & THIẾT ĐẶT MẶT KHẨU

I/ Xem trước khi in

- ❑ Thao tác này cho phép xem nhanh toàn bộ hình dáng thật của văn bản khi kết xuất văn bản ra máy in (in ra giấy).
- ❑ Vào menu **File -> Print preview** hoặc nhấn biểu tượng **Preview** trên thanh công cụ.
- ❑ Thanh công cụ liên quan: Thanh **Print preview**:

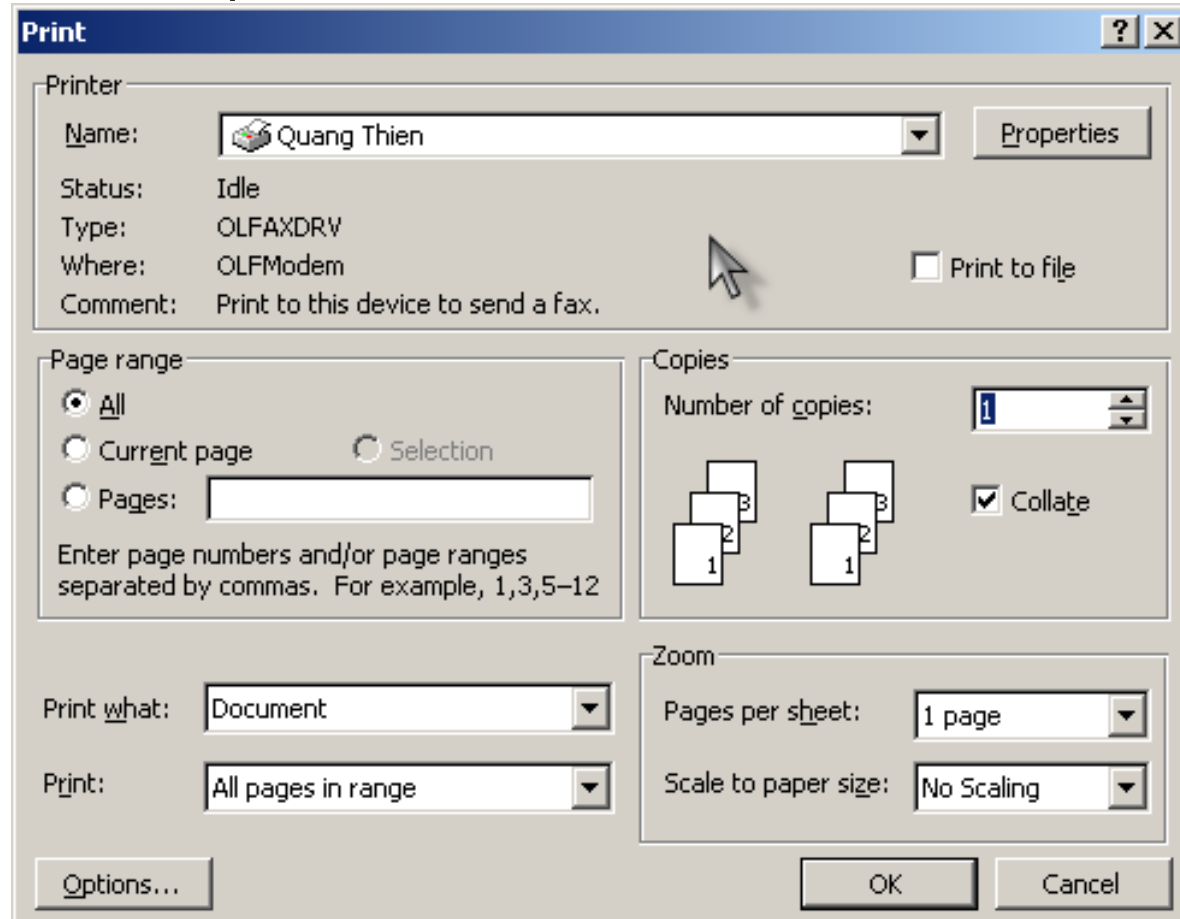


II/ Thực hiện in

❑ Vào menu **File -> Print** (Hoặc nhấn tổ hợp phím **Ctrl+P**).

➤ **Page range:** Chọn số trang in.

➤ **Copies:** Chọn số lần in cho một trang.



➤ **Collate:** Chọn thứ tự in trong trường hợp in nhiều lần cho một trang.

III/ Thiết đặt mật khẩu

- Vào **Tools - Option...** Chọn thẻ **Save** (hoặc vào menu **File -> Save as...Tools -> General Options...**)
- Gõ mật khẩu bảo vệ khi mở vào hộp văn bản **Password to open** và mật khẩu bảo vệ cấu trúc (nếu muốn) vào hộp **Password to modify**.

File sharing options for "Document2"

Password to open:

Password to modify:

Read-only recommended

OK

Cancel

Module 5

BẢNG TÍNH ĐIỆN TỬ MICROSOFT EXCEL

BÀI 1

KHỞI ĐỘNG VÀ GIỚI THIỆU CÁC THÀNH PHẦN GIAO DIỆN

I/ Khởi động Excel

- ❑ Sử dụng các cách khởi động Excel giống như các cách đã khởi động Word.
- ❑ Tập tin Excel sinh ra mặc định có phần mở rộng là **XLS**

II/ Cấu tạo bảng tính MS-Excel

The image shows a screenshot of the Microsoft Excel application window titled "Microsoft Excel - Book1". The interface includes a menu bar (File, Edit, View, Insert, Format, Tools, Data, Window, Help), a toolbar with various icons, and a formatting toolbar. The main workspace is a grid with columns labeled A through J and rows numbered 1 through 17. The formula bar at the top shows "A1" and an equals sign (=). A callout box points to the formula bar with the text "Thanh công thức". Another callout box, shaped like a starburst, points to the grid with the text "Vùng bảng tính". A third callout box points to the sheet tabs at the bottom, which are labeled "Sheet1", "Sheet2", and "Sheet3", with the text "Sheet1, Sheet2, Sheet3 là 3 Worksheet mặc định của một Workbook". The status bar at the bottom left shows "Ready" and the bottom right shows "NUM".

Thanh công thức

Vùng bảng tính

Sheet1, Sheet2, Sheet3 là 3 Worksheet mặc định của một Workbook

II/ Cấu tạo bảng tính MS-Excel (tt)

- ❑ Sau khi khởi động Excel, chúng ta có một cửa sổ bảng tính (**Workbook**).
- ❑ Trong một bảng tính có nhiều trang bảng tính (**Worksheet**).
- ❑ Một **Workbook** có tối đa **255 Worksheet**.
- ❑ Trên một **Worksheet** được chia thành nhiều cột và nhiều dòng. Bao gồm **256 cột** và **65536 dòng**.
- ❑ Cột (Columns): Được kí hiệu bằng các chữ cái **A, B, ..., Z, AA, ..., IV**
- ❑ Dòng (Rows): Được đánh số thứ tự bằng các chữ số **1, 2, 3, ..., 65536**

II/ Cấu tạo bảng tính MS-Excel (tt)

- ❑ Ô (Cell): Giao của một dòng và một cột gọi là ô.
- ❑ Ô được xác định bởi một tọa độ (gọi là địa chỉ ô) dựa theo kí hiệu của cột và số thứ tự của dòng.

	A	B	C	D	E
1					
2					
3					
4					
5					
6					
7					

Địa chỉ của ô là **B2** →

Địa chỉ của ô là **C5** →

BÀI 2

CÁC THAO TÁC CƠ BẢN TRÊN BẢNG TÍNH EXCEL

I/ Nhập dữ liệu

1/ Nhập dữ liệu kiểu số, kiểu ngày tháng:

- Dữ liệu kiểu số được nhập vào mặc định nằm bên phải của ô.
- Muốn nhập số có **phần thập phân**, đánh dấu “.” (bên phải bàn phím) vào giữa phần nguyên và phần thập phân.
- Kiểu ngày tháng nên nhập theo định dạng **ngày/tháng/năm** của Việt Nam (**dd/mm/yyyy**).

2/ Nhập dữ liệu kiểu văn bản:

- Dữ liệu kiểu văn bản được nhập vào mặc định nằm bên trái của ô.
- Để nhập kiểu số dạng văn bản thì nhập số đằng sau dấu nháy đơn ‘

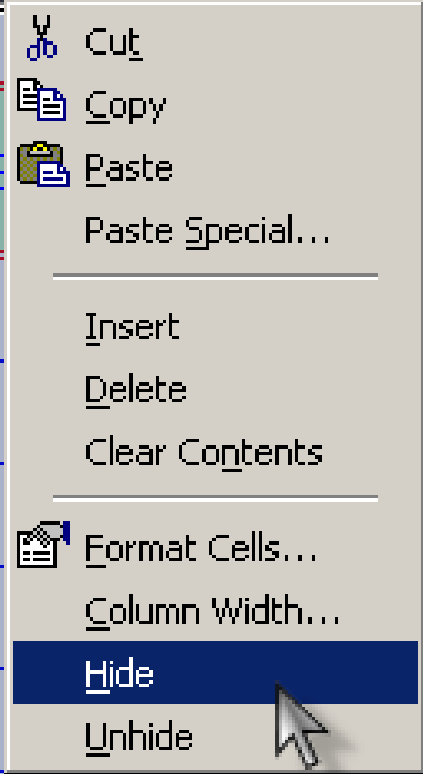
II/ Hiệu chỉnh bảng tính

- 1/ Chọn, sao chép, di chuyển vùng bảng tính:** Thực hiện các thao tác giống như trên Windows.
- 2/ Chèn thêm dòng:** Dòng trong Excel được chèn sẽ nằm bên trên dòng hiện tại. Vào menu **Insert -> Rows.**
- 3/ Chèn thêm cột:** Cột trong Excel được chèn sẽ nằm bên trái cột hiện tại. Vào menu **Insert -> Columns.**
- 4/ Xóa dòng, cột:** Chọn những dòng, cột cần xóa. Nháy phải chuột vào những dòng, cột đã chọn **-> Delete.**
- 5/ Thay đổi độ rộng của cột và chiều cao của dòng:** Di chuyển chuột vào vị trí giao giữa 2 tên cột, hoặc dòng, khi đó con trỏ chuột chuyển thành mũi tên 2 chiều, kéo và thả. (Hoặc **Format -> Column/Row -> Width/Height**).

III/ Hiển thị hay che dấu cột và dòng

- Để ẩn các cột hay dòng ta chọn các cột hay dòng đó, kích chuột phải -> **Hide**
- Để hiển thị lại các cột hay dòng, ta chọn các cột hay dòng có chứa các cột hay dòng bị ẩn đó, kích chuột phải -> **Unhide**

C	D	E	F	G
QUÊ QUÁN	DLT			
Hà Nội	9			
Quảng Nam	6			
Quảng Ngãi	7			
Quảng Ngãi	9			
Hà Nội	6			
Bình Định	6.5	6	6.33	5/11
Quảng Ngãi	5	7	5.67	7/11



A context menu is open over column D, showing options: Cut, Copy, Paste, Paste Special..., Insert, Delete, Clear Contents, Format Cells..., Column Width..., Hide (highlighted), and Unhide. A mouse cursor is pointing at the 'Unhide' option.

IV/ Cố định dòng tiêu đề, cột tiêu đề

- ❑ Khi nhập bảng tính có nhiều cột, nhiều dòng, chúng ta có nhu cầu muốn cố định dòng/cột tiêu đề trong khi các dòng/cột dưới được phép cuộn lên hoặc cuộn xuống.
- ❑ **Các bước thực hiện:**
 - Đưa con trỏ tới bất kỳ ô nào nằm ở dưới dòng tiêu đề.
 - Vào menu **Window**, chọn **Freeze Panes**.
- ❑ Muốn gỡ bỏ việc cố định tiêu đề: Vào menu **Window**, chọn **Unfreeze Panes**.

V/ Công cụ điền nội dung tự động

1/ Điền tự động theo cấp số cộng (Sử dụng phím **Ctrl**):

- Nhập số 1 vào ô A2.
- Nhấn giữ phím **Ctrl**.
- Đưa chuột vào **góc dưới bên phải** của ô, khi con trỏ chuột biến thành 2 dấu cộng (dấu lớn và dấu nhỏ).
- Kéo và thả sẽ cho kết quả.

	A	B
1	SỐ TT	
2	1	
3		
4		
5		
6		
7		
8		
9		
10		

Trước khi kéo

	A	B
1	SỐ TT	
2	1	
3	2	
4	3	
5	4	
6	5	
7	6	
8	7	
9	8	
10	9	

Sau khi kéo

V/ Công cụ điền nội dung tự động (tt)

2/ Điền tự động theo cặp số cộng:

- Nhập 2 số vào 2 ô trên dưới liên tiếp nhau. **Công sai** của cặp số cộng là **hiệu** của số dưới với số trên.
- Chọn cả 2 ô vừa nhập số. Đưa chuột vào góc dưới bên phải của ô dưới, khi con trỏ chuột biến thành dấu cộng thì **kéo và thả**.

V/ Công cụ điền nội dung tự động (tt)

3/ Điền tự động theo cấp số nhân:

- Nhập 2 số vào 2 ô trên dưới liên tiếp nhau. **Công bội** của cấp số cộng là **thương** của số dưới với số trên.
- Chọn cả 2 ô vừa nhập số. Đưa chuột vào góc dưới bên phải của ô dưới, khi con trỏ chuột biến thành dấu cộng thì kích phải chuột **kéo và thả**. Một menu ngữ cảnh xuất hiện, chọn **Growth Trend**.

VI/ Các thao tác với trang bảng tính

1/ Chèn thêm một Worksheet vào Workbook:

- Worksheet mới được chèn vào sẽ nằm bên trái Worksheet hiện tại.
- Vào menu **Insert -> Worksheet**
- Hoặc kích chuột phải lên một Worksheet bất kỳ, chọn **Insert... -> Worksheet**

2/ Đổi tên một Worksheet: Kích chuột phải lên Worksheet cần đổi tên và chọn **Rename** (Hoặc kích đúp chuột lên Worksheet).

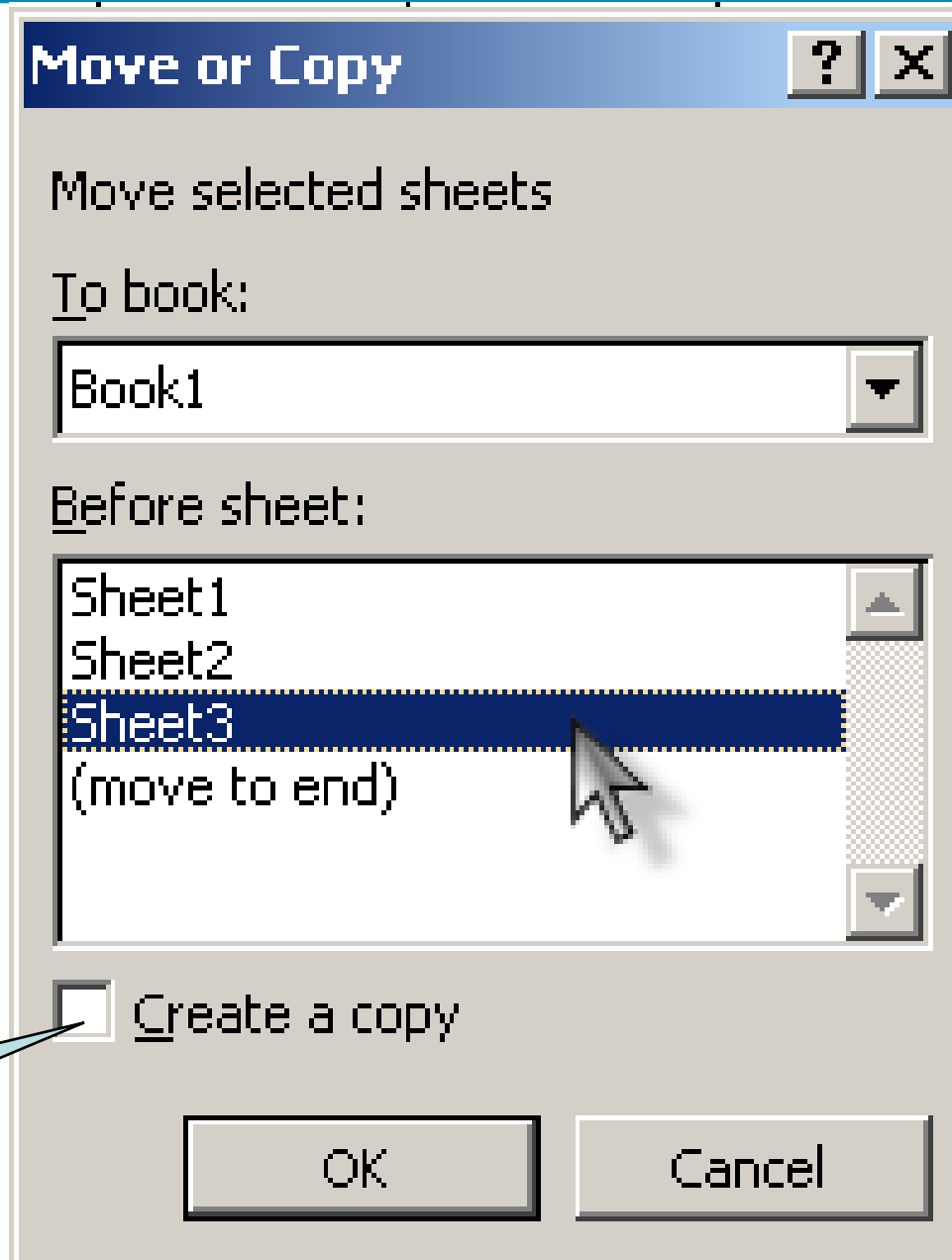
3/ Xóa một Worksheet: Kích phải chuột lên Worksheet cần xóa và chọn **Delete**.

VI/ Các thao tác với trang bảng tính (tt)

4/ Sao chép hay di chuyển Worksheet:

- **Cách 1:** Dùng chuột kéo và thả (nếu giữ thêm phím **Ctrl** trong khi kéo và thả thì thực hiện thao tác sao chép)
- **Cách 2:** Kích chuột phải lên Worksheet muốn di chuyển và chọn **Move or Copy...**

Dùng để sao chép

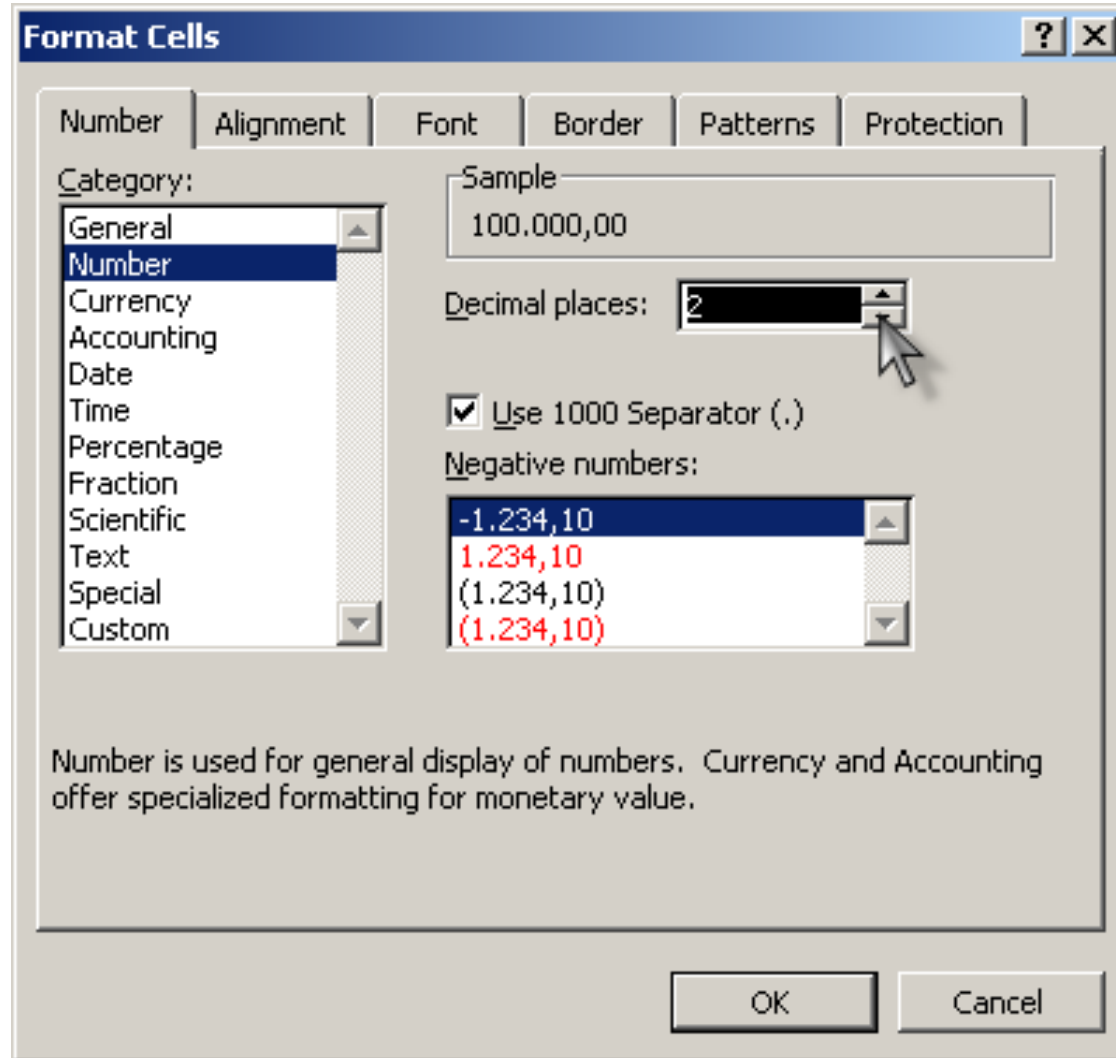


BÀI 3

ĐỊNH DẠNG BẢNG TÍNH

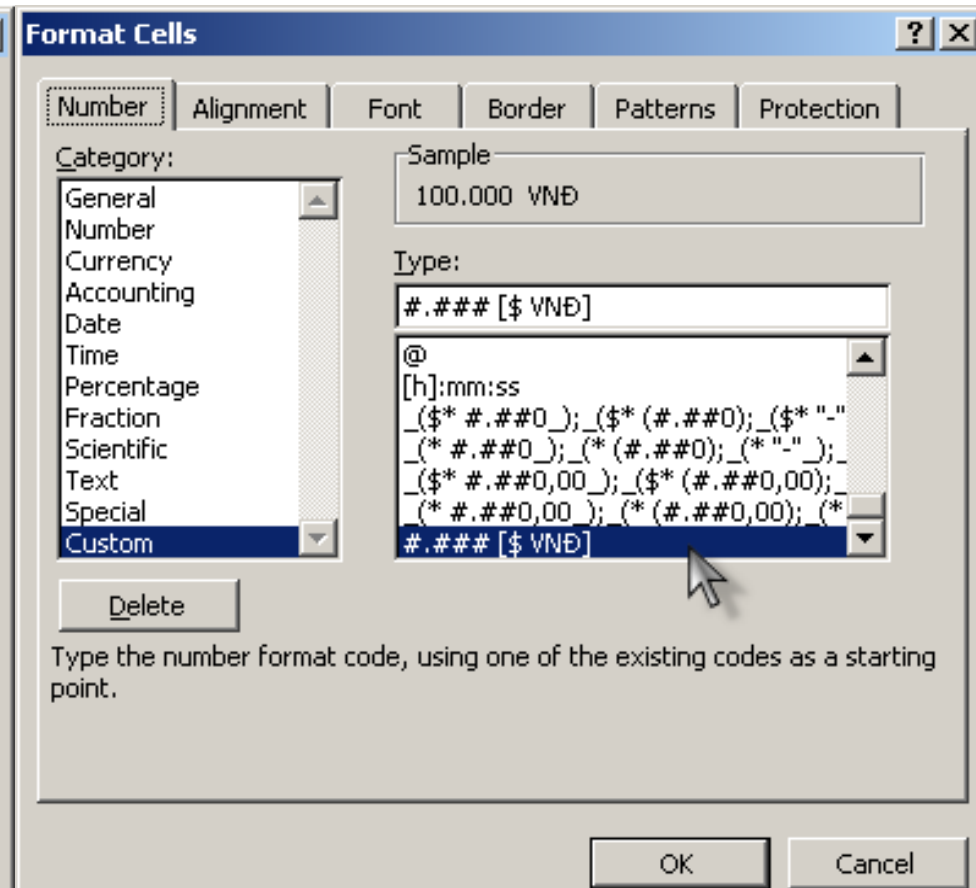
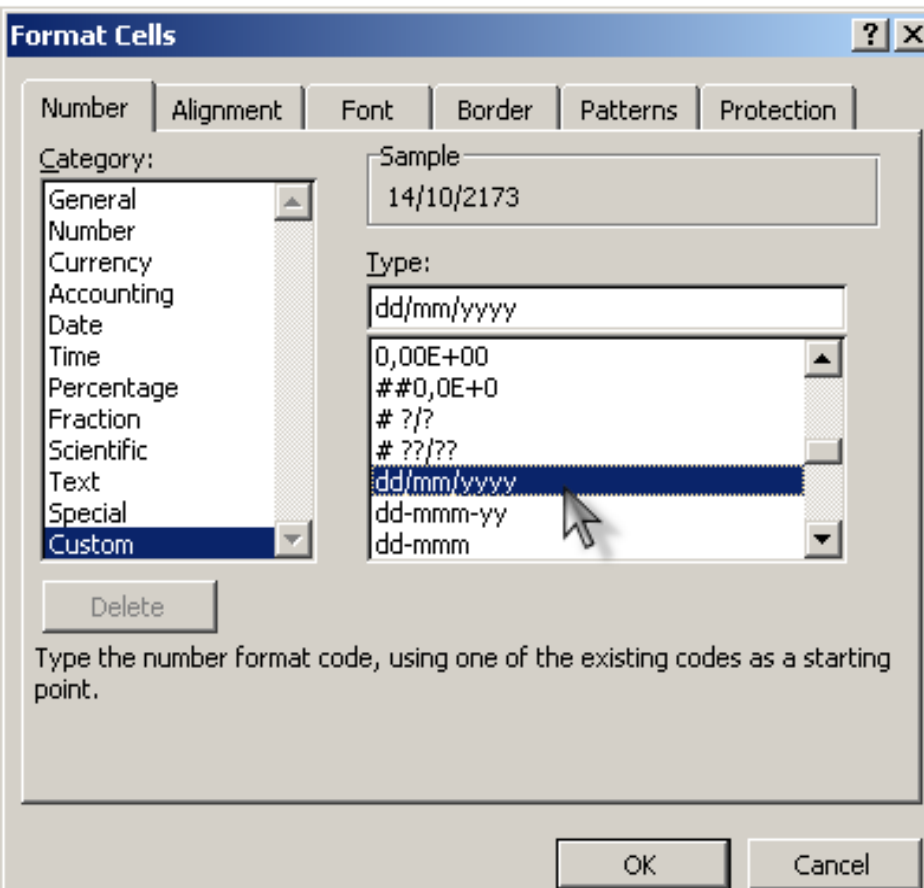
I/ Định dạng dữ liệu kiểu số

- ❑ Cần chỉnh các thông số thích hợp trong **Regional Options**.
- ❑ Chọn vùng bảng tính cần định dạng.
- ❑ Vào menu **Format -> Cells...**
- ❑ Đứng ở Tab Number.
- ❑ Chọn mục Number.
- ❑ **Decimal places:** Số chữ số thập phân.
- ❑ **Use 1000 Separator (,):** Sử dụng dấu ngăn cách hàng ngàn.
- ❑ **Negative numbers:** Các cách thể hiện cho số âm.
- ❑ **Định dạng kiểu ngày tháng:** Chọn **dd/mm/yyyy** trong **Custom**



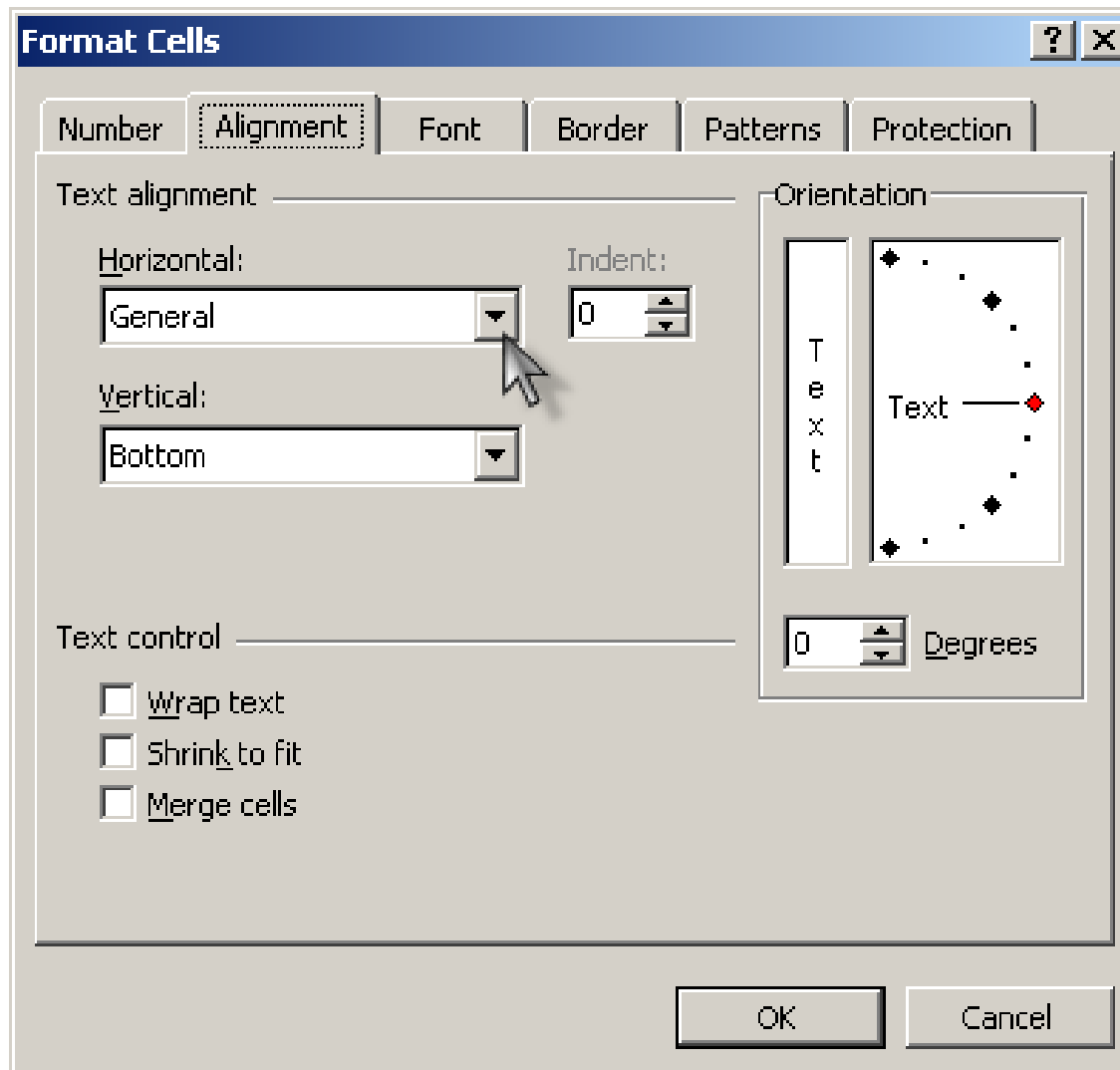
II/ Định dạng dữ liệu kiểu ngày tháng và tiền tệ

- ❑ **Kiểu ngày tháng:** Chọn định dạng **dd/mm/yyyy** trong mục **Custom** của Tab **Number**.
- ❑ **Kiểu tiền tệ:** Chọn định dạng **#,### [\$ VNĐ]** trong mục **Custom** của Tab **Number**.



III/ Định dạng ô chứa văn bản

- ❑ Vào menu **Format -> Cells...**
- ❑ Đứng ở Tab **Alignment**.



III/ Định dạng ô chứa văn bản (tt)

- ❑ **Horizotal:** Các định dạng lề canh cho dữ liệu trong ô theo chiều nằm ngang.
- ❑ **Vertical:** Các định dạng lề canh cho dữ liệu trong ô theo chiều thẳng đứng.
- ❑ **Orientation:** Xác định hướng nghiêng cho dữ liệu trong ô.
- ❑ **Wrap text:** Tự động ngắt dòng trong trường hợp độ rộng của ô nhỏ hơn chiều dài của chuỗi.
- ❑ **Shrink to fit:** Tự động điều chỉnh cỡ chữ (Không thay đổi **Size** chữ) trong trường hợp độ rộng của ô nhỏ hơn chiều dài của chuỗi.
- ❑ **Merge cells:** Trộn nhiều ô lại thành một ô. (Hoặc có thể kích chuột lên biểu tượng gộp ô  trên thanh định dạng).

BÀI 4

CÔNG THỨC VÀ HÀM

I/ Tạo công thức cơ bản

- Công thức được tạo ra nhằm mục đích để tính toán và trả lại giá trị cho bảng tính.
- Phải nhập dấu **=** trước mọi công thức.

The screenshot shows the Microsoft Excel interface. The formula bar at the top displays the formula `=A1+B1`. The spreadsheet grid shows the following data:

	A	B	C	D	E	F	G	H	I	J
1	2	5			=A1+B1					
2	6	6			12					
3	9	9			18					
4	7	3			10					
5	6	4			10					
6	3	7			10					
7										
8										
9										
10										
11										
12										

A callout box with the text "Thành công thức" points to the formula bar.

II/ Các toán tử trong công thức

- Toán tử liên kết chuỗi: **&**
 - Nối chuỗi với chuỗi: **“Tin”&” Học”** -> **Tin Học**
 - Nối số với số: **123&456** -> **123456**
 - Nối số với chuỗi: **“Excel “&2000** -> **Excel 2000**
 - Nối chuỗi với một biểu thức số học:
“Phim “&2002/2&”đêm” -> **Phim 1001 đêm**
 - Nối chuỗi với hàm: **“Tổng số là “&Sum(10;20)** -> **Tổng số là 30**
- Toán tử so sánh: **>, <, >=, <=, =, <>**
 - Các biểu thức so sánh luôn cho kết quả là một giá trị **logic** (True hay False)
 - Ví dụ: **240<350** -> **True**
250=360-100 -> **False**

III/ Các loại địa chỉ

- ❑ **Địa chỉ tương đối:** Loại địa chỉ này khi sao chép công thức sẽ bị thay đổi địa chỉ dòng hoặc địa chỉ cột.
- ❑ **Địa chỉ tuyệt đối:** Sẽ không bị thay đổi địa chỉ dòng và địa chỉ cột khi sao chép công thức. Kí hiệu địa chỉ tuyệt đối: **\$Cột\$Dòng**.
- ❑ **Địa chỉ hỗn hợp:**
 - **Tuyệt đối dòng:** Sẽ không bị thay đổi địa chỉ dòng. Kí hiệu: **Cột\$Dòng**.
 - **Tuyệt đối cột:** Địa chỉ cột không thay đổi. Kí hiệu: **\$CộtDòng**.

IV/ Các hàm cơ bản trên Excel

- ❑ Sau khi tính toán **hàm** sẽ trả về **một giá trị** (số, chuỗi,...)
- ❑ Dạng thức chung của hàm:

<TÊN HÀM>(**Đôi số 1, đôi số 2,..., đôi số n**)

- ❑ Một hàm có thể có một đôi số, nhiều đôi số, hoặc có hàm không có đôi số nào.
- ❑ Các đôi số trong hàm ngăn cách nhau bởi **dấu phẩy (,)** hay **dấu chấm phẩy (;)** là tùy thuộc vào việc thiết đặt trong **Regional Options** của mỗi máy.
- ❑ Ví dụ: **=SUM(10;C8;A2:C6)**

Tên hàm

Các đôi số

1/ Khái niệm đối số (hay tham số)

□ Đối số có thể là:

- Một giá trị kiểu số (Ví dụ: 6)
- Một chuỗi ký tự (Ví dụ: “TP”)
- Một địa chỉ ô hay vùng địa chỉ ô (Ví dụ: A2 hay A3: E9)
- Tên vùng (Ví dụ: NKCNO)
- Biểu thức đại số (Ví dụ: $2*3+6/4$)
- Những hàm khác.

2/ Các hàm toán học

□ Hàm SUM:

- **Cú pháp: =SUM(đối số 1; đối số 2;...;đối số n)**
- **Công dụng:** Tính tổng các đối số.
- **Chú ý:** Đối số phải là một giá trị kiểu số, có tối đa 30 đối số.
- **Ví dụ: =SUM(10;20;30) -> 60**

□ Hàm ABS

- **Cú pháp: =ABS(đối số)**
- **Công dụng:** Hàm sẽ trả về giá trị tuyệt đối của đối số.
- **Ví dụ: =ABS(-17) -> 17**

□ Hàm INT

- **Cú pháp: =INT(đối số)**
- **Công dụng:** Trả về phần nguyên của đối số.
- **Ví dụ: =INT(15,456) -> 15**

2/ Các hàm toán học (tt)

□ Hàm MOD:

- **Cú pháp: =MOD(đối số 1; đối số 2)**
- **Công dụng:** Trả về số dư khi thực hiện phép chia đối số 1 cho đối số 2.
- **Ví dụ: =MOD(10;3) -> 1**

□ Hàm ROUND

- **Cú pháp: =ROUND(Tham số; Vị trí)**
 - ✓ =0: Làm tròn tại vị trí dấu thập phân.
 - ✓ >0: Làm đến số chữ số lẻ được chỉ định.
 - ✓ <0: Làm tròn ở phần nguyên với số ký số bạn chỉ ra.
- **Công dụng:** Trả về giá trị của tham số sau khi đã làm tròn
- **Ví dụ:**
 - ✓ =Round(65,617;0) -> **66**
 - ✓ =Round(65,617;2) -> **65,62**
 - ✓ =Round(65,617;-2) -> **100**

2/ Các hàm toán học (tt)

□ Hàm SQRT

- **Cú pháp: =SQRT(đối số)**
- **Công dụng:** Trả về căn bậc 2 của đối số.
- **Ví dụ: =SQRT(9) -> 3**
- **Chú ý:** Nếu đối số là số âm hàm sẽ trả về lỗi **#NUM!**

3/ Các hàm luận lý

□ Hàm IF:

- **Cú pháp: =IF(logical_test;value_if_true;value_if_false)**
 - ✓ **Logical_test:** điều kiện để xét, logical có thể là kết quả của một hàm luận lý như AND, OR,...
 - ✓ **Value_if_true:** giá trị trả về nếu điều kiện logical_test là **TRUE**.
 - ✓ **Value_if_false:** giá trị trả về nếu điều kiện logical_test là **FALSE**.
- **Công dụng:** Nếu **biểu thức logic đúng** thì hàm IF trả về **đối số thứ nhì**, ngược lại nếu **biểu thức logic sai** thì hàm IF trả về **đối số thứ 3**.
- **Chú ý:**
 - ✓ Có thể có 7 hàm IF được lồng vào nhau để tạo nên công thức phức tạp hơn.
 - ✓ Biểu thức điều kiện phải có giá trị là TRUE hoặc FALSE, nếu không hàm IF trả về giá trị **#VALUE**

3/ Các hàm luận lý (tt)

□ Hàm IF: (tt)

- Ví dụ: =IF(3>5;“Sai”;“Đúng”)=?

□ Hàm AND:

- **Cú pháp**: =AND(logical_1,logical_2,...)

logical_1,logical_2,... là các điều kiện cần kiểm tra.

- **Công dụng**: Trả về kết quả TRUE nếu tất cả điều kiện đều TRUE, trả về FALSE nếu một trong các điều kiện FALSE.

- **Ví dụ**:

✓ =AND(1>0; 2>=2; 4<5; 4<=3) -> **FALSE**

✓ =AND(1>0; 2>=2; 4<5; 4<=4) -> **TRUE**

□ Hàm OR:

- **Cú pháp**: =OR(logical_1,logical_2,...)

- **Công dụng**: Trả về TRUE nếu một trong các điều kiện là TRUE. Trả về FALSE nếu tất cả các điều kiện là FALSE.

3/ Các hàm luận lý (tt)

□ Hàm OR: (tt)

➤ Ví dụ:

✓ =OR(1>5; 2>=3; 4<4; 4<=3) ->**FALSE**

✓ =OR(1>0; 2>=3; 4>5; 4<=3) ->**TRUE**

□ Hàm NOT:

➤ Cú pháp: **=NOT(logical)**

➤ Công dụng: Trả về phủ định của một biểu thức Logic

➤ Ví dụ:

✓ =NOT(2=2+1) ->**TRUE**

✓ =NOT(2<=2+1) ->**FALSE**

4/ Các hàm xử lý chuỗi

□ Hàm LEFT:

➤ **Cú pháp:** =LEFT(text;num_chars)

✓ **text** là chuỗi cần trích ký tự.

✓ **num_chars** là ký tự mà bạn cần trích bên trái chuỗi **text**.

➤ **Công dụng:** Trích bên trái một chuỗi một hoặc nhiều ký tự dựa vào số ký tự mà bạn chỉ định.

➤ **Chú ý:**

✓ **num_chars** không phải là số âm.

✓ **num_chars** nếu lớn hơn độ dài của chuỗi thì sẽ trả về toàn bộ chuỗi text.

✓ **num_chars** nếu bỏ qua thì mặc định là 1.

➤ **Ví dụ:** =Left("Trung tam tin hoc"; 5) -> **Trung**

□ **Hàm Right:** Tương tự hàm Left nhưng sẽ rút trích số ký tự bên phải của chuỗi.

4/ Các hàm xử lý chuỗi (tt)

□ Hàm MID:

- **Cú pháp: =MID(text;start_num,num_chars)**
 - ✓ **Text:** là chuỗi hoặc tham chiếu đến chuỗi.
 - ✓ **start_num:** vị trí bắt đầu trích lọc chuỗi con trong **text**.
 - ✓ **num_chars:** số ký tự của chuỗi mới cần trích từ chuỗi **text**.
- **Công dụng:** Trích một chuỗi con từ một chuỗi **text**, bắt đầu từ vị trí **start_num** với số ký tự được chỉ định **num_chars**.
- **Chú ý:**
 - ✓ **start_num:** lớn hơn chiều dài chuỗi text thì hàm trả về chuỗi rỗng ""
 - ✓ **start_num:** nhỏ hơn 1 hàm trả về lỗi #VALUE!
 - ✓ **num_chars:** âm MID trả về lỗi #VALUE!
- **Ví dụ:** =MID("Trung tam tin hoc";11;7) -> tin hoc

4/ Các hàm xử lý chuỗi (tt)

□ Hàm LEN:

- Cú pháp: **=LEN(Chuỗi số)**
- Công dụng: Trả về độ dài (số ký tự) của chuỗi số.

□ Hàm VALUE:

- Cú pháp: **=VALUE(Chuỗi số)**
- Công dụng: Đổi chuỗi số ra thành dữ liệu kiểu số.

5/ Các hàm thống kê

□ Hàm AVERAGE:

➤ **Cú pháp: =AVERAGE(number1;number2;...)**

number1,number2,... có thể có từ 1 đến 30 số mà bạn cần tính trung bình cộng.

➤ **Công dụng:** Tính trung bình cộng của các đối số.

□ Hàm COUNT:

➤ **Cú pháp: =COUNT(value1;value2;...)**

value1,value2,... có thể có từ 1 đến 30 vùng giá trị mà bạn muốn đếm số ô chứa dữ liệu kiểu số.

➤ **Công dụng:** Đếm số ô dữ liệu kiểu số trong vùng tham chiếu.

□ Hàm MIN:

➤ **Cú pháp: =MIN(number1;number2;...)**

➤ **Công dụng:** Trả về số nhỏ nhất trong các đối số truyền vào.

5/ Các hàm thống kê (tt)

- ❑ **Hàm MAX:** Tương tự hàm MIN nhưng sẽ trả về số lớn nhất trong các đối số truyền vào.

- ❑ **Hàm RANK:**
 - **Cú pháp: =RANK(number; ref; order)**
 - ✓ **number** giá trị mà bạn cần tìm vị thứ.
 - ✓ **ref** là mảng hoặc vùng tham chiếu đến một danh sách giá trị kiểu số. Những giá trị không phải là số được bỏ qua.
 - ✓ **order:** phương thức sắp xếp:
 - **order = 0**, hoặc bỏ qua thì số lớn nhất có vị trí nhỏ nhất (Sắp xếp giảm).
 - **order = 1** thì số nhỏ nhất có vị trí nhỏ nhất (Sắp xếp tăng).
 - **Công dụng:** Tìm vị thứ của một số trong dãy số.

5/ Các hàm thống kê (tt)

□ Hàm COUNTIF:

- Cú pháp: **=COUNTIF(Dãy kiểm tra; Điều kiện)**
 - ✓ **Vùng kiểm tra:** Là một vùng ô.
 - ✓ **Điều kiện:** Là điều kiện để đếm, có thể là hằng số hay địa chỉ của ô hay dạng thức ">16000".
- **Công dụng:** Hàm **Countif** trả về một số bằng cách tìm trong **Dãy kiểm tra** có bao nhiêu ô thỏa mãn **điều kiện**.

□ Hàm SUMIF:

- Cú pháp: **=SUMIF(Dãy kiểm tra; Điều kiện; Dãy tính tổng)**
 - ✓ **Dãy tính tổng:** Tùy theo cách tính toán mà có thể có dãy tính tổng hay không.
- **Công dụng:** Tính tổng của **dãy tính tổng** thoả **dãy kiểm tra** đối với **điều kiện**.

6/ Các hàm ngày tháng

□ Hàm TODAY:

- **Cú pháp: =TODAY()**
- **Công dụng:** Trả về ngày hiện tại trong hệ thống của bạn. Nếu định dạng ô là General trước khi hàm nhập công thức, kết quả trả về ở định dạng ngày tháng.

□ Hàm DATE:

- **Cú pháp: =DATE(year; month; day)**
- **Công dụng:** Trả về một chuỗi hoặc một số thể hiện một ngày tháng đầy đủ. Nếu định dạng ô là **General** trước khi nhập hàm thì kết quả trả về là chuỗi ngày tháng.

6/ Các hàm ngày tháng (tt)

☐ Hàm DAY:

➤ Cú pháp: **=DAY(serial_number)**

✓ **serial_number** dạng chuỗi số tuần tự của ngày cần tìm. Ngày tháng này nên nhập bằng hàm DATE hoặc kết quả trả về từ hàm khác.

✓ Có thể bị lỗi nếu bạn nhập serial_number là một chuỗi dạng văn bản.

➤ **Công dụng:** Trả về thứ tự của ngày từ chuỗi ngày tháng.

☐ Hàm **Month** và hàm **Year**: Tương tự như hàm **Day** nhưng sẽ cho giá trị **tháng** và **năm**.

7/ Các hàm tìm kiếm & tham chiếu

□ Hàm VLOOKUP:

- **Cú pháp:** =VLOOKUP(Giá trị dò tìm; Bảng chứa giá trị dò tìm; Cột trả về của bảng chứa giá trị dò tìm; Kiểu dò tìm)
 - ✓ VLOOKUP xuất phát từ **vertical lookup**: dò tìm theo phương đứng, hay theo cột.
 - ✓ Kiểu dò tìm: Dùng kiểu dò tìm là 0
- **Công dụng:** Dò tìm một giá trị ở cột đầu tiên bên trái của một bảng chứa giá trị dò tìm. Nếu tìm thấy sẽ trả về giá trị ở cùng trên cùng một dòng với giá trị tìm thấy trên cột mà bạn chỉ định. Hàm VLOOKUP thường dùng để điền thông tin vào bảng dữ liệu từ bảng dữ liệu phụ.

□ Hàm HLOOKUP: Tương tự như hàm VLOOKUP nhưng lúc này dò tìm theo **dòng**.

7/ Các hàm tìm kiếm & tham chiếu (tt)

□ Hàm INDEX:

- Cú pháp: **=INDEX(array; row_num; column_num)**
 - ✓ **array:** Là một mảng .
 - ✓ **row_num:** Chỉ số dòng cần trả về giá trị.
 - ✓ **column_num:** Chỉ số cột cần trả về giá trị.
- **Công dụng:** Trả về một giá trị nằm bên trong một mảng dựa vào chỉ số dòng và cột.

□ Hàm MATCH:

- Cú pháp: **=MATCH(lookup_value; lookup_array; match_type)**
 - ✓ **Lookup_value:** Là giá trị cần tìm trong bảng giá trị.
 - ✓ **Lookup_array:** Là một dãy ô liên tục để tìm kiếm giá trị.
 - ✓ **Match_type:** Dùng kiểu dò tìm là 0.
- **Công dụng:** Trả về vị trí (chỉ mục) của một giá trị từ một dãy giá trị.

BÀI 6

CƠ SỞ DỮ LIỆU TRÊN EXCEL

I/ Khái niệm về Cơ sở dữ liệu

- ❑ Tập hợp các thông tin về một đối tượng được lưu trữ trên máy tính, nhằm phục vụ cho công việc quản lý hay truy xuất đối tượng đó, được gọi là cơ sở dữ liệu (CSDL).
- ❑ Một CSDL bao gồm một hoặc nhiều bản ghi (Record) dữ liệu khác nhau, trong mỗi bản ghi có một hay nhiều trường dữ liệu (Field).

II/ Tổ chức vùng điều kiện

- ❑ Muốn rút trích dữ liệu theo một điều kiện nào đó từ CSDL của Excel thì trước hết phải tạo **vùng điều kiện**.
- ❑ Vùng điều kiện phải có tối thiểu là một dòng và một cột. Được tổ chức như sau:
 - Nếu chỉ có một điều kiện (điều kiện đơn):

Field
Điều kiện

- Nếu có nhiều điều kiện cho cùng một trường mang tính chất **hoặc**:

Field
Điều kiện 1
Điều kiện 2

II/ Tổ chức vùng điều kiện

- Nếu điều kiện kết hợp với nhiều trường mang tính chất **và**:

Field1	Field2
Điều kiện 1	Điều kiện 2

- Nếu điều kiện kết hợp với nhiều trường mang tính chất **hoặc**:

Field1	Field2
Điều kiện 1	
	Điều kiện 2

III/ Rút trích dữ liệu

- Đặt con trỏ vào vùng dữ liệu cần rút trích.
- Vào menu **Data -> Filter -> Advanced Filter**

- **Filter the list, in-place:**

Kết quả rút trích sẽ xuất hiện ngay trên CSDL ban đầu.

- **Copy to another location:**

Chọn nơi khác để thể hiện kết quả rút trích.

- **List range:** Chọn vùng CSDL ban đầu.

- **Criteria range:** Chọn vùng điều kiện rút trích.

- **Copy to:** Chọn nơi cần thể hiện kết quả rút trích.

Advanced Filter

Action _____

Filter the list, in-place

Copy to another location

List range: _____

Criteria range: _____

Copy to: _____

Unique records only

OK Cancel

IV/ Tìm kiếm/Lọc dữ liệu

- Vào menu **Data -> Filter -> Autofilter ->** Tất cả các trường trong CSDL sẽ được thể hiện dưới dạng một **Combo Box** gồm một danh sách các giá trị được lọc tự động trong các giá trị hiện tại nó đang chứa. Thông qua các **Combo Box** này, người dùng có thể tìm kiếm hoặc trích lọc thông tin một cách nhanh chóng.

	STT	HỌ	TÊN	NGÀY SINH
2				
3	9	Trần Thanh	Hà	24/12/1981
4	4	Cao Thị	Hồng	03/01/1984
5	7	Nguyễn Thị Hồng	Lan	03/01/1984
6	2	Nguyễn Thị Hồng	Lan	03/04/1983
7	1	Nguyễn Thị	Lan	03/01/1984
8	6	Nguyễn Thị	Lan	02/04/1981

BÀI 7

IN ẤN VÀ BẢO VỆ BẢNG TÍNH

I/ Cài đặt trang in

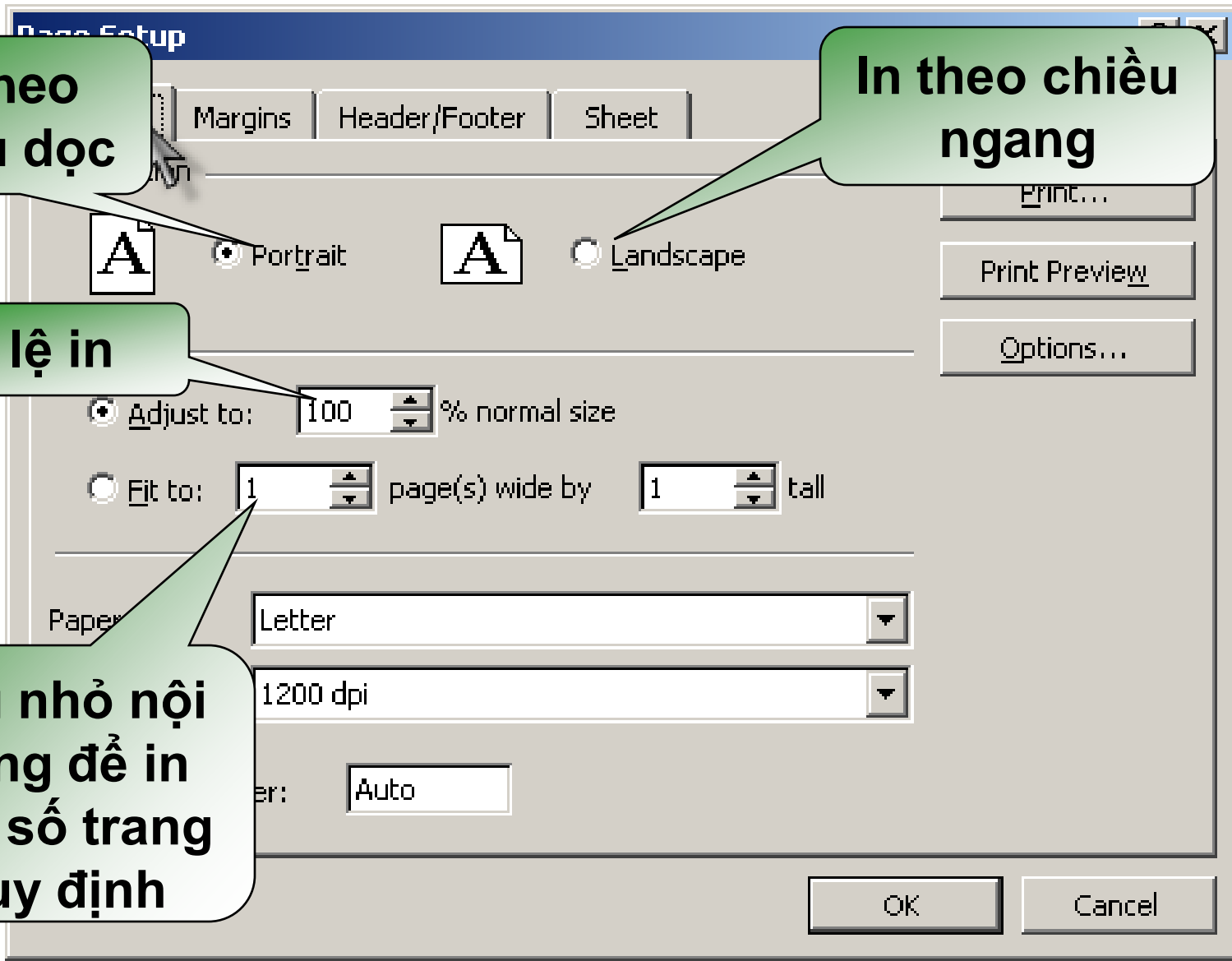
- ❑ Vào menu **File -> Page Setup...Tab Page**

In theo chiều dọc

In theo chiều ngang

Tỉ lệ in

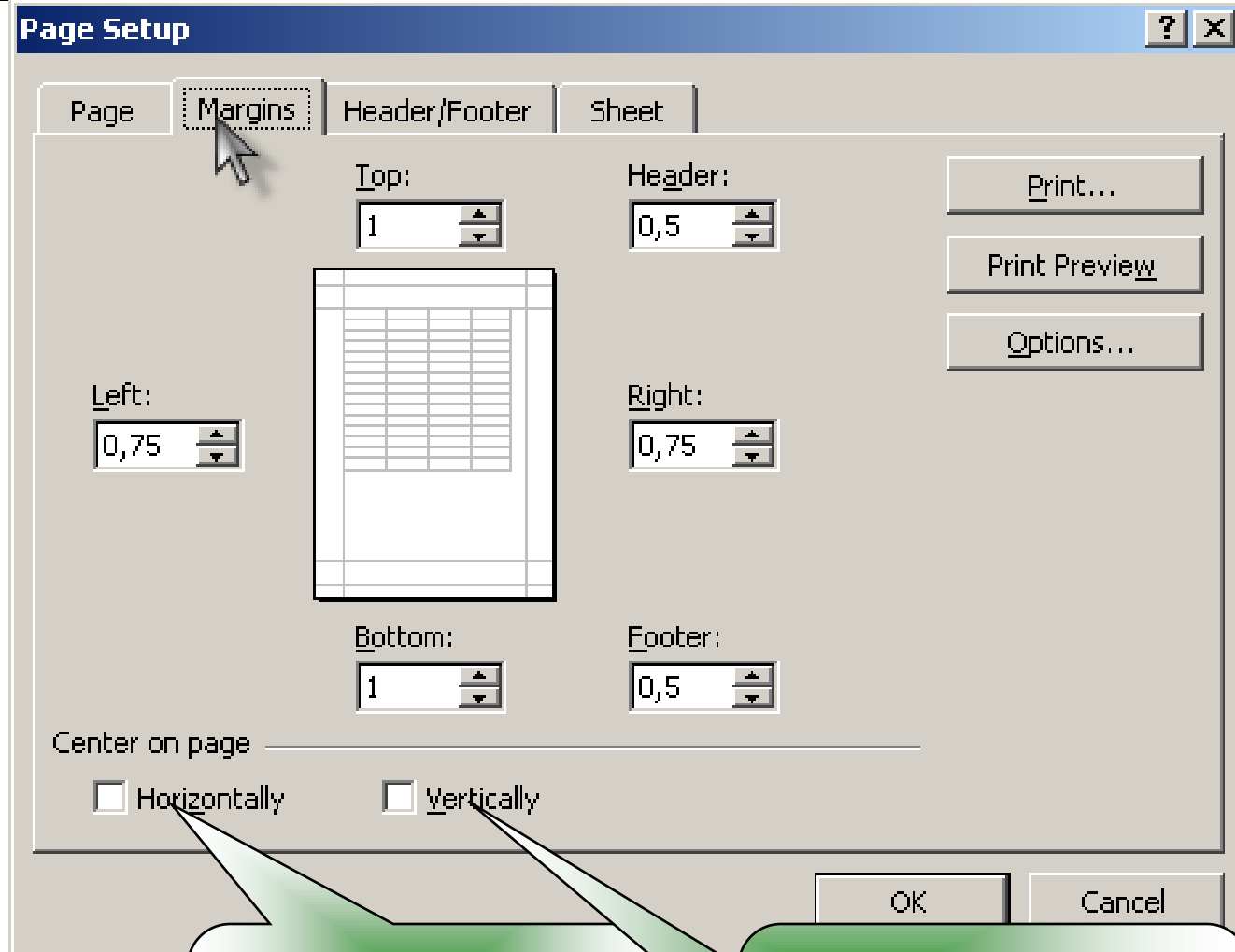
Thu nhỏ nội dung để in vào số trang quy định



I/ Cài đặt trang in (tt)

☐ Tab **Margin**:

- **Top:** lề trên
- **Left:** Lề trái
- **Right:** Lề phải
- **Bottom:** Lề dưới
- **Header:** Khoảng cách từ lề trên giấy đến tiêu đề đầu.
- **Footer:** Khoảng cách từ lề trên giấy đến tiêu đề đầu.



Canh giữa nội dung theo chiều ngang

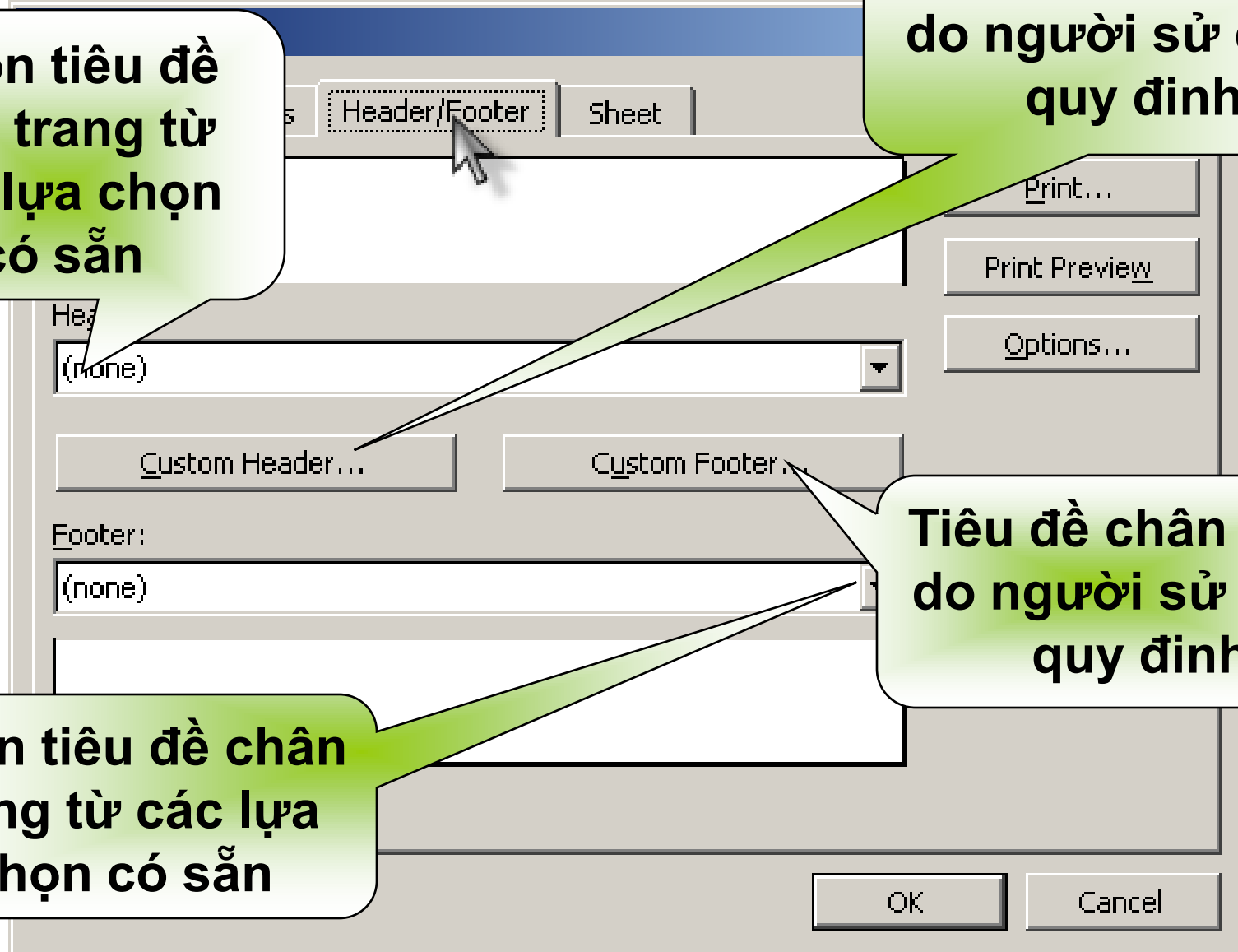
Canh giữa nội dung theo chiều dọc

I/ Cài đặt trang in (tt)

Tab **Header/Footer:**

Chọn tiêu đề đầu trang từ các lựa chọn có sẵn

Tiêu đề đầu trang do người sử dụng quy định



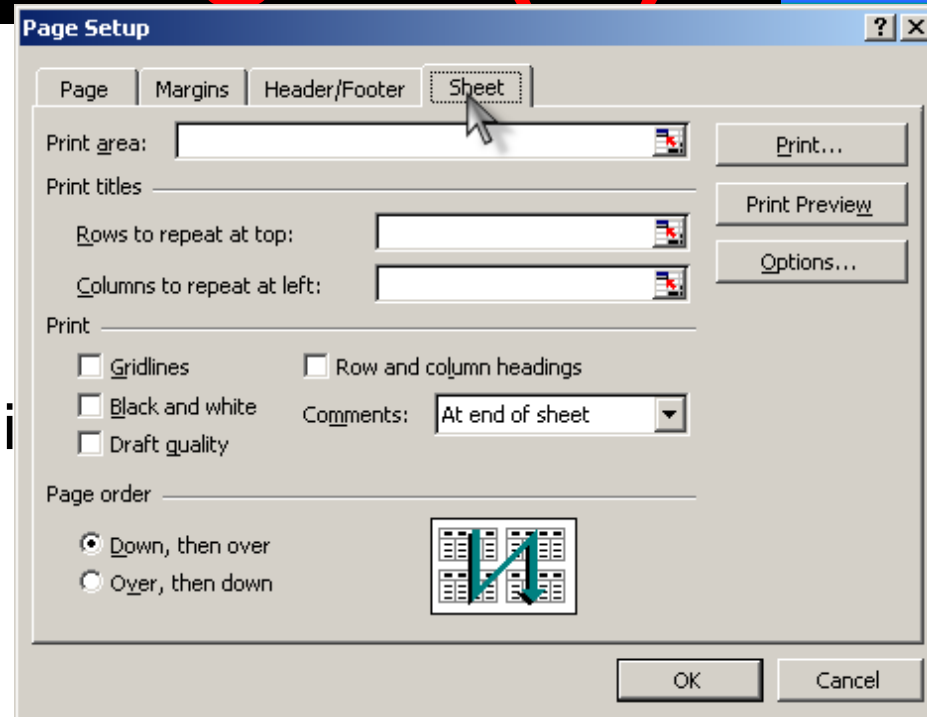
Chọn tiêu đề chân trang từ các lựa chọn có sẵn

Tiêu đề chân trang do người sử dụng quy định

I/ Cài đặt trang in (tt)

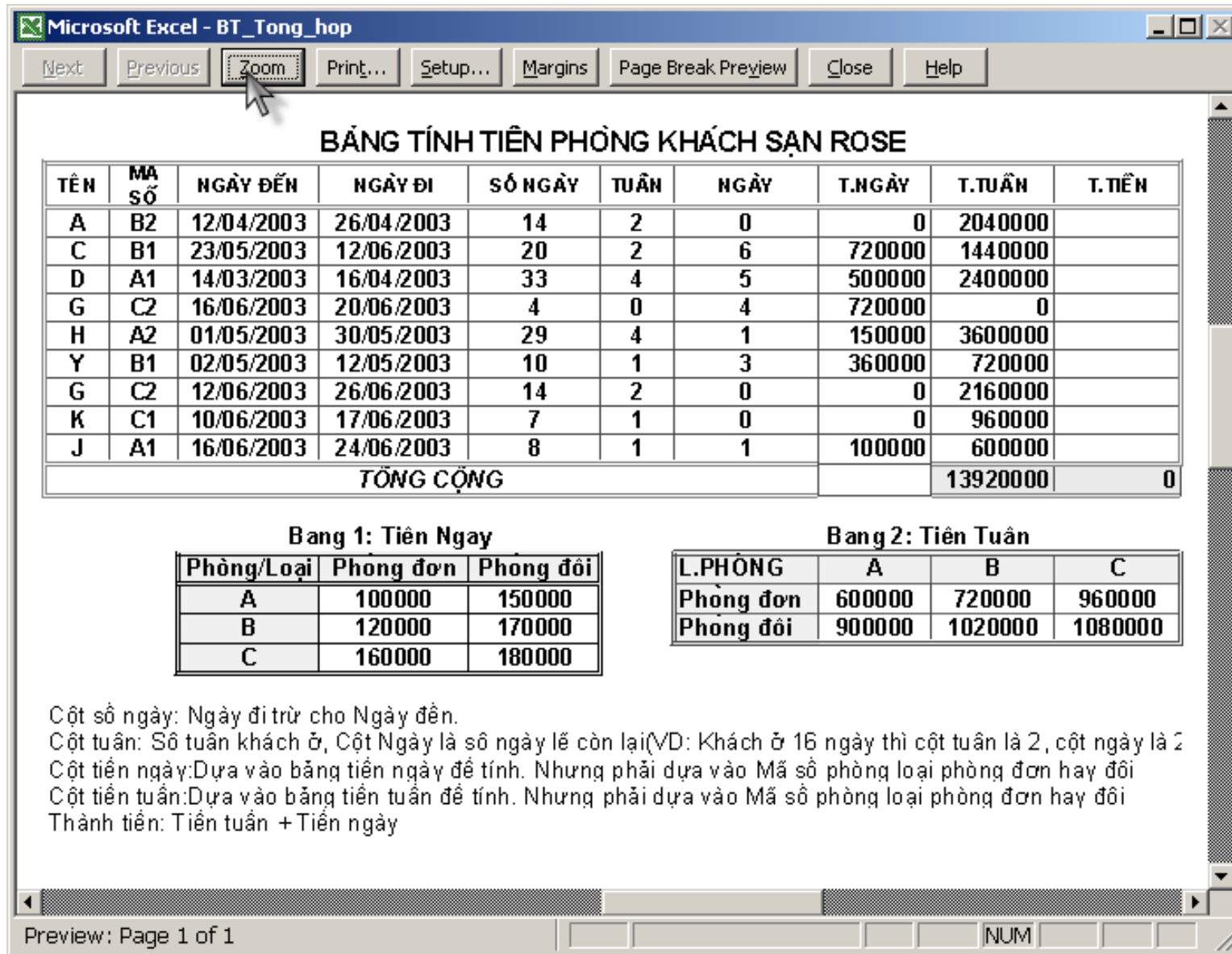
☐ Tab Header/Footer:

- ❖ **Print Area:** Vùng dữ liệu cần in.
- ❖ **Row to repeat at top:** In lại tiêu đề dòng.
- ❖ **Columns to repeat at left:** In lại tiêu đề cột.
- ❖ **Gridlines:** In đường lưới a Excel.
- ❖ **Black and White:** in phần văn bản, không in màu n.
- ❖ **Draft quality:** Không in các đường viền do người sử dụng kẻ.
- ❖ **Row and column headings:** dòng và t.
- ❖ **Down, then over:** In từ ng dưới, sau đó qua phải.
- ❖ **Over, then down :** In từ trái sang phải, sau đó ng dưới.



II/ Xem trước khi in

- Vào menu **File -> Print Preview**



The screenshot shows the Microsoft Excel interface with the title bar "Microsoft Excel - BT_Tong_hop". The menu bar includes "Next", "Previous", "Zoom", "Print...", "Setup...", "Margins", "Page Break Preview", "Close", and "Help". A mouse cursor is pointing at the "Zoom" button. The main content is a table titled "BẢNG TÍNH TIỀN PHÒNG KHÁCH SẠN ROSE".

TÊN	MA SỐ	NGÀY ĐẾN	NGÀY ĐI	SỐ NGÀY	TUẦN	NGÀY	T.NGÀY	T.TUẦN	T.TIỀN
A	B2	12/04/2003	26/04/2003	14	2	0	0	2040000	
C	B1	23/05/2003	12/06/2003	20	2	6	720000	1440000	
D	A1	14/03/2003	16/04/2003	33	4	5	500000	2400000	
G	C2	16/06/2003	20/06/2003	4	0	4	720000	0	
H	A2	01/05/2003	30/05/2003	29	4	1	150000	3600000	
Y	B1	02/05/2003	12/05/2003	10	1	3	360000	720000	
G	C2	12/06/2003	26/06/2003	14	2	0	0	2160000	
K	C1	10/06/2003	17/06/2003	7	1	0	0	960000	
J	A1	16/06/2003	24/06/2003	8	1	1	100000	600000	
TỔNG CỘNG								13920000	0

Below the main table are two smaller tables:

Bang 1: Tiền Ngày

Phòng/Loại	Phòng đơn	Phòng đôi
A	100000	150000
B	120000	170000
C	160000	180000

Bang 2: Tiền Tuần

L.PHÒNG	A	B	C
Phòng đơn	600000	720000	960000
Phòng đôi	900000	1020000	1080000

At the bottom, there is explanatory text:

Cột số ngày: Ngày đi trừ cho Ngày đến.
Cột tuần: Số tuần khách ở, Cột Ngày là số ngày lễ còn lại(VD: Khách ở 16 ngày thì cột tuần là 2, cột ngày là 2
Cột tiền ngày:Dựa vào bảng tiền ngày để tính. Nhưng phải dựa vào Mã số phòng loại phòng đơn hay đôi
Cột tiền tuần:Dựa vào bảng tiền tuần để tính. Nhưng phải dựa vào Mã số phòng loại phòng đơn hay đôi
Thành tiền: Tiền tuần + Tiền ngày

The status bar at the bottom shows "Preview: Page 1 of 1" and "NUM".

II/ Xem trước khi in

☐ Thanh công cụ **Print Preview**:

Về trang trước

Thực hiện in

Hiển thị các đường lề

Đóng cửa sổ

Next

Previous

Zoom

Print...

Setup...

Margins

Page Break Preview

Close

Help

Phóng to, thu nhỏ

Hiển thị chế độ ngắt trang

Trợ giúp

Sang trang tiếp

Định dạng lại trang in

III/ Bảo vệ bảng tính

- ❑ Đây là một tiện ích của Excel giúp cho bảng tính của người sử dụng không bị người khác sửa đổi công thức hoặc không xem được công thức.
- ❑ Để bảo vệ Worksheet nào thì đứng ở Worksheet đó. Vào menu **Tools -> Protection -> Protect Sheet...** Điền mật khẩu bảo vệ...
 - Khi đã bảo vệ bằng cách như trên thì người khác sẽ không thể sửa đổi được công thức trên trang bảng tính (vẫn có thể xem được công thức).
- ❑ Để cho người khác không thể xem được công thức trên ô hay trên thanh công thức thì trước khi làm thao tác như trên hãy chọn những ô nào muốn ẩn công thức và vào menu **Format -> Cells...** Qua tab **Protection**, check vào mục **Hidden**.

Khi bạn sinh ra bạn đã khóc

trong tiếng cười của mọi người.

Vậy bạn hãy sống làm sao

để khi bạn mất bạn sẽ cười

trong khi mọi người khóc! ...

HỌC PHẦN TIN HỌC ĐẠI CƯƠNG
ĐẾN ĐÂY LÀ KẾT THÚC!...



Tin Học Đại Cường



Nội dung chính

- Phần 1: Cài đặt và sử dụng hệ điều hành Windows XP.
- Phần 2: Soạn thảo văn bản với MS Word.
- Phần 3: (MS Word, MS Excel,...).

Phần 1



Cài đặt & sử dụng Hệ điều hành MS Windows XP

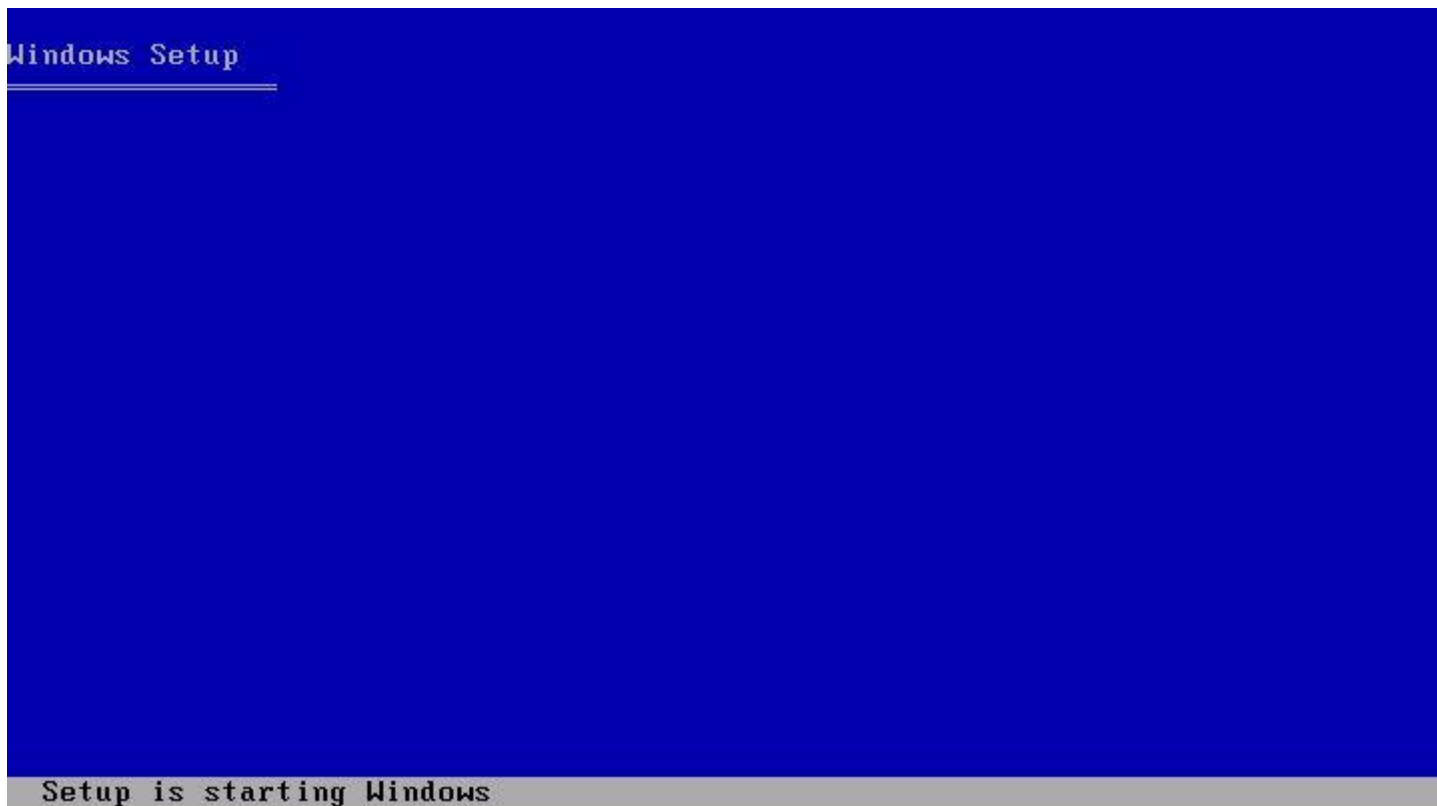


Bài 1: Cài đặt WindowsXP

- Cần phải có đĩa CD Windows
- Chuẩn bị máy, có đĩa cứng.
- Cho máy khởi động (boot) từ CD.

Bắt đầu cài đặt Đưa đĩa CD Win XP

Chú ý:
Máy
phải
đặt ở
chế độ
Boot từ
CD





Chọn dạng cài đặt

Enter:
tiếp tục
R: Sửa lỗi
windows
F3: Thoát

Windows XP Professional Setup

Welcome to Setup.

This portion of the Setup program prepares Microsoft(R)
Windows(R) XP to run on your computer.

- To set up Windows XP now, press ENTER.
- To repair a Windows XP installation using Recovery Console, press R.
- To quit Setup without installing Windows XP, press F3.

ENTER=Continue R=Repair F3=Quit

Màn hình thông báo bản quyền

F8: Đồng ý xác nhận bản quyền.
ESC: Không đồng ý

Windows XP Licensing Agreement

Microsoft Windows XP Professional

END-USER LICENSE AGREEMENT

IMPORTANT-READ CAREFULLY: This End-User License Agreement ("EULA") is a legal agreement between you (either an individual or a single entity) and Microsoft Corporation for the Microsoft software product identified above, which includes computer software and may include associated media, printed materials, "online" or electronic documentation, and Internet-based services ("Product"). An amendment or addendum to this EULA may accompany the Product. YOU AGREE TO BE BOUND BY THE TERMS OF THIS EULA BY INSTALLING, COPYING, OR OTHERWISE USING THE PRODUCT. IF YOU DO NOT AGREE, DO NOT INSTALL OR USE THE PRODUCT; YOU MAY RETURN IT TO YOUR PLACE OF PURCHASE FOR A FULL REFUND.

1. **GRANT OF LICENSE.** Microsoft grants you the following rights provided that you comply with all terms and conditions of this EULA:

* **Installation and use.** You may install, use, access, display and run one copy of the Product on a single computer, such as a workstation, terminal or other device ("Workstation Computer"). The Product may not be used by more than two (2) processors at any one time on any

F8=I agree ESC=I do not agree PAGE DOWN=Next Page



Phân vùng đĩa cứng

ENTER: Để cài WinXP vào phân vùng đã chọn.

C: Để tạo một ổ đĩa mới từ phân vùng trống đã chọn.

D: Để xoá phân vùng đã chọn và tạo ra một phân vùng trống

Windows XP Professional Setup

The following list shows the existing partitions and unpartitioned space on this computer.

Use the UP and DOWN ARROW keys to select an item in the list.

- To set up Windows XP on the selected item, press ENTER.
- To create a partition in the unpartitioned space, press C.
- To delete the selected partition, press D.

4095 MB Disk 0 at Id 0 on bus 0 on atapi [MBR]

Unpartitioned space	4095 MB
---------------------	---------

ENTER=Install C=Create Partition F3=Quit



Định dạng đĩa cứng

Chọn loại định dạng đĩa cứng (FAT32, NTFS).

Nên chọn NTFS (vì có tính bảo mật cao,...)

Windows XP Professional Setup

A new partition for Windows XP has been created on
4095 MB Disk 0 at Id 0 on bus 0 on atapi [MBR].

This partition must now be formatted.

From the list below, select a file system for the new partition.
Use the UP and DOWN ARROW keys to select the file system you want,
and then press ENTER.

If you want to select a different partition for Windows XP,
press ESC.

Format the partition using the NTFS file system (Quick)
Format the partition using the FAT file system (Quick)
Format the partition using the NTFS file system
Format the partition using the FAT file system

ENTER=Continue ESC=Cancel

Chờ định dạng đĩa

Windows XP Professional Setup

Please wait while Setup formats the partition

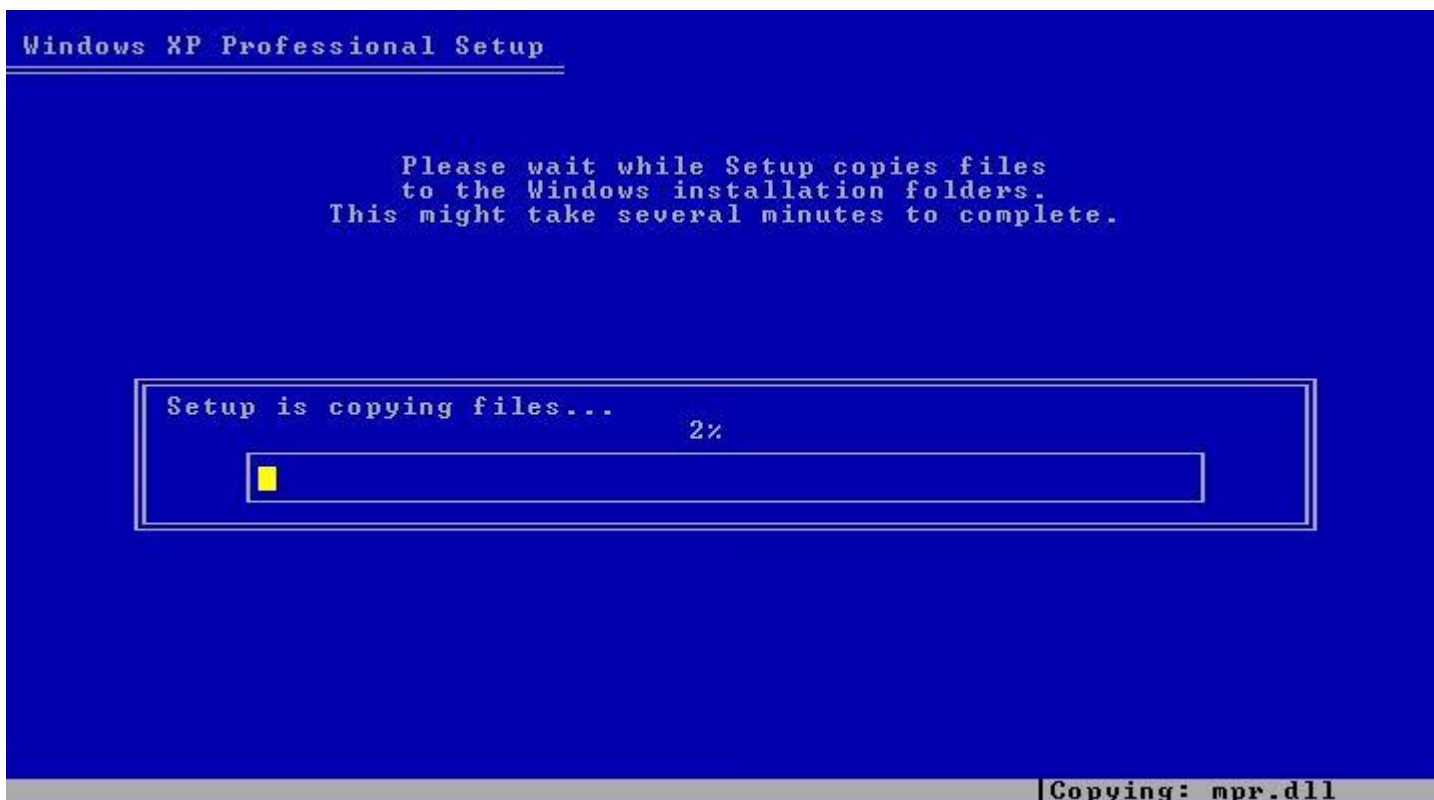
C: Partition1 [New <Raw>] 4087 MB (4086 MB free)
on 4095 MB Disk 0 at Id 0 on bus 0 on atapi [MBR].

Setup is formatting...

66%



Chờ chương trình cài đặt copy file






Tự động khởi động lại máy

Windows XP Professional Setup

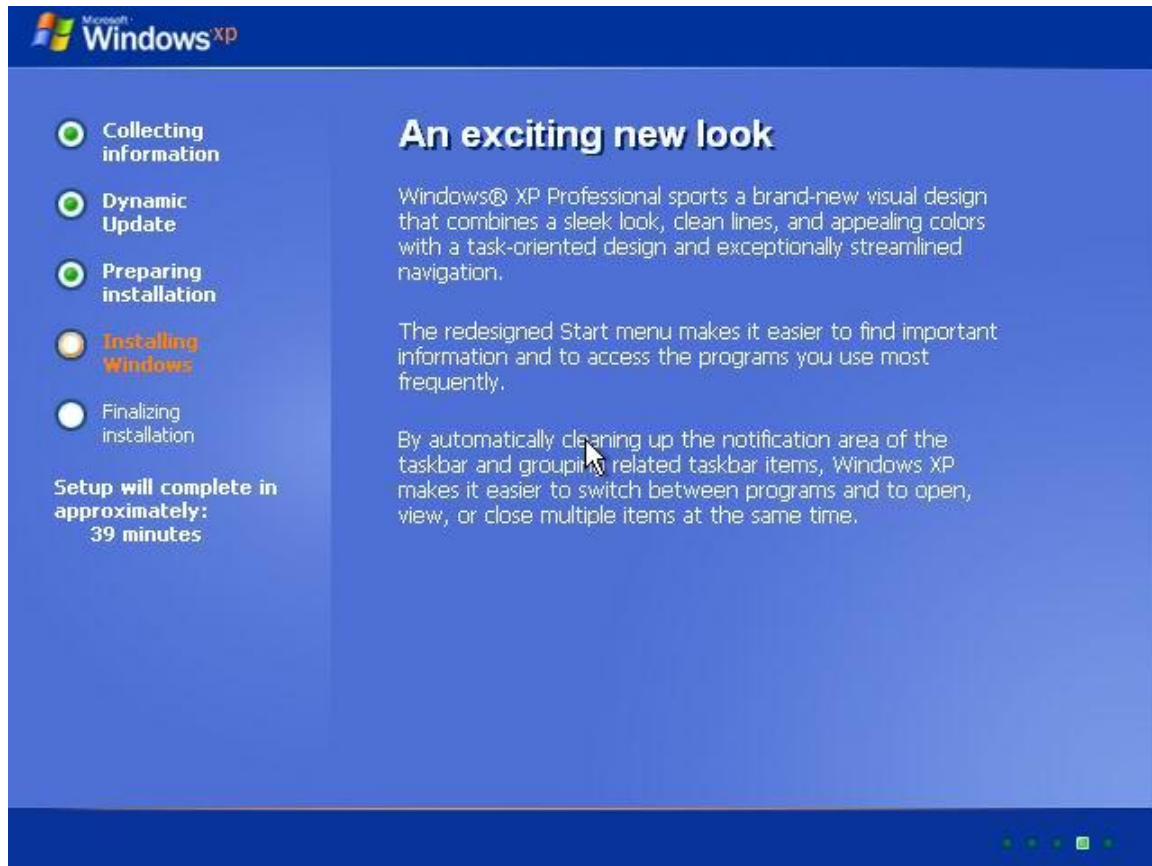
This portion of Setup has completed successfully.
If there is a floppy disk in drive A:, remove it.
To restart your computer, press ENTER.
When your computer restarts, Setup will continue.

Your computer will reboot in 8 seconds....



ENTER=Restart Computer

Chờ chương trình bắt đầu cài



Tiếp tục chờ

Microsoft
Windows^{XP}

- Collecting information
- Dynamic Update
- Preparing installation
- Installing Windows**
- Finalizing installation

**Setup will complete in approximately:
37 minutes**

Installing Devices
[Progress bar]

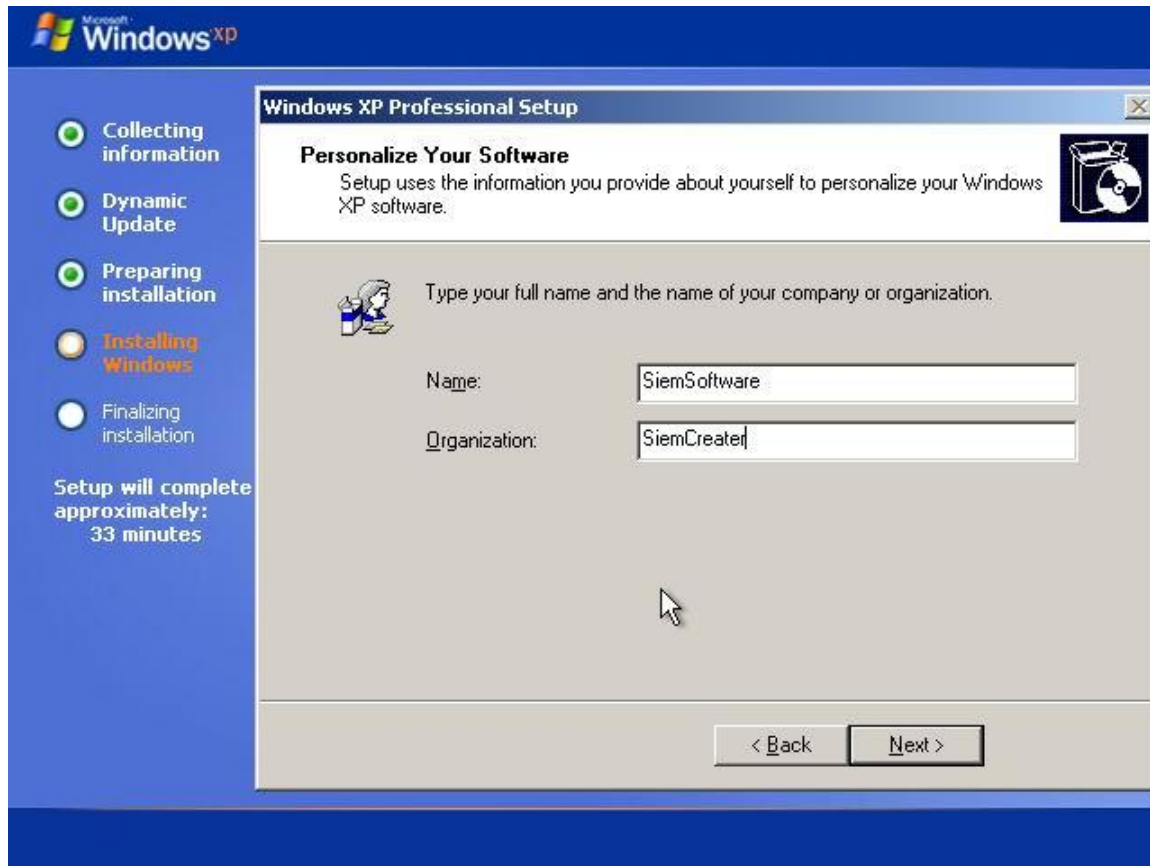
Easily move documents and personal settings to a new computer

If you've had a computer for a while, you've probably gotten it set up the way you like. Files are organized a certain way, you have bookmarked a long list of favorite Web sites, and you've customized your desktop just the way you want it.

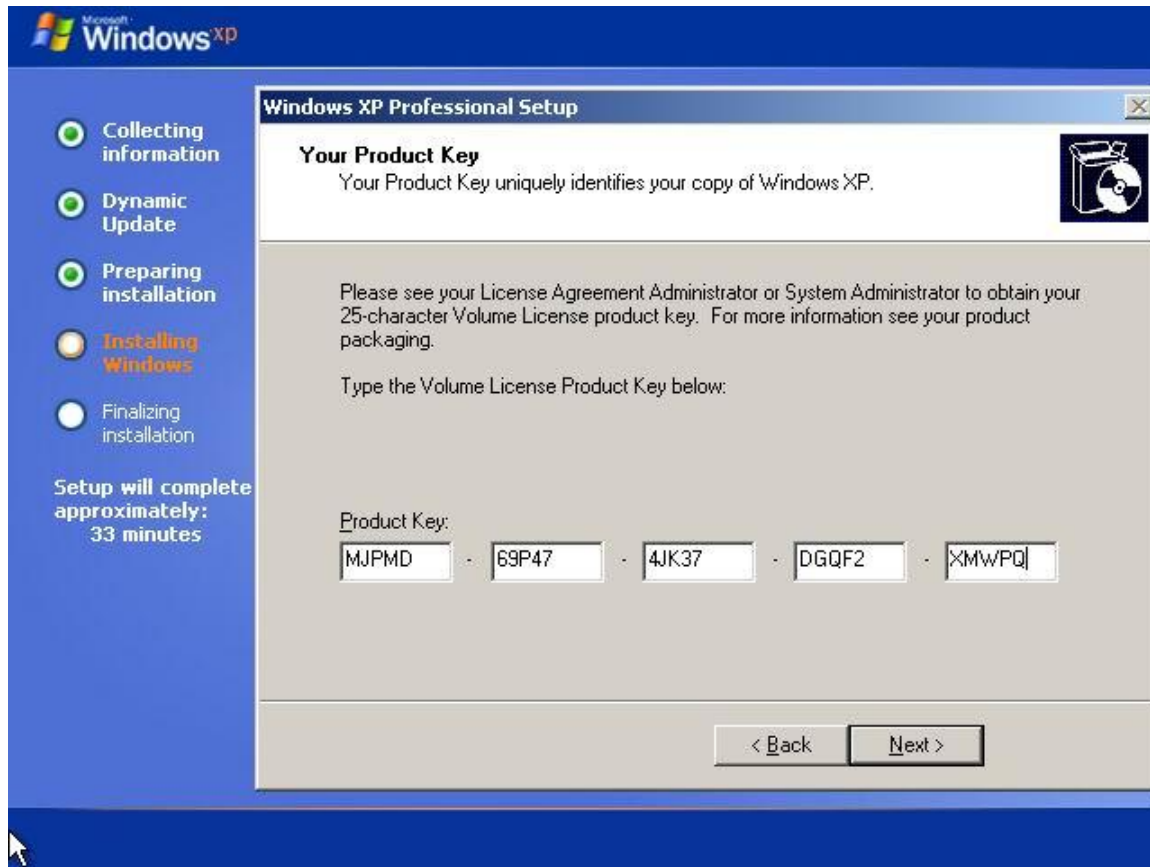
Those personal settings won't be wiped out when you buy a new computer with Windows® XP. With the Files and Settings Transfer Wizard, it's simple to move your files and settings from one computer to another.

The wizard is located in the System Tools folder. For a command-line version that system administrators can use on multiple computers at once, try the User State Migration Tool (USMT) in the ValueAdd folder.

Màn hình sau xuất hiện nhập tên(name) và tổ chức



Nhập mã số (CD key)



Nhập tên máy tính Password cho quản trị máy (administrator)



The screenshot shows the Windows XP Professional Setup window. The title bar reads "Windows XP Professional Setup". The main window title is "Computer Name and Administrator Password". Below the title, it says "You must provide a name and an Administrator password for your computer." There is a CD-ROM icon in the top right corner of the window.

On the left side of the window, there is a vertical navigation pane with five steps, each with a circular progress indicator:

- Collecting information (selected, green circle)
- Dynamic Update (green circle)
- Preparing installation (green circle)
- Installing Windows (orange circle)
- Finalizing installation (white circle)

Below the navigation pane, it says "Setup will complete approximately: 33 minutes".

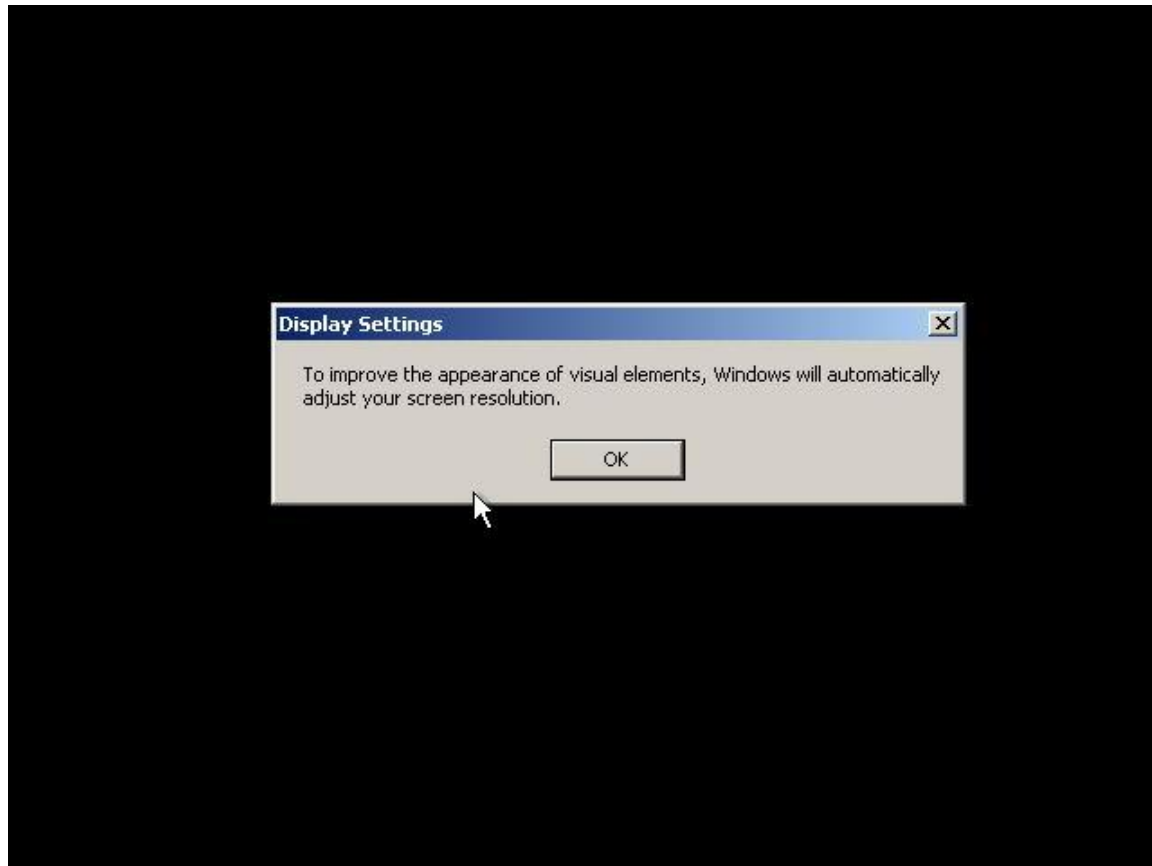
The main content area has the following sections:

- A computer icon with the text: "Setup has suggested a name for your computer. If your computer is on a network, your network administrator can tell you what name to use."
- A text input field labeled "Computer name:" containing the text "SIEMSOFTWARE".
- A person icon with the text: "Setup creates a user account called Administrator. You use this account when you need full access to your computer."
- The text "Type an Administrator password." followed by two password input fields, each containing six dots.
- At the bottom right, there are two buttons: "< Back" and "Next >".

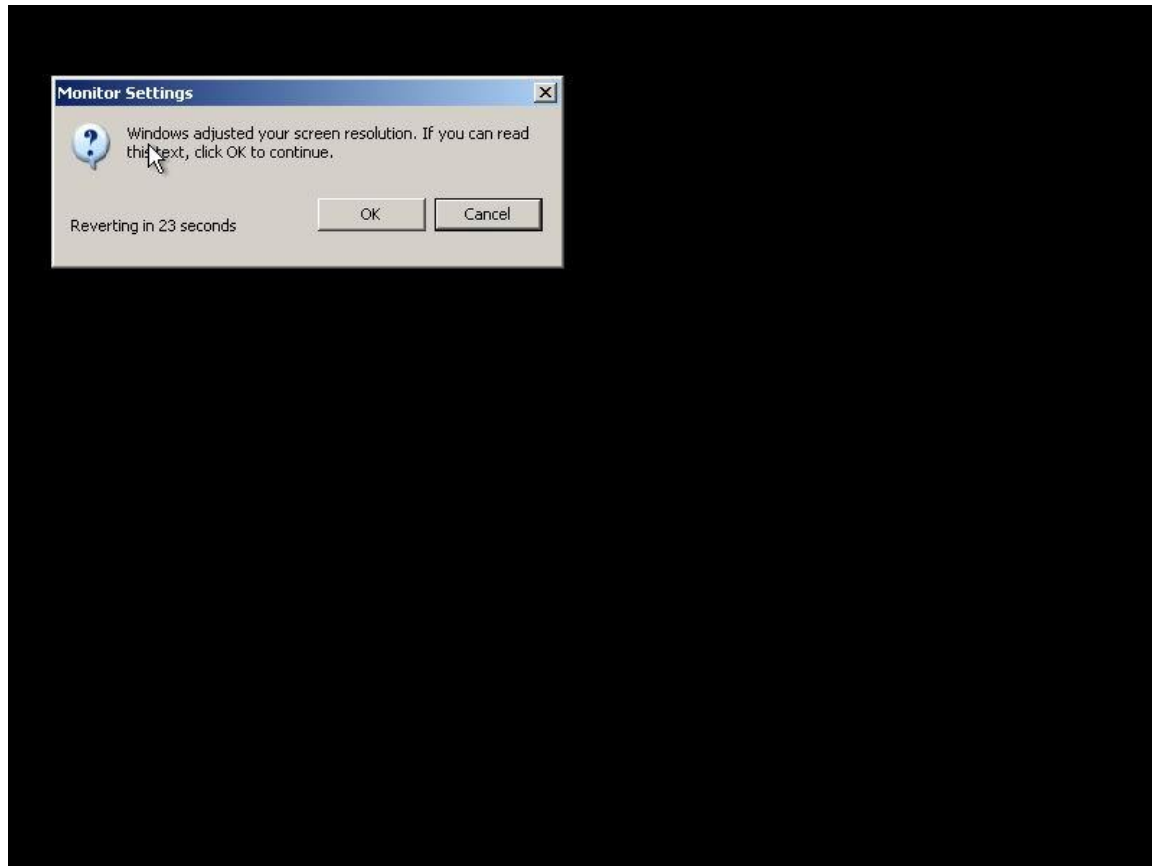
Chọn ngày giờ và múi giờ



Windows nhận dạng màn hình nhấn OK (nếu có)



Nhận dạng độ phân giải màn hình



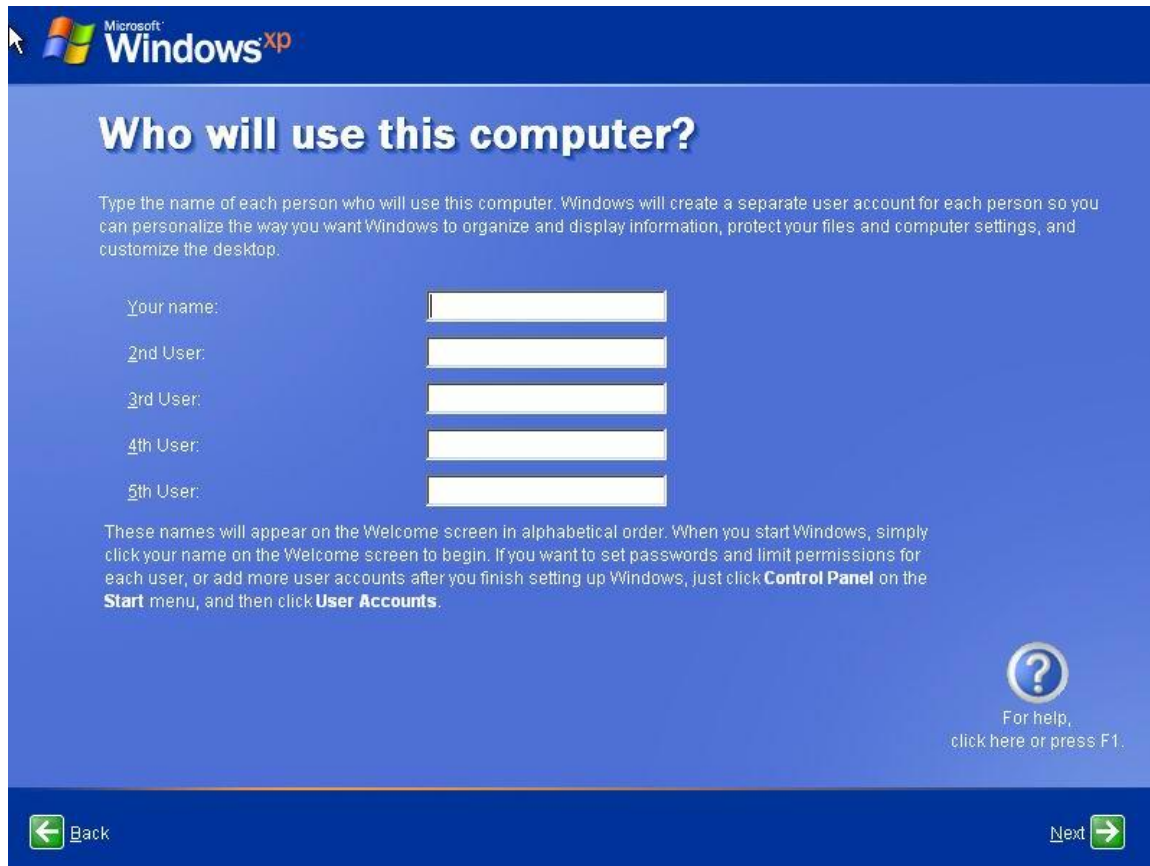
Màn hình xác nhận hoàn thành



Nhấn next để tiếp tục



Nhập tên các người dùng



Microsoft
Windows^{XP}

Who will use this computer?

Type the name of each person who will use this computer. Windows will create a separate user account for each person so you can personalize the way you want Windows to organize and display information, protect your files and computer settings, and customize the desktop.

Your name:


2nd User:



3rd User:

4th User:

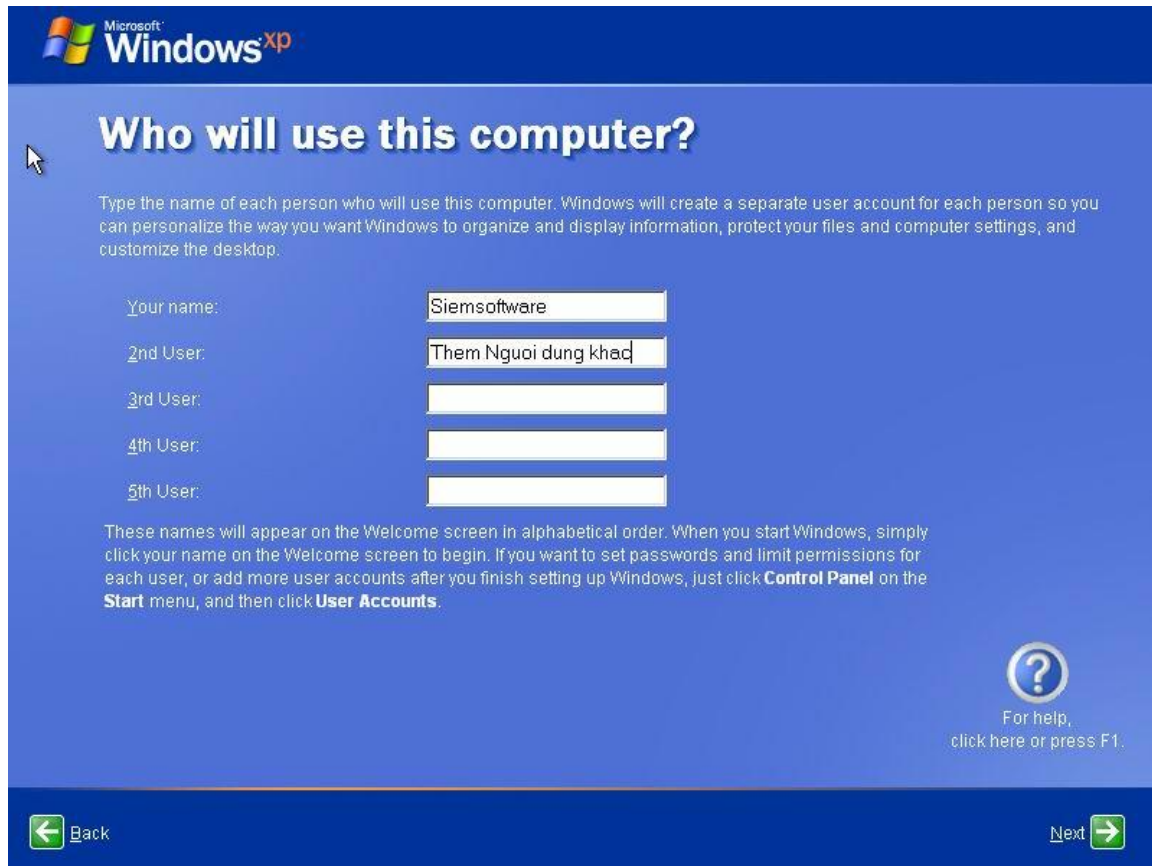
5th User:

These names will appear on the Welcome screen in alphabetical order. When you start Windows, simply click your name on the Welcome screen to begin. If you want to set passwords and limit permissions for each user, or add more user accounts after you finish setting up Windows, just click **Control Panel** on the **Start** menu, and then click **User Accounts**.

 For help, click here or press F1.

 Back  Next

Tên không trùng với tên máy



Microsoft
Windows^{XP}

Who will use this computer?

Type the name of each person who will use this computer. Windows will create a separate user account for each person so you can personalize the way you want Windows to organize and display information, protect your files and computer settings, and customize the desktop.

Your name:


2nd User:



3rd User:

4th User:

5th User:

These names will appear on the Welcome screen in alphabetical order. When you start Windows, simply click your name on the Welcome screen to begin. If you want to set passwords and limit permissions for each user, or add more user accounts after you finish setting up Windows, just click **Control Panel** on the **Start** menu, and then click **User Accounts**.

 For help, click here or press F1.

 Back Next 

Hoàn thành cài đặt



Bài 2: LÀM QUEN VỚI WINDOWS



- **Khái niệm hệ điều hành:**
 - Là tập các chương trình cơ sở có nhiệm vụ điều khiển phần cứng máy tính.
 - Làm nền tảng cho các chương trình ứng dụng.
 - Tạo ra môi trường giao tiếp giữa người và máy.



Các hệ điều hành phổ biến

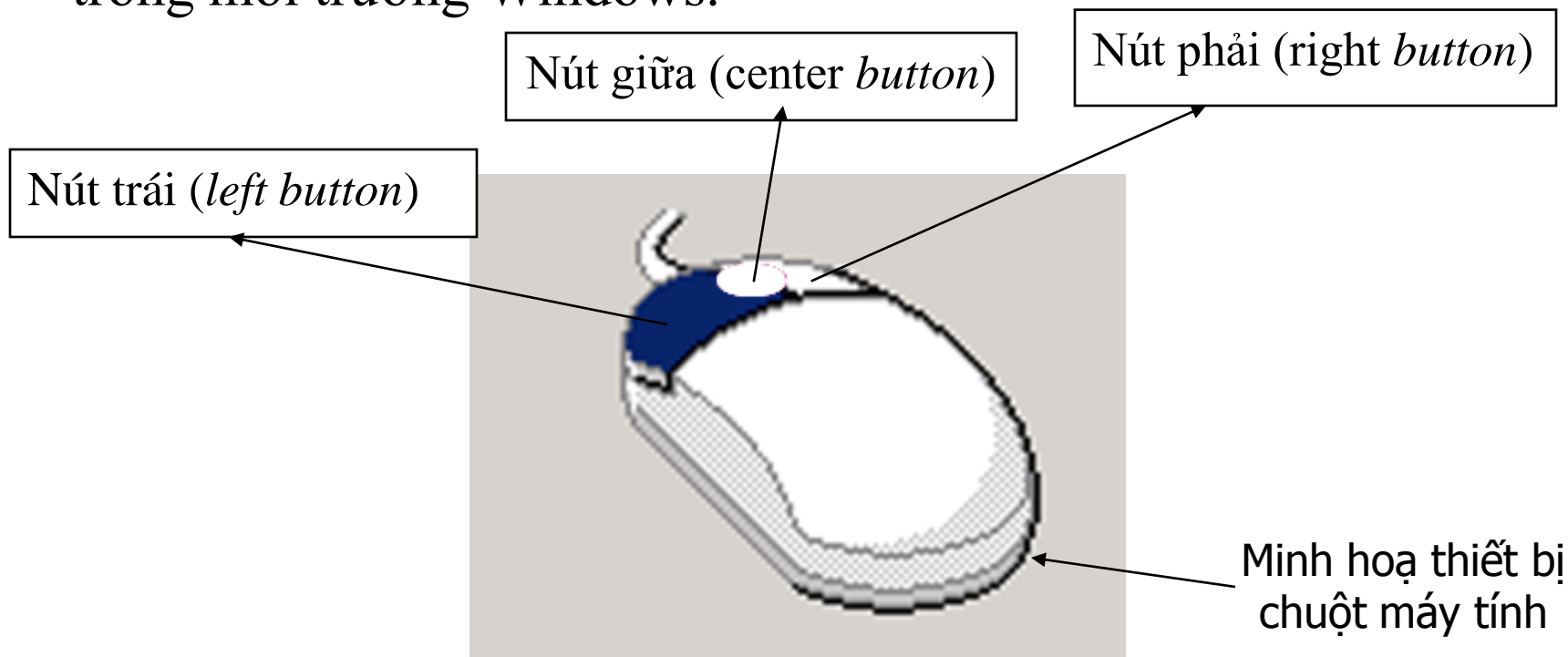
- **DOS (Disk Operating System):** Là HĐH đầu tiên của máy tính. Hiện nay không còn dùng phổ biến.
- **Microsoft Windows:** Là hệ điều hành phổ biến nhất hiện nay vì có ưu điểm dễ sử dụng. HĐH Windows có nhiều phiên bản như Windows 98, Windows 2000, Windows XP,...
- **Linux:** Là HĐH nguồn mở hoàn toàn miễn phí. Ưu điểm ổn định, nhược điểm khó sử dụng.

Các ưu điểm của HĐH Microsoft Windows

- Giao tiếp với người dùng thông qua **giao diện đồ họa**.
- Là **HĐH đa nhiệm**, có thể thực hiện nhiều trình ứng dụng song song cùng một lúc.
- Có thể lưu trữ tên tập tin dài đến **250 ký tự**.
- Sử dụng cơ chế **tự động tìm và nhận thiết bị phần cứng(Plug and Play)**.
- Cung cấp các khả năng có sẵn để nối mạng giữa các máy tính: chia sẻ tài nguyên, Email, Web, ...
- Hỗ trợ tốt thành phần **đa phương tiện (Multimedia)**.

Sử dụng chuột:

Chuột (mouse): Được sử dụng nhiều nhất trong WINDOWS. Không có chuột có lẽ không thể dùng được các chương trình trong môi trường Windows.





Một số động tác hay dùng đối với chuột

Nhấp trái (Click trái): Bấm nút trái chuột một lần. Thông thường nhấp trái chuột dùng để **chọn một đối tượng** nào đó.

Nhấp phải (Click phải): Bấm nút phải chuột 1 lần. Thông thường nhấp phải chuột dùng để **chọn một menu chọn** nào đó.

Nhấp đúp (Double Click): Bấm nút trái chuột 2 lần liên tiếp. Thông thường nhấp đúp chuột dùng để **khởi động một ứng dụng** nào đó.

Kéo - Rê (Drop - Drag): Nhấp và giữ nút (trái) chuột, di chuyển chuột đến vị trí nào đó và thả nút chuột. thông thường kéo – rê thường **kết hợp với phím Ctrl để copy một đối tượng** nào đó.



Thực hành chọn đối tượng

- Chọn 1 đối tượng
- Chọn nhiều đối tượng không liên tục
- Chọn 1 nhóm đối tượng liên tục



Màn hình Desktop:

Giao diện màn hình chính của Windows được gọi là “**Desktop**” bởi vì Màn hình nền Windows XP là cửa sổ đầu tiên của Hệ điều hành dành cho người sử dụng. Người dùng ra lệnh cho hệ điều hành bằng cách thao tác với biểu tượng.

Nơi cất giữ tài liệu do người dùng tạo ra

Nơi quản lý toàn bộ tài nguyên của máy tính như: ổ đĩa, thư mục, tập tin,...

Mạng máy tính: nơi truy xuất, chia sẻ tài nguyên của các máy khác trong mạng

Thùng rác: nơi lưu trữ những tập tin, thư mục bị xóa

Nút Start: chứa các chương trình được cài đặt vào máy tính

Thanh tác vụ: chứa các chương trình đang được mở

Khay hệ thống

start

minh

Adobe Acrobat Prof...

Microsoft PowerPoin...

<

100%

8:45 PM

Nhận biết biểu tượng



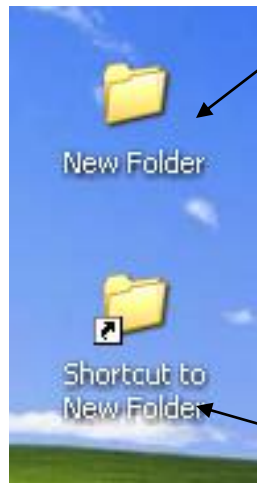
Biểu tượng đặc biệt của hệ điều hành

- Gồm biểu tượng My Documents, My Computer, My Network Places, Recycle Bin, Internet Explorer.
- Khi cài đặt xong hệ điều hành thì các biểu tượng này được tạo ra ngay trên màn hình nền.

Nhận biết biểu tượng

Biểu tượng của thư mục

- Một thư mục được hệ điều hành Windows biểu diễn bằng một biểu tượng. Hình ảnh của biểu tượng là túi hồ sơ màu vàng như hình minh họa và tên biểu tượng xuất hiện bên dưới.



Biểu tượng lối tắt cho thư mục

- Tiếng Anh gọi là **Shortcut**, có hình ảnh là túi hồ sơ màu vàng có thêm mũi tên ở góc dưới bên trái.

Thực hành chọn một biểu tượng

Để chọn một biểu tượng chúng ta thao tác đơn giản là nhấp chuột lên biểu tượng đó.



Biểu tượng ở trạng thái tự do

Biểu tượng ở trạng thái được chọn

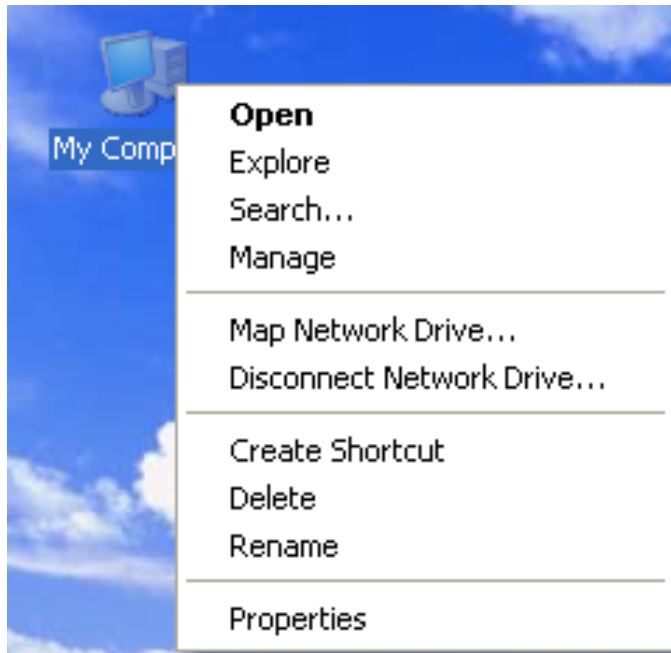


Thực hành di chuyển biểu tượng

- Đầu tiên là chọn biểu tượng My Computer, kéo di chuột. Một hình ảnh của biểu tượng xuất hiện ở dạng nét mờ thể hiện vị trí di chuyển của biểu tượng.

Minh họa biểu tượng di chuyển

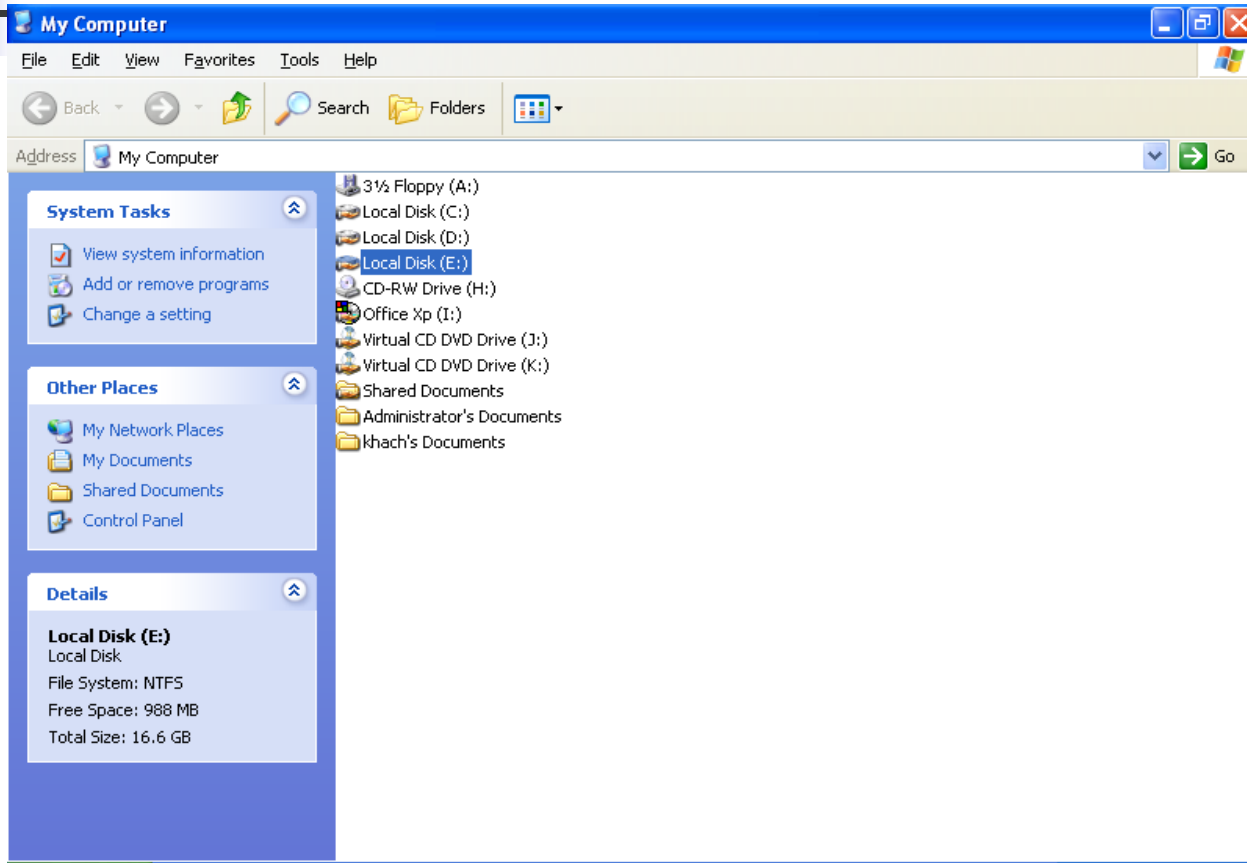
Thao tác với hộp lệnh (menu tắt)



- Đầu tiên là chọn biểu tượng, sau đó nhấp chuột phải trên vùng chọn sẽ làm xuất hiện hộp lệnh.
- Nhấp chuột trên mục lệnh của hộp lệnh có ý nghĩa là thi hành lệnh.

Hộp lệnh xuất hiện sau khi nhấp chuột phải

Thao tác với hộp lệnh



Cửa sổ My Computer xuất hiện sau khi thực hiện lệnh Open trên hộp lệnh



Xem thông tin ổ đĩa

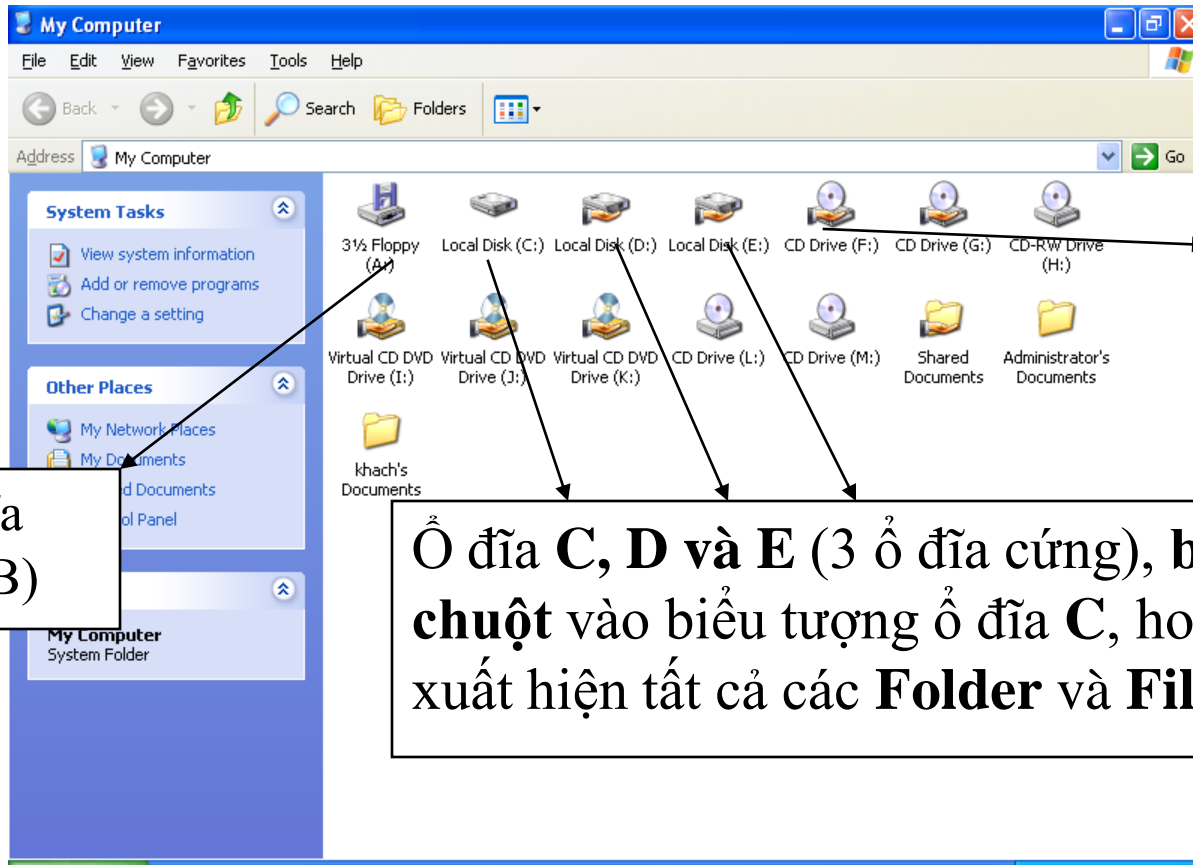
Từ màn hình Desktop, để nhìn thấy các ổ đĩa trên máy

+ Double Click vào **My Computer**: Đĩa mềm, ổ cứng, đĩa CD-ROM.

+ Double Click vào ổ đĩa, có thể duyệt qua các **File** và **Folder** từ một trong các ổ đĩa này.

Muốn trở về thư mục trước đó thì Click nút **Up**

Xem thông tin ổ đĩa



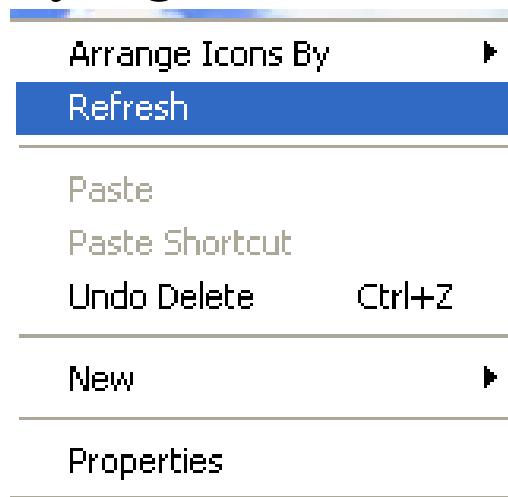
Ổ đĩa CD ROM(Đĩa Compact)

Ổ đĩa A (đĩa mềm 1.4 MB)

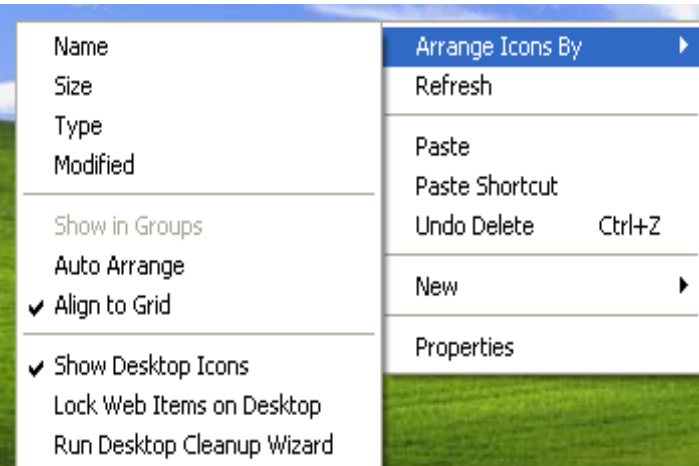
Ổ đĩa C, D và E (3 ổ đĩa cứng), **bấm đúp chuột** vào biểu tượng ổ đĩa C, hoặc D sẽ xuất hiện tất cả các **Folder** và **File**

Hộp lệnh của màn hình nền - Desktop

- Nhấp chuột phải vào chỗ trống trên màn hình nền sẽ làm xuất hiện hộp lệnh cho phép người sử dụng điều khiển màn hình.
- Chúng ta tìm hiểu ý nghĩa và thao tác từng lệnh sau:



Nhóm lệnh Arrange Icons



Mục lệnh Arrange Icons

Cho phép sắp xếp các đối tượng trong cửa sổ theo các mục:

- *By Name: sắp theo tên.*
- *By Type: sắp theo kiểu hay là phần mở rộng của tên tệp.*
- *By Size: sắp theo dung lượng nhớ.*
- *By Date: sắp theo ngày tháng khởi tạo/chỉnh sửa đối tượng.*
- *Auto Arrange: có nghĩa là tự động sắp xếp.*



Các Nhóm lệnh khác

Lệnh Line Up Icons:

- Có ý nghĩa là sắp xếp các biểu tượng trên màn hình nền có hàng có lối

Lệnh Refresh (gọi là làm tươi):

- Có ý nghĩa cập nhật thông tin mới nhất.

Lệnh Paste (gọi là dán):

- Có ý nghĩa sao chép nội dung đã được tạo ảnh bằng lệnh Copy (gọi là sao chép) hay lệnh Cut (gọi là cắt) lên màn hình nền.



Các Nhóm lệnh khác

Lệnh Paste Shortcut (gọi là tạo nút bấm nhanh):

- Có ý nghĩa tạo nút bấm nhanh trên màn hình nền cho nội dung đã được tạo ảnh bằng lệnh Copy (gọi là sao chép) hay lệnh Cut.

Nhóm lệnh New:

← Mực lệnh New



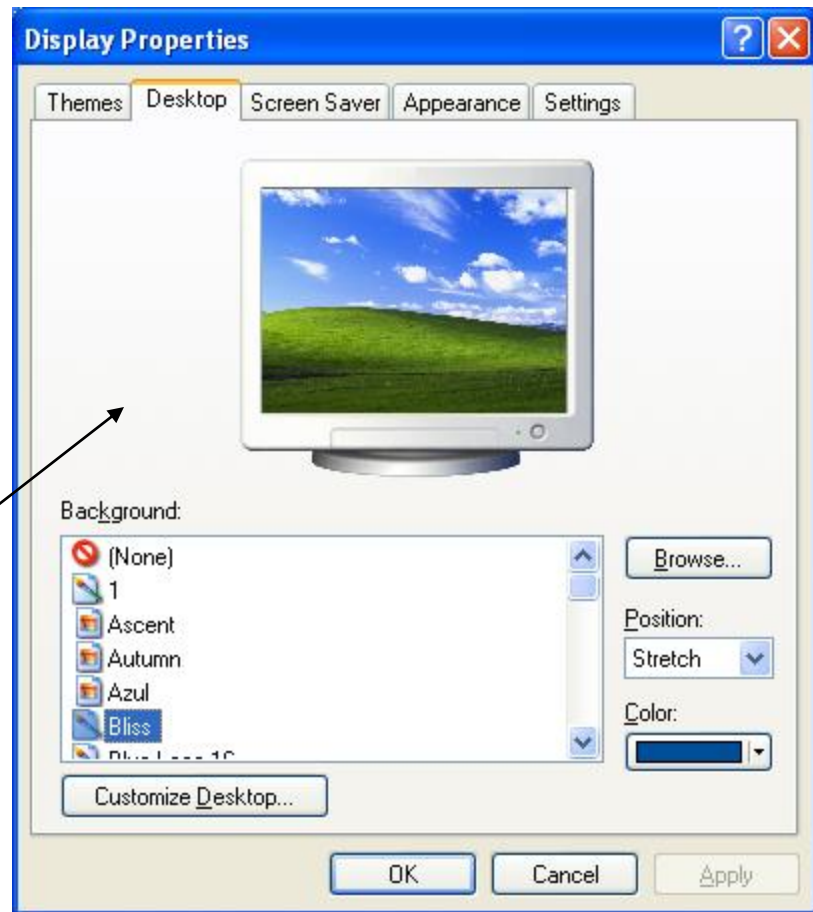
Nhóm lệnh New

- lệnh Folder: cho phép tạo thư mục mới.
- lệnh Shortcut: cho phép tạo nút bấm nhanh trên màn hình nền cho các đối tượng bất kỳ như tệp tin, thư mục,...

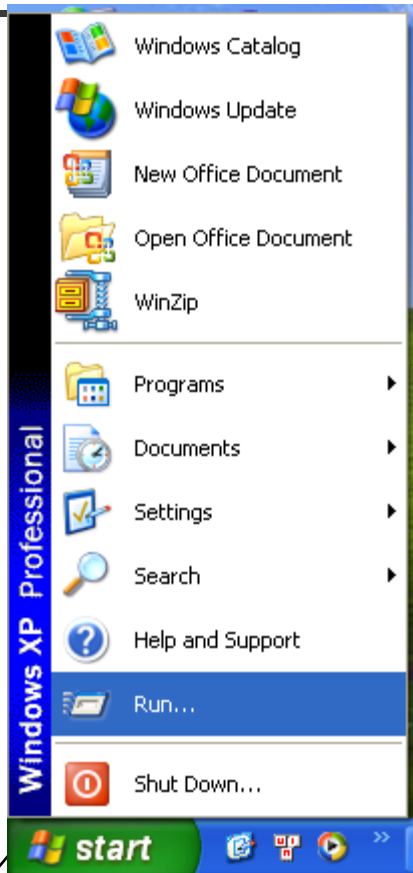
Lệnh Properties

- Cho phép mở hộp thoại để xem và chỉnh sửa các thuộc tính của màn hình nền.

Hộp thoại cho phép chỉnh sửa các thuộc tính của màn hình



Thanh thực đơn Start

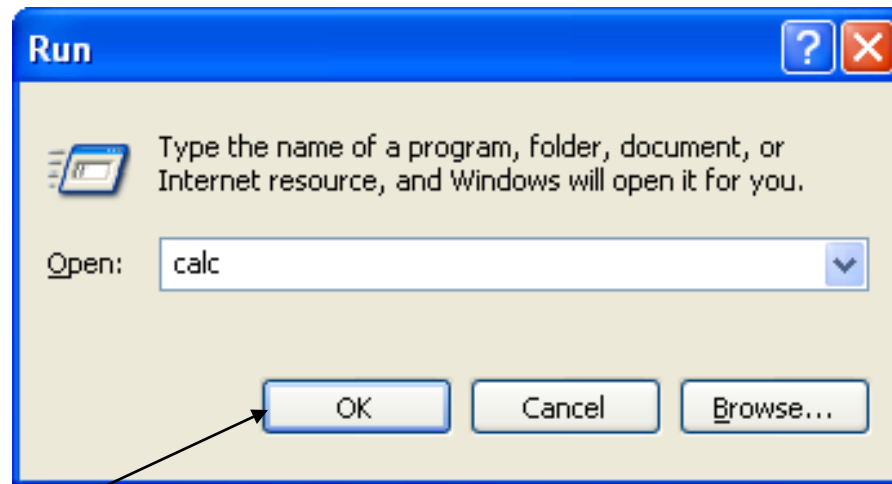


- Nhấp chuột vào nút Start có mặt trên màn hình nền sẽ làm xuất hiện hộp danh mục chọn được phân chia theo chủ đề cho phép người sử dụng dễ dàng ra lệnh cho máy tính.

Nhấp chuột vào nút Start làm xuất hiện hộp danh mục chọn

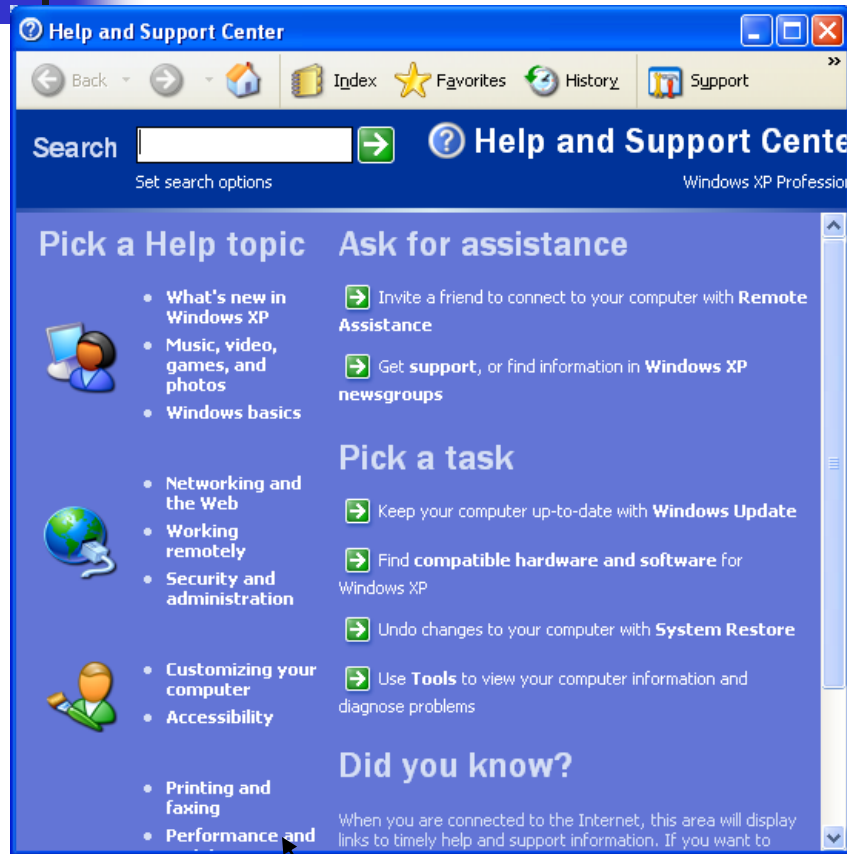
Mục Run - cho phép gõ lệnh

- Nhấp chuột vào nút Start, chọn mục Run làm xuất hiện hộp thoại nhập lệnh. Ví dụ gõ lệnh "calc" để mở bảng tính số học.



Nhập lệnh vào ô nhập Open, sau đó nhấn nút OK

Mục Help - mở phần trợ giúp



Hộp thoại Help

- Phần trợ giúp rất cần thiết khi sử dụng Windows nhưng đòi hỏi người sử dụng phải biết tiếng Anh. Để sử dụng phần Trợ giúp, nhấp chuột vào nút **Start**, sau đó chọn mục **Help and Support** để mở hộp thoại Help.

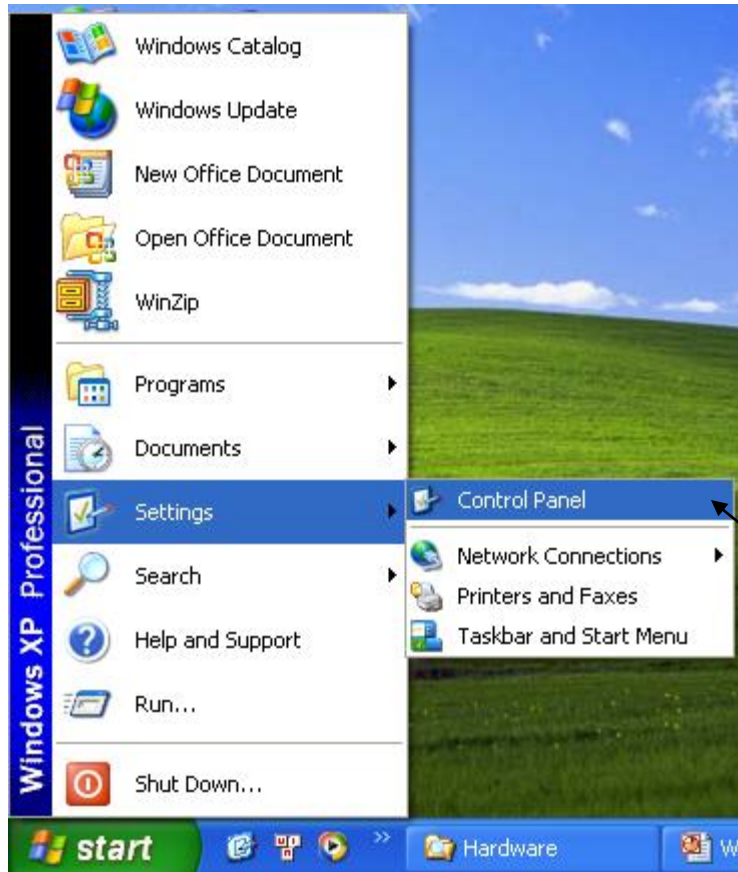
Mục Search - hỗ trợ tìm kiếm thông tin trên máy



- Nhấp chọn nút **Start** → **Search** → **For Files or Folders** làm xuất hiện hộp thoại hỗ trợ tìm kiếm.

Chọn Search – For Files or Folders

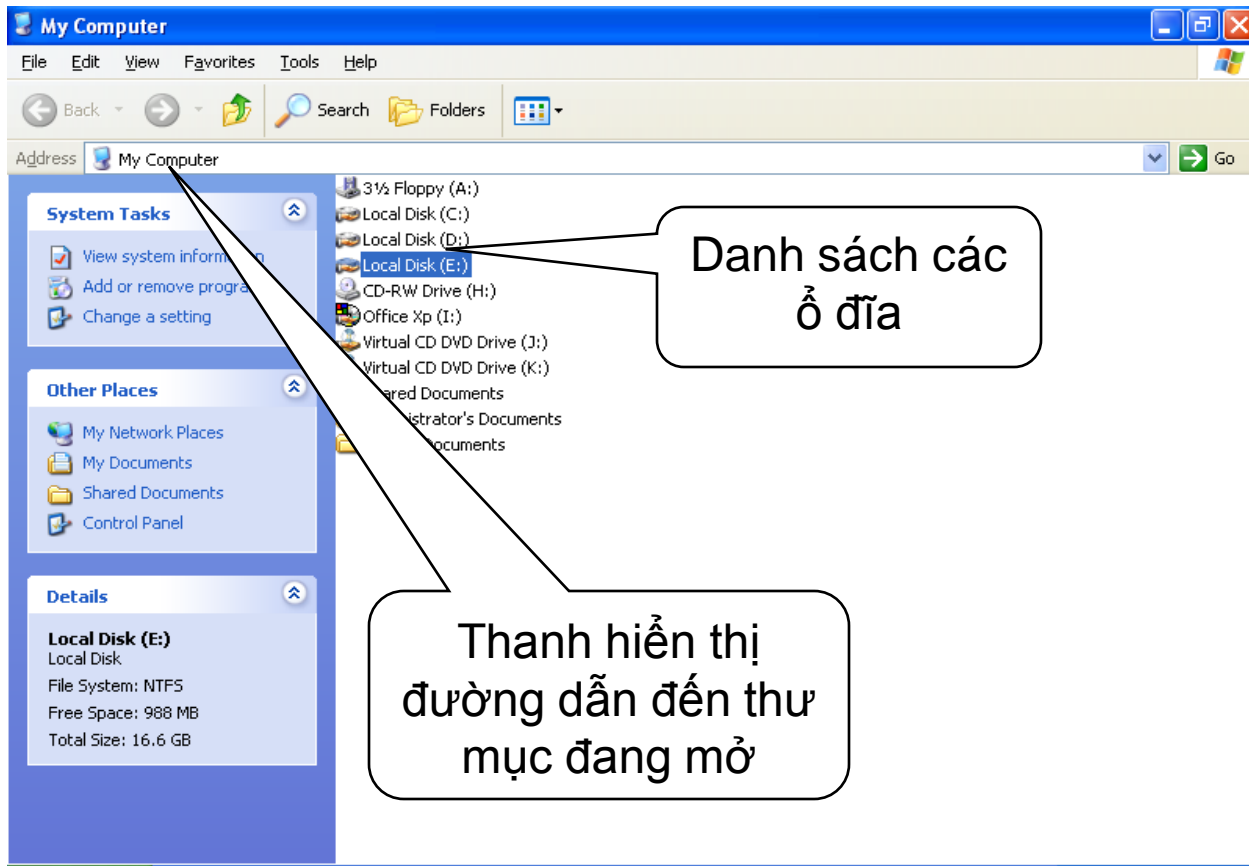
Mục Settings - hỗ trợ quản trị hệ thống



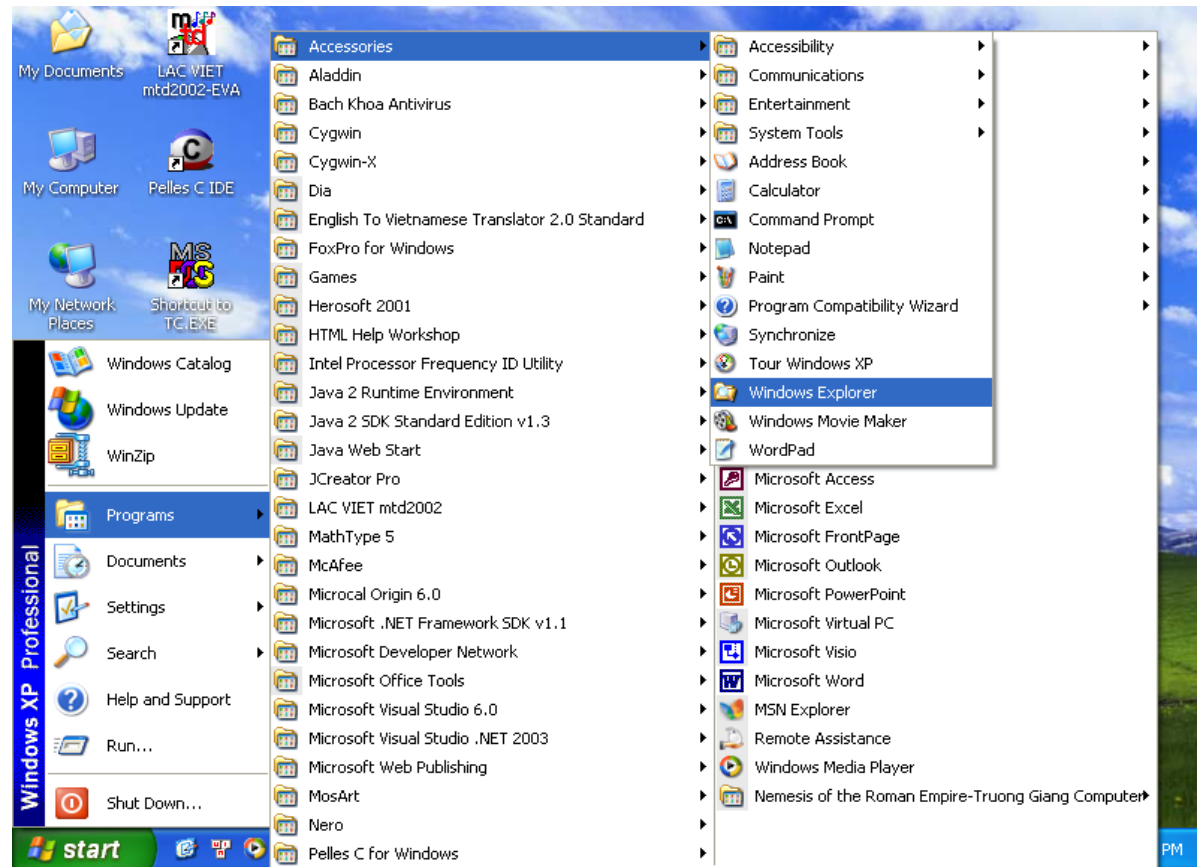
- Nhấp chuột vào nút **Start**
→ **Settings**, sau đó chọn các mục con như Control Panel, Printers...

Chọn Settings-Control Panel

Cửa sổ My Computer

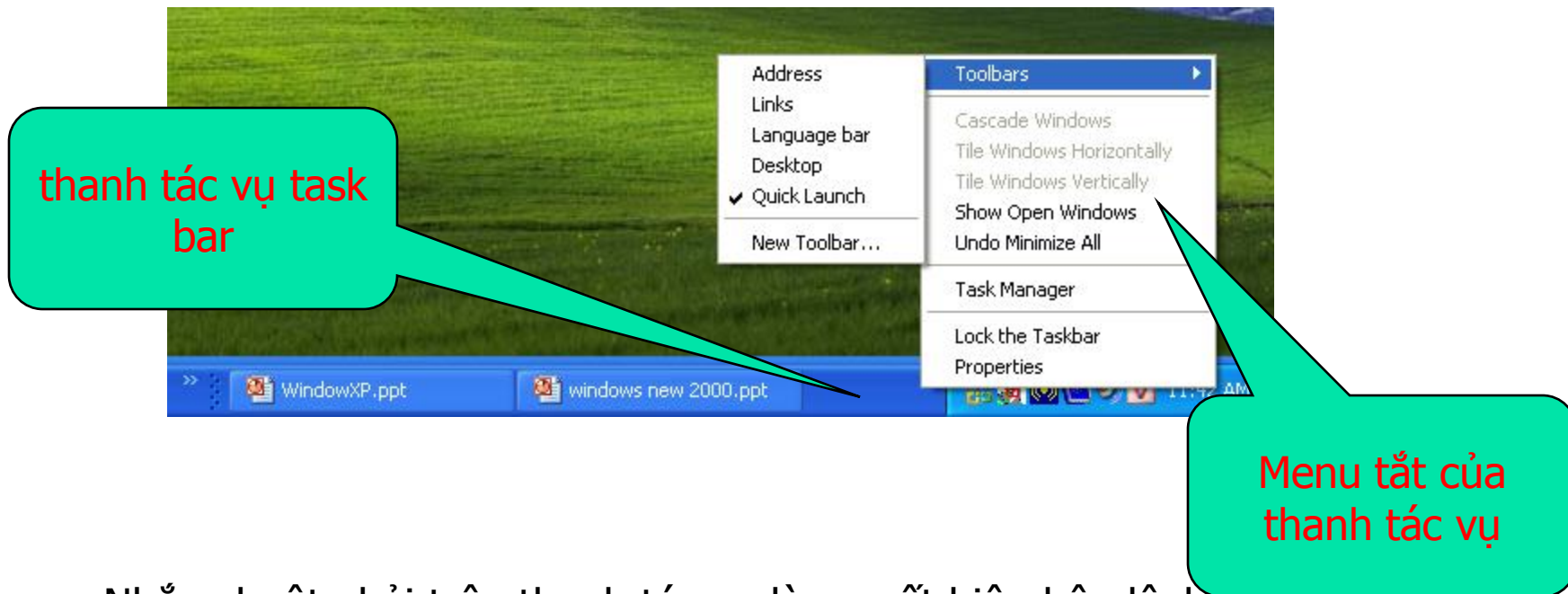


Các chương trình nằm trong mục Programs



Menu tắt của thanh tác vụ

- Nhấp chuột phải trên thanh tác vụ làm xuất hiện hộp lệnh.

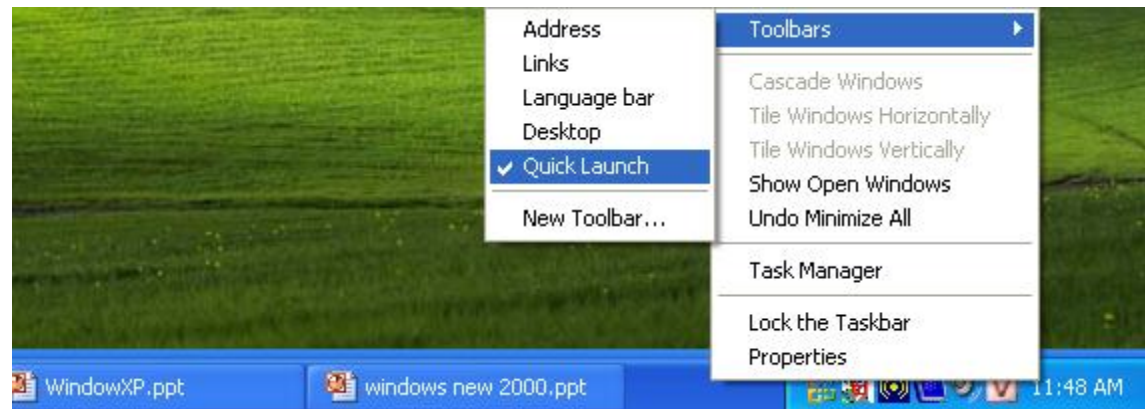


Nhấp chuột phải trên thanh tác vụ làm xuất hiện hộp lệnh

Ý nghĩa các lệnh như sau

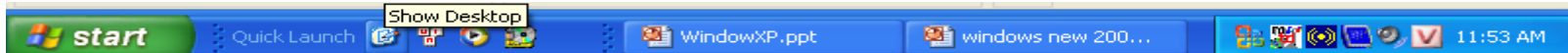
Lệnh Toolbars:

- Trên thanh tác vụ có thể mở nhiều thanh công cụ để làm việc. Có thể mở hoặc đóng bớt các thanh công cụ qua nhóm lệnh Toolbars.

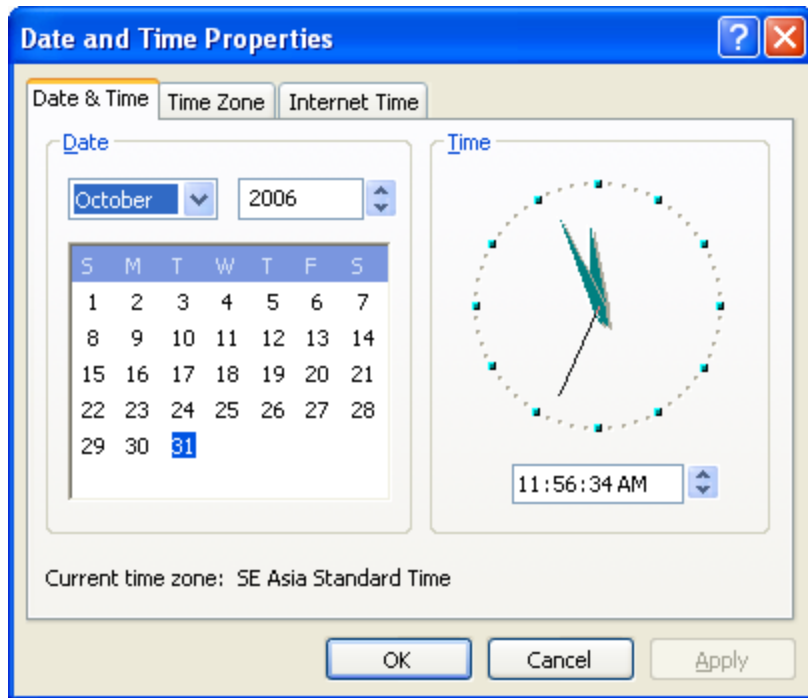


Ý nghĩa các lệnh như sau

- Mặc dù có thể chọn nhiều thanh công cụ đặt trên thanh tác vụ nhưng chúng ta chỉ nên chọn một thanh công cụ đó là thanh Quick Launch.
- Thanh công cụ Quick Launch là một tiện ích không thể bỏ qua đối với người sử dụng. chúng ta nên đặt các nút lối tắt ở đây để tiện sử dụng.
- Trên Quick Launch thường có đặt nút Show Desktop (hình minh hoạ) đây là nút đưa chúng ta nhanh chóng trở về màn hình nền.



Ý nghĩa các lệnh như sau



Lệnh Adjust Date/Time:

- Cho phép mở hộp thoại Date/Time Properties để hiệu chỉnh đồng hồ máy tính.
- Thẻ Date&Time cho phép chỉnh sửa ngày/tháng/năm và giờ. Thẻ Time
- Zone cho phép chỉnh múi giờ đúng theo múi giờ của Việt Nam.

Hộp thoại cho phép chỉnh sửa thời gian hệ thống



Ý nghĩa các lệnh như sau

Lệnh Cascade Windows:

- cho phép sắp xếp các cửa sổ đang mở theo dạng xếp mái ngói.

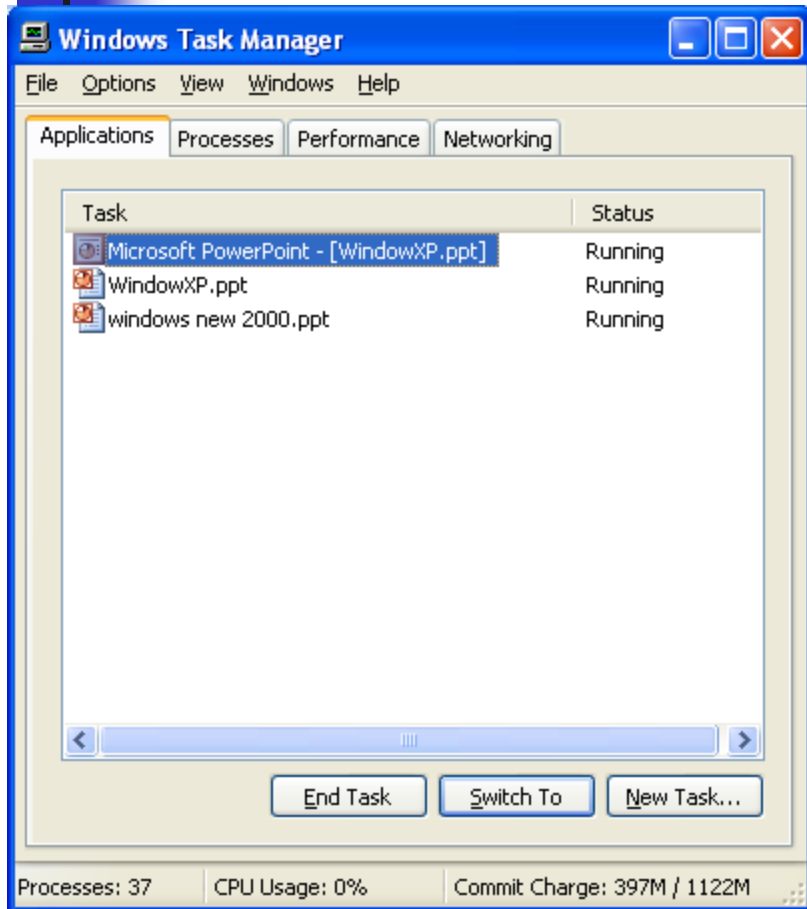
Lệnh Tile Windows Horizontally và Tile Windows Vertically:

- cho phép xếp các cửa sổ dàn ngang trên màn hình nền, không có cửa sổ bị che lấp.

Lệnh Minimize All Windows:

- cho phép thu nhỏ tất cả các cửa sổ đang mở cùng một lần.

Ý nghĩa các lệnh như sau



Lệnh Task Manager:

- cho phép mở cửa sổ quản lý chương trình - Windows Task Manager.
- Trong thẻ Applications của cửa sổ này, người sử dụng có thể chọn một chương trình hay nhiều chương trình và ra lệnh đóng chương trình bằng cách nhấn nút End Task. Đây là chức năng rất tiện ích cho việc đóng những chương trình đang gây tắc nghẽn hệ thống.

Hộp thoại Windows Task Manager

Khởi động lại và tắt máy tính

Để tắt máy đúng cách ta làm theo trình tự sau:

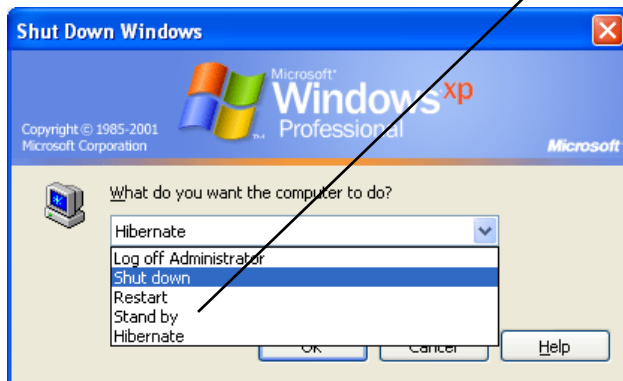
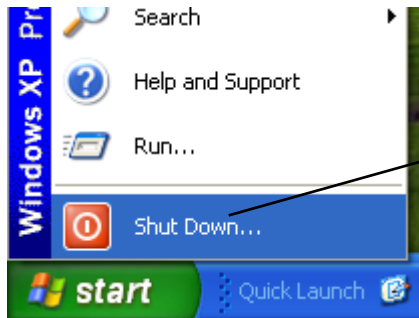
1. Đóng bất kỳ chương trình nào đang mở, lưu tất cả các tài liệu nếu cần thiết

2. Bấm chuột vào nút **Start** \ chọn **Shut Down...**

3. Chọn **Shut Down** (nếu muốn tắt máy)
Restart (nếu bạn muốn khởi động lại máy)
Log off Username (nếu bạn muốn thoát khỏi phiên hiện hành và để đăng nhập vào hệ thống với một tên Username khác mà không muốn khởi động lại máy)
Stand By (nếu bạn muốn cho máy ở chế độ ngủ đông)

4. Bấm **OK** để chấp nhận

5.. **Tắt nguồn điện** dẫn vào màn hình và máy tính.



Tắt máy tính theo kiểu áp đặt



- Tắt nguồn điện bằng cách bấm nút **POWER** trên hộp máy (có thể phải giữ tay trên nút khoảng 30 giây) hoặc
- Nhấn nút **Reset** trên hộp máy.

Các bộ phận chính của một máy PC



Khởi động lại máy

Trong trường hợp máy tính không còn điều khiển được bằng bàn phím và chuột thì:

- Nhấn tổ hợp phím **Ctrl + Alt + Del**, hoặc từ cửa sổ màn hình nền, nhấn chọn nút **Start → Shutdown** làm xuất hiện hộp thoại **Shut Down Windows**. Trong ô chọn, chọn mục **Restart**.



Tóm tắt : Học viên cần nắm

1. Tắt và khởi động máy đúng cách.
2. Nhận diện và Sử dụng các công cụ trên màn hình Desktop, sắp xếp các đối tượng.
3. Các thao tác khi sử dụng chuột.

Bài 3: Làm quen với tập tin và thư mục



Mục tiêu bài học:

- Học viên nắm vững khái niệm về tập tin và thư mục, cách tổ chức và quản lý chúng trong Hệ điều hành Windows.
- Thành thạo việc chọn nhóm biểu tượng liên kề và rời rạc.
- Thực hiện các thao tác cơ bản với tập tin và thư mục trên màn hình nền như: tạo, chỉnh sửa, mở nội dung, di chuyển, và xoá.



Ổ đĩa vật lý và ổ đĩa Logic

- Các ổ đĩa cứng có dung lượng nhớ rất lớn nên hệ điều hành có chức năng chia nhỏ ổ đĩa cứng thành các ổ đĩa gọi là ổ đĩa cứng logic để người sử dụng có thể tiện sử dụng.
- Các ổ đĩa được HĐH gán bằng các chữ cái như sau:
 - A:, B: Đĩa mềm
 - C:, D:, ... Đĩa cứng
 - E:, F:, ... Đĩa CDROM



Tập tin (File)

- Trong hệ điều hành Windows, tệp tin là đối tượng chứa dữ liệu. Ví dụ các văn bản sau khi nhập vào máy được lưu thành các tệp tin để sau đó có mở ra xem lại, chỉnh sửa/in ấn và có thể xoá đi.
- Cấu trúc của tệp tin gồm 2 phần:
- **Tên tệp tin.*phần mở rộng***
VD: baitho.*doc*



Đặc trưng của tập tin

- Độ dài của tên tập tin không quá 8 ký tự (đối với DOS) và không quá 250 ký tự (đối với Windows 98 trở lên).
- Kiểu tập tin phụ thuộc vào phần mở rộng của tập tin (VD: **baitap.doc** – có phần mở rộng là **.doc** cho nên đây là tập tin **word**).



Một số kiểu tập tin thông dụng

- **.doc, .txt, .rtf**: Các tập tin văn bản Word
- **.xls** : Các tập tin bảng tính Excel
- **.exe, .bat**: Các tập tin chương trình
- **.com** : tập tin lệnh
- **.gif, .jpeg, .bmp**: Các tập tin chứa hình ảnh
- **.mp3, .dat, . Wav**: Các tập tin âm thanh, video
- **html, htm**: Các tập tin siêu văn bản
- **sql, mdb**: Các tập tin chứa cơ sở dữ liệu

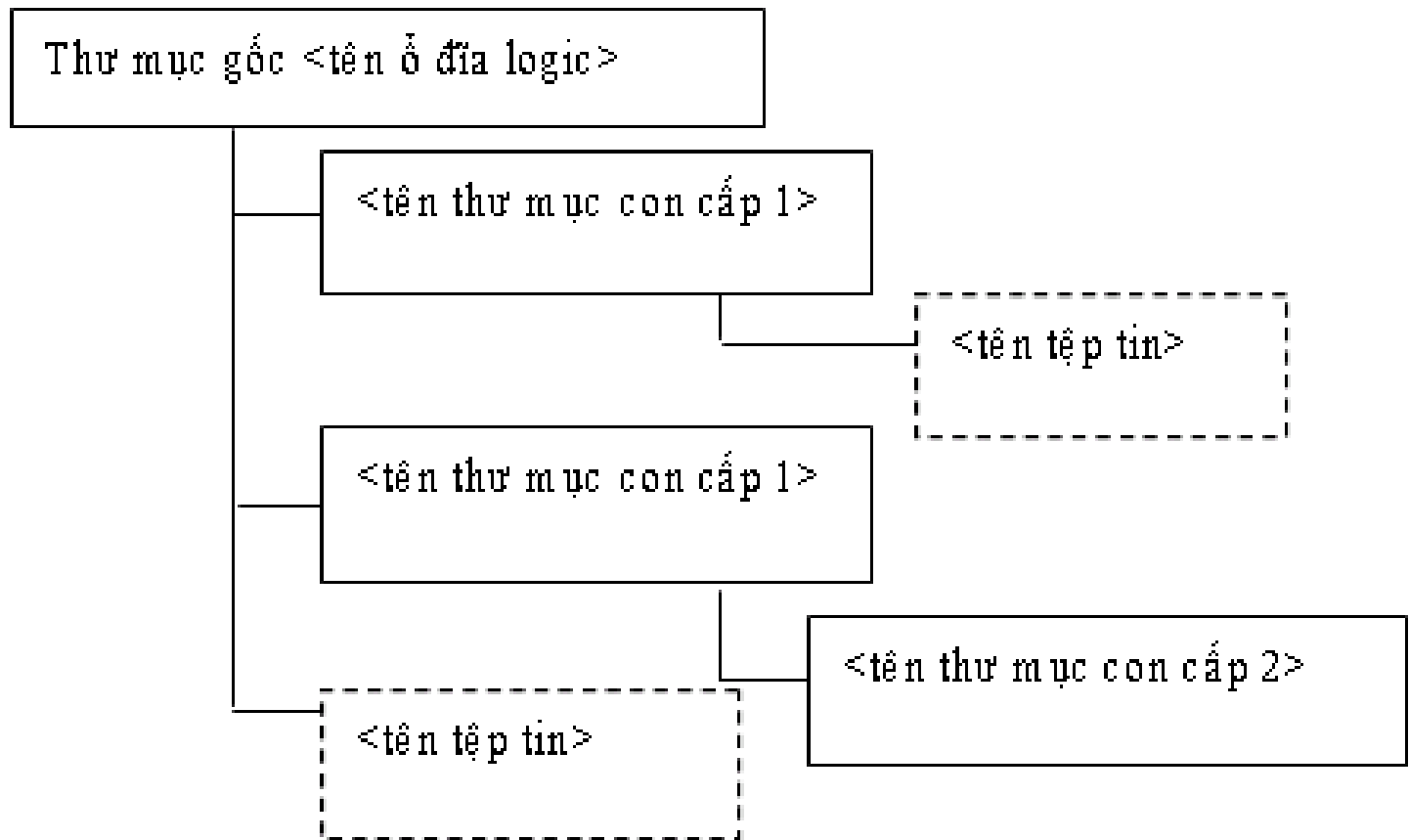
.....



Thư mục - Folder

- Để lưu giữ, sắp xếp các tập tin thành một hệ thống phân cấp có tính chặt chẽ và tiện dụng khi tìm kiếm, hệ điều hành Windows cho phép người sử dụng xây dựng cây thư mục theo cách thức:
 - Các đặc trưng của thư mục:
 - Ổ đĩa logic của máy tính được xác định là thư mục gốc
 - Có thể tạo nhiều thư mục con trong thư mục
 - Các thư mục cùng cấp không được trùng tên
 - Tập tin phải được chứa trong một thư mục

Thư mục - Folder



Hệ thống cây thư mục của HĐH Windows



Đường dẫn cho tập tin

- Để diễn tả vị trí của tập tin trong hệ thống thư mục chúng ta cần viết đường dẫn theo cách sau:
- [tên qui ước đĩa logic:] [\] [<tên thư mục> \ ... \ <tên thư mục> \ <tên tập tin>]

VD: Đường dẫn *C:|congvan2004|danhsachCB1.doc*

- Chỉ ra tập tin *danhsachCB1.doc* đang được chứa trong thư mục *congvan2004* thuộc đĩa C.

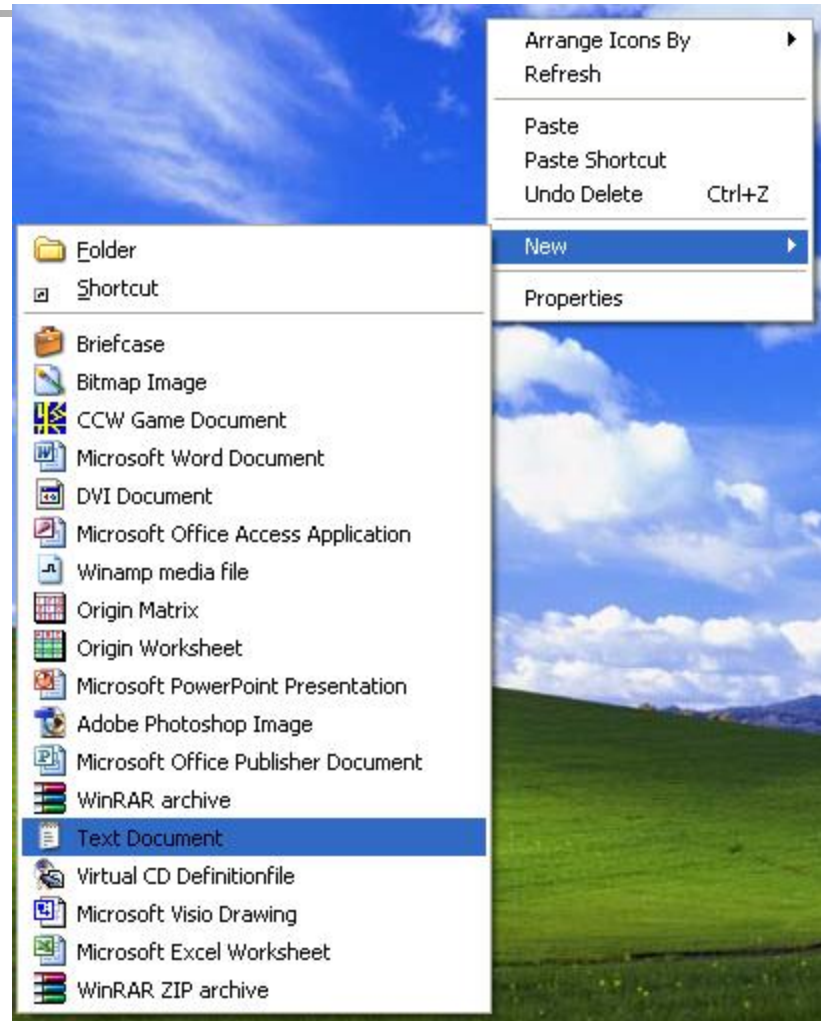


Tạo mới một tập tin

- Nhấp chuột phải trên màn hình nền làm xuất hiện hộp lệnh.
- **Lựa chọn loại tập tin muốn tạo** (Vd: Chọn mục **New** → **Text Document** (Tập tin được tạo ra chỉ là một tập tin rỗng chưa có nội dung gì bên trong).
- hoặc **Microsoft Word Document** - Tập tin văn bản, **Microsoft Excel Worksheet** - bảng tính Excel,...).

Tạo mới một tập tin

- Minh họa các mục chọn trên hộp lệnh để tạo tập tin kiểu Text



Tạo mới một thư mục



Ba thư mục mới và hai
tệp tin được tạo

- Nhấp **chuột phải** trên màn hình nền làm xuất hiện hộp lệnh.

- Chọn mục **New**
→ **Folder**.

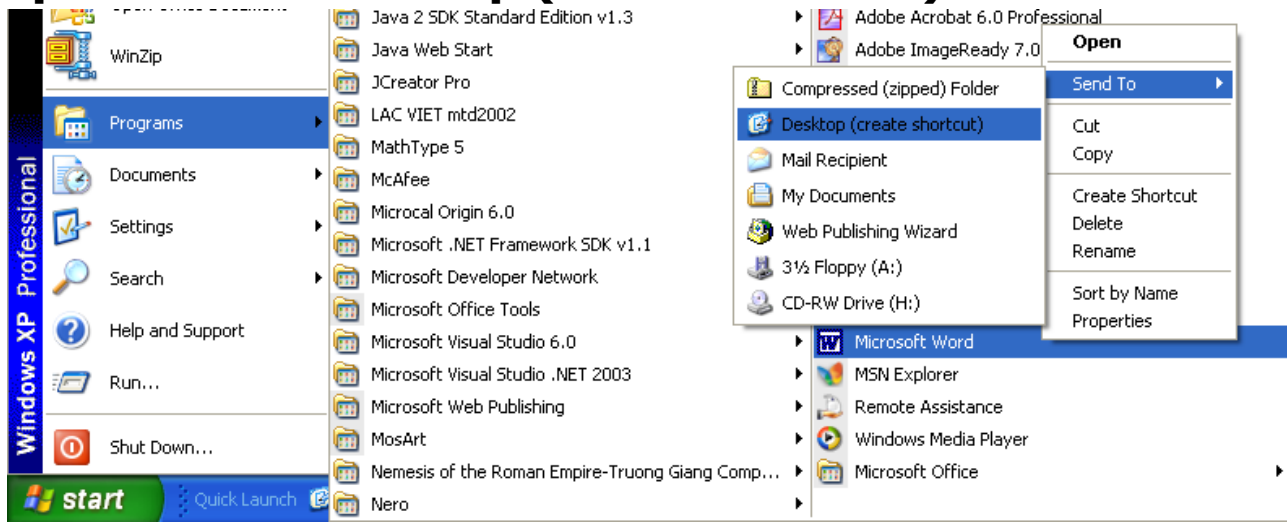
- **Đặt tên thư mục** mà bạn vừa mới tạo (thư mục mới có tên mặc định là **New Folder**)

Tạo biểu tượng lối tắt

Shortcut (lệnh tắt): Là một biểu tượng lối tắt được đặt trên **desktop** cho phép bạn chạy các ứng dụng một cách dễ dàng hơn. (*tạo 1 lần và sử dụng nhiều lần*).

Tạo một Shortcut:

- Nhấp nút Start, chọn mục Program → Microsoft Word.
- Nhấp chuột phải tại mục Calculator để mở hộp lệnh. Chọn mục SendTo → Desktop (create shortcut).



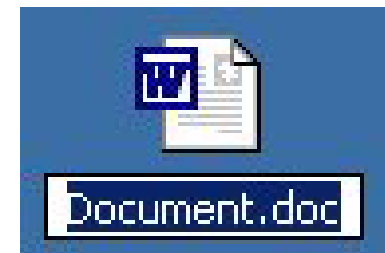
Các mục đậm màu thể hiện các vị trí cần di chuyển chuột đến

Đổi tên của biểu tượng

- Nhấp chuột phải lên biểu tượng làm xuất hiện hộp lệnh.
- Chọn mục **Rename**. Sau đó gõ tên mới vào ô nhập.
- Hoặc Chỉ cần nhấp chuột một lần vào phần tên của biểu tượng lập tức hệ điều hành cho phép gõ tên mới như hình minh họa.



Chọn lệnh Rename để đổi tên



Biểu tượng ở trạng thái cho phép nhập tên mới



Thi hành một ứng dụng:

- **Bằng biểu tượng** : **Double click** vào biểu tượng trên màn hình **Desktop**.
- **Bằng Menu Start**: Click chuột vào nút **Start\Program** click vào ứng dụng cần khởi động.
- **Bằng Mycomputer**: Nhấp đúp vào biểu tượng **Mycomputer**, lần theo đường dẫn chứa tập tin muốn thi hành.
- **Bằng lệnh Run**: Click chuột vào nút **Start \ Run**, một hộp thoại xuất hiện, gõ tên chương trình cần chạy (Vd: **Excel**) vào hộp **Open** rồi nhấn **OK**.

Thay đổi cách hiển thị của các đối tượng

Click chuột phải vào menu **View** của cửa sổ và lựa chọn các mục như hình, lựa chọn 1 trong bốn mục dưới đây để hiển thị:

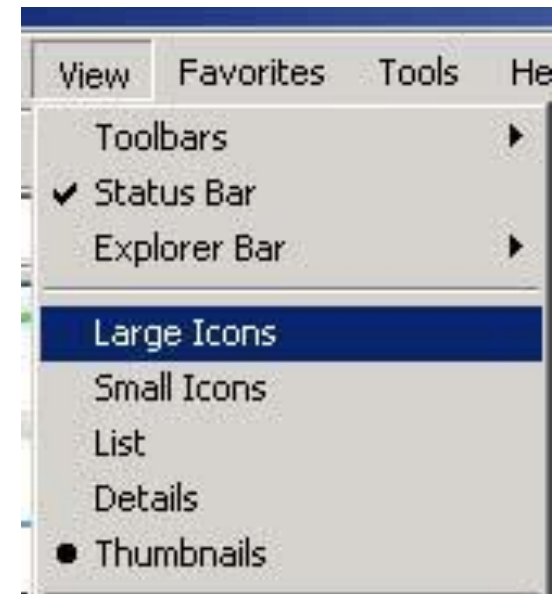
Large Icons: Biểu tượng lớn.

Small Icons: Biểu tượng nhỏ.

List: Kiểu liệt kê.

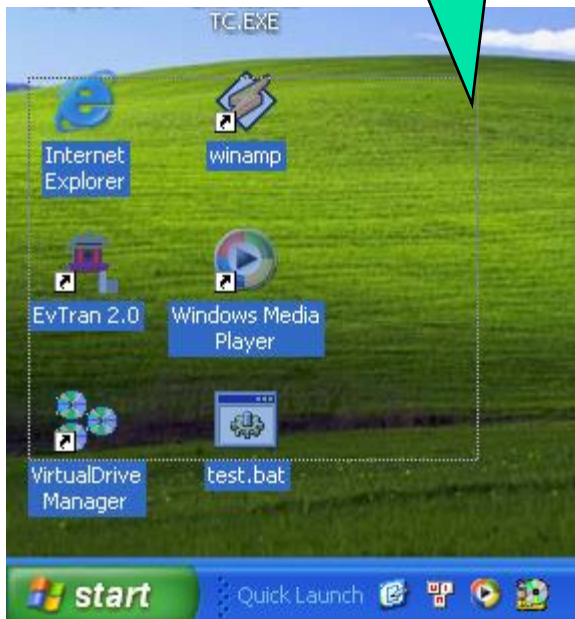
Detail: Kiểu nhìn chi tiết.

Thumbnails: Hiển thị nhanh các tập tin ảnh.



Chọn nhóm tập tin, thư mục

xuất hiện đường
bao hình chữ
nhật



Chọn nhóm đối tượng liên kề

Thao tác bằng thiết bị chuột

- Kéo di chuột tạo một đường hình chữ nhật bao quanh các biểu tượng muốn chọn. Các biểu tượng được chọn đổi sang màu tối nên thao tác chọn được gọi nôm na là "bôi đen đối tượng".

Chọn nhóm tập tin, thư mục

Chọn nhóm biểu tượng rời rạc Thao tác kết hợp giữa chuột và bàn phím

- Nhấp chuột chọn một biểu tượng.
- Giữ phím Ctrl và nhấp chuột vào biểu tượng khác.
- (Điểm quan trọng ở đây là giữ phím Ctrl khi chọn các đối tượng. Có thể nhấp chọn lần thứ hai trên một đối tượng để nhanh chóng hủy chọn chỉ riêng cho đối tượng đó).





Chọn nhóm tập tin, thư mục

Chọn các tập tin /thư mục không liên tục: *Bấm phím **Ctrl** và Click chọn tên thư mục cần chọn.*

Chọn các tập tin /thư mục liên tục: *Bấm phím **Shift** và Click chọn chuột vào đầu và cuối của khối thư mục mà mình muốn chọn.*

Hủy chọn: *Để hủy chọn toàn bộ các đối tượng đã chọn , ta nhấp chuột tại vị trí trống bất kỳ.*

Chuyển về thư mục trên một cấp: *Click nút **Back** hoặc nút **Up**.*



Đổi tên tập tin, thư mục:

Có 3 cách:

Chọn tên tập tin/thư mục cần đổi tên.

*Cách 1: Chọn menu lệnh **File\ Rename** → nhập tên mới.*

*Cách 2: Click phải chuột → **Rename***

*Cách 3: Nhấn phím **F2***



Sao chép, di chuyển tập tin, thư mục

Chọn các tập tin / thư mục cần sao chép (di chuyển), làm theo một trong 3 cách sau:

Cách 1: vào menu **Edit\Copy** (hoặc **Cut** nếu muốn di chuyển)

Cách 2: nhấn **Ctrl + C** (hoặc **Ctrl + X** nếu muốn di chuyển)

Cách 3: **Ctrl + kéo chuột** (hoặc **Shift + kéo chuột** nếu muốn di chuyển)

Mở thư mục cần sao chép đến (thư mục đích) rồi chọn **Edit\Paste (hoặc nhấn **Ctrl + V**)**

Sao chép, di chuyển tập tin, thư mục

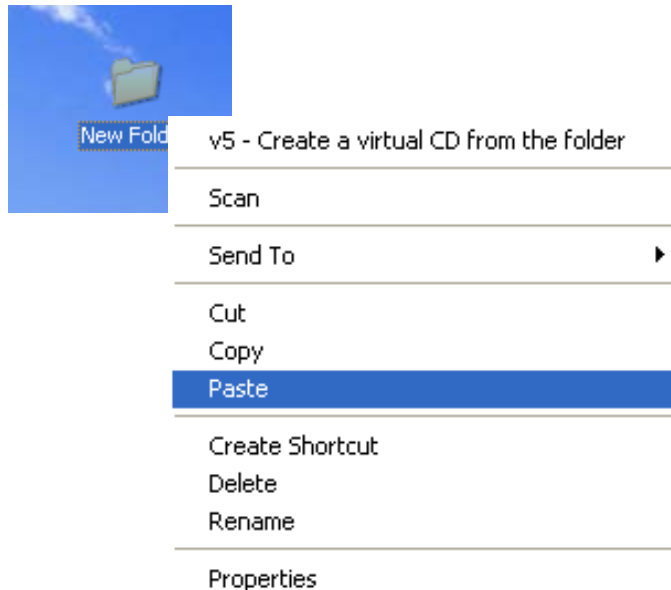


*Chuẩn bị: tạo sẵn trên màn hình nền thư mục có tên là **congvan** và tập tin **Image1**.*

- Chọn nhóm biểu tượng gồm thư mục **congvan** và tập tin **Image1**.
- Nhấp chuột phải trên vùng chọn làm xuất hiện hộp lệnh. Sau đó chọn mục Copy.

Nhấp chuột phải làm xuất hiện hộp lệnh, sau đó chọn mục Copy

Sao chép, di chuyển tập tin, thư mục



Chọn mục Paste

- Tạo mới thư mục tên là New Folder. Nhấp chuột phải trên thư mục này làm xuất hiện hộp lệnh và sau đó chọn mục Paste để dán bản sao của **congvan** và **Image1** vào trong thư mục này.

Gợi ý thêm: Sử dụng tổ hợp phím để thao tác nhanh

- Bấm tổ hợp phím Ctrl+C tương đương với việc chọn mục Copy trong hộp lệnh.
- Bấm tổ hợp phím Ctrl+V tương đương với việc chọn mục Paste trong hộp lệnh.

Di chuyển tệp tin đến thư mục khác

Chọn biểu tượng tệp tin.

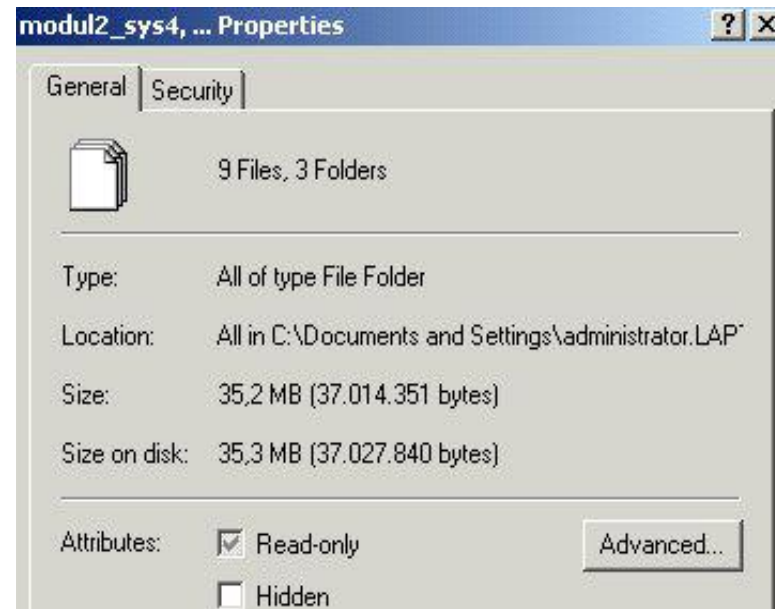
- Nhấp chuột phải lên vùng chọn để làm xuất hiện hộp lệnh. Sau đó chọn mục Cut.
- Nhấp nút phải lên thư mục, nơi sẽ cất giữ tệp tin, để làm xuất hiện hộp lệnh. Tiếp theo là chọn mục Paste.

Gợi ý thêm: Sử dụng tổ hợp phím để thao tác nhanh

- Bấm tổ hợp phím Ctrl+X tương đương với chọn mục Cut trong hộp lệnh.

Hiển thị thông tin của tập tin, thư mục

- Chọn nhóm biểu tượng.
- Nhấp chuột phải trên vùng chọn làm xuất hiện hộp lệnh, chọn mục Properties.



Chọn mục thuộc tính

Thông tin của thư mục xuất hiện



Hiển thị thông tin của tập tin, thư mục

Thông tin của tập tin, thư mục xuất hiện trong hộp Properties cho người dùng biết số lượng tập tin, thư mục và quan trọng nhất là tổng dung lượng nhớ (ví dụ theo hình minh họa là 35.2MB)

Đặt thuộc tính chỉ đọc (read-only)

Trong hộp thoại Properties, nhấp chọn ô Read-only để đặt thuộc tính chỉ đọc cho tập tin hay thư mục và có thể nhấp chọn ô Hidden để đặt thuộc tính che dấu cho tập tin hay thư mục.

chọn cho phép đặt thuộc tính chỉ đọc



Xoá các tập tin, thư mục

Chọn các tập tin / thư mục cần xoá, làm theo các cách sau:

Cách 1: Vào menu **File/Delete**

Cách 2: Ấn phím **Delete**

Cách 3: Click phải → chọn **Delete**

→ Nhấn **Yes** để chấp nhận xoá (nếu không muốn xoá nhấn **No**)

Lưu ý: Sau khi xoá tập tin/ thư mục sẽ được Windows bỏ vào thùng rác. Nếu muốn xoá vĩnh viễn, ta cũng làm như thao tác trên nhưng nhấn giữ thêm phím **Shift** trong khi chọn **Delete.**(tổ hợp phím **Shift-Del**)

Xoá các tệp tin, thư mục

- Biểu tượng của thùng rác trên màn hình nền



thùng rác rỗng



thùng rác chứa tệp tin hoặc thư
mục đã bị xóa

Lưu ý: Nếu dùng tổ hợp phím Shift-Del để thực hiện xóa tệp tin, thư mục thì đối tượng bị xóa sẽ mất hẳn không lưu lại trong thùng rác.



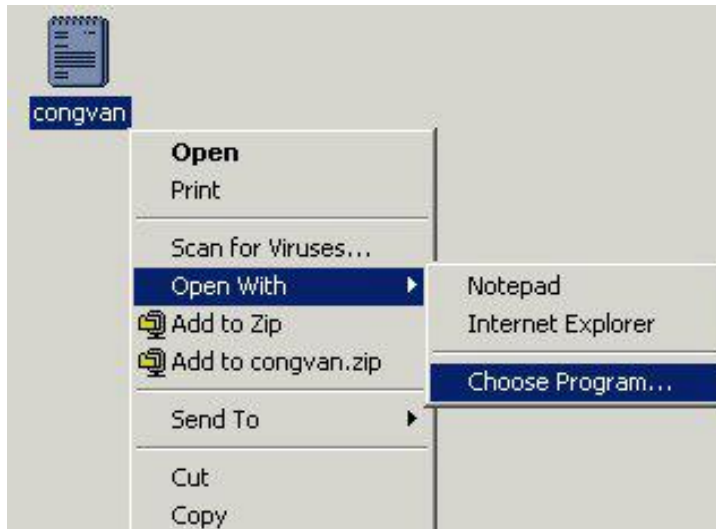
Khôi phục tập tin, thư mục đã xoá

Double Click vào biểu tượng **Recycle Bin** ở trong **desktop** → Chọn **tập tin/ thư mục** cần khôi phục:

- **Cách 1:** Vào menu **File \ Restore**
- **Cách 2:** **Click phải \ Restore**

Mở tệp tin

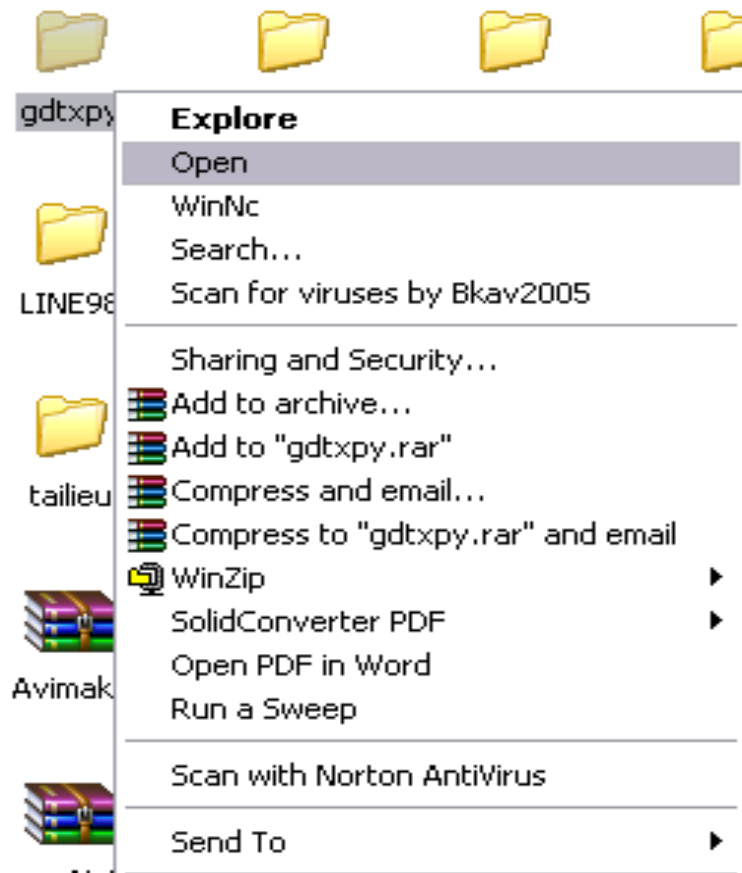
- Nhấp đúp lên biểu tượng tệp tin hoặc
- Nhấp chuột phải trên biểu tượng tệp tin làm xuất hiện hộp lệnh, khi đó có hai lệnh để chọn: lệnh Open hoặc lệnh Open with
- Chọn một chương trình mở tệp và nhấp nút OK.



chọn mục Open With để mở tệp bằng chương trình tự chọn

Chọn lựa chương trình mở tệp là Microsoft Word

Mở thư mục



chọn mục Open để mở thư mục

Có hai cách để mở thư mục:

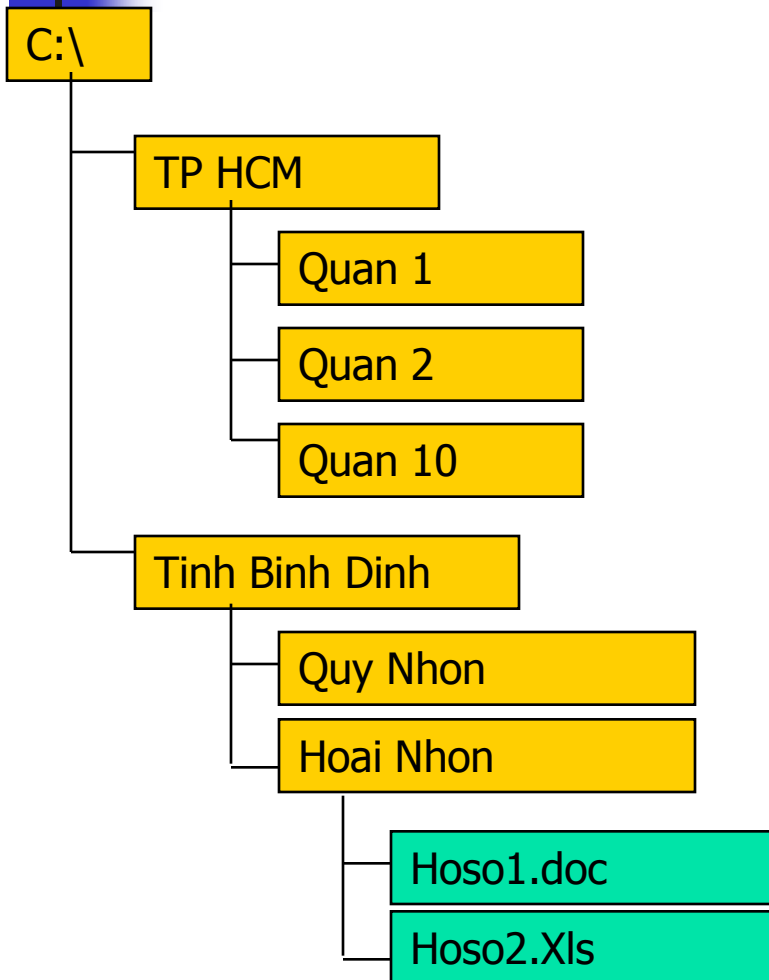
- nhấp đúp lên biểu tượng thư mục hoặc
- nhấp chuột phải trên biểu tượng thư mục làm xuất hiện hộp lệnh, sau đó chọn mục Open.



Tóm tắt bài 3

- *Phân biệt được tập tin, thư mục, các kiểu tập tin, hiểu khái niệm đường dẫn.*
- *Tạo mới tập tin, thư mục.*
- *Đổi tên tập tin, thư mục.*
- *Copy, di chuyển tập tin, thư mục.*
- *Hiển thị thông tin, thay đổi cách hiển thị của tập tin, thư mục.*
- *Xóa, phục hồi tập tin, thư mục.*

BÀI TẬP



Câu 1: Tạo cây thư mục như hình bên.

Câu 2: Sao chép 2 tập tin hoso1.doc và hoso2.xls vào thư mục Quan 2.

Câu 3: Xoá thư mục Hoai Nhon

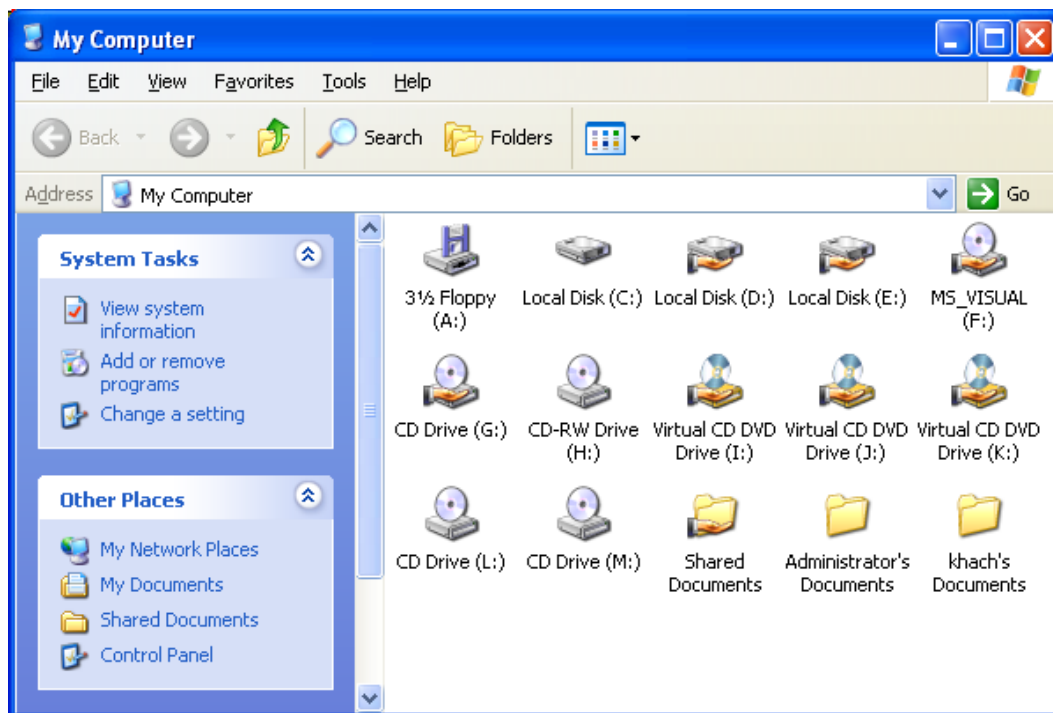
Câu 4: Khôi phục lại thư mục vừa xoá



Bài 4: LÀM VIỆC VỚI CỬA SỔ

Các thao tác cơ bản

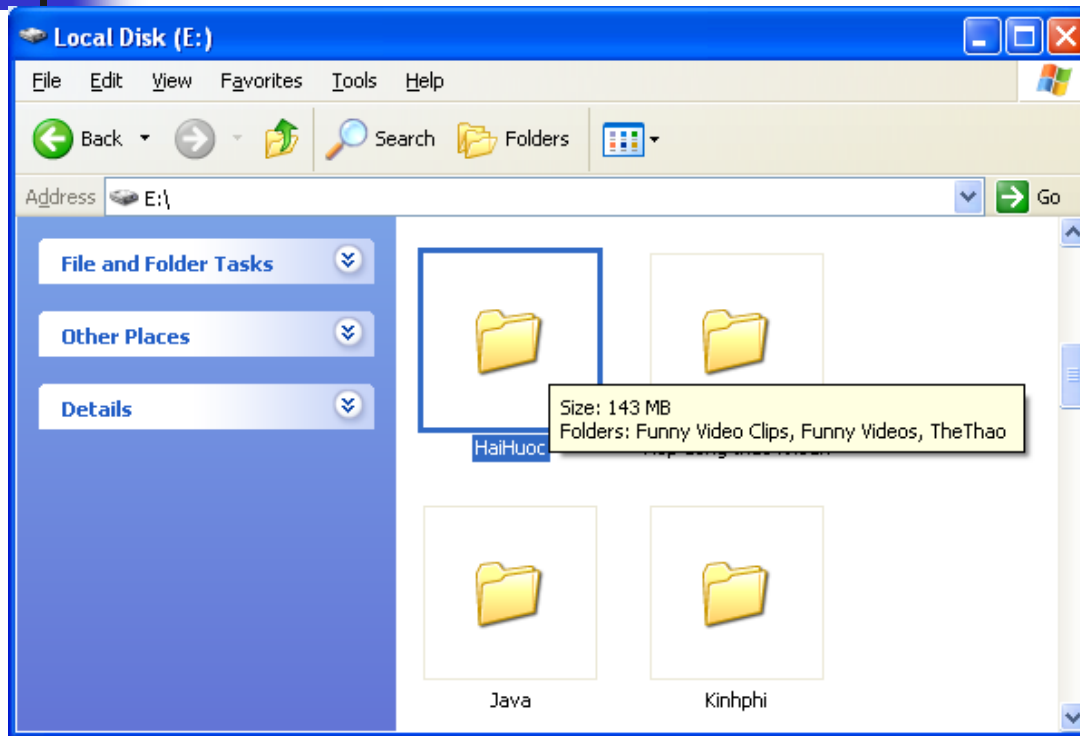
Mở cửa sổ :



Cửa sổ My Computer

- **Bằng My computer:** Nhấp đúp chuột lên biểu tượng My computer.
- **Bằng biểu tượng thư mục, tập tin:** Nhấp đúp lên biểu tượng **thư mục, tập tin** cần mở.

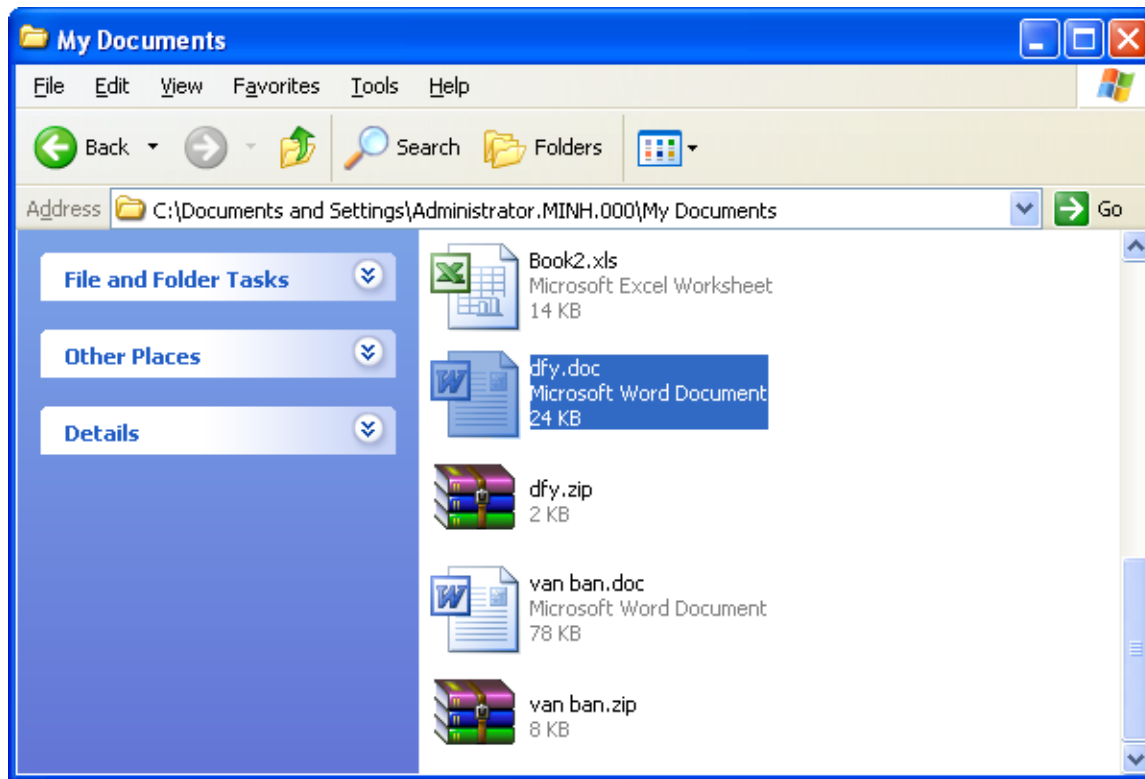
Cửa sổ làm việc với thư mục



Mở cửa sổ thư mục


- Thao tác nhấp đúp chuột lên biểu tượng thư mục bất kỳ có trên màn hình để mở cửa sổ làm việc với thư mục.

Cửa sổ làm việc với tệp tin

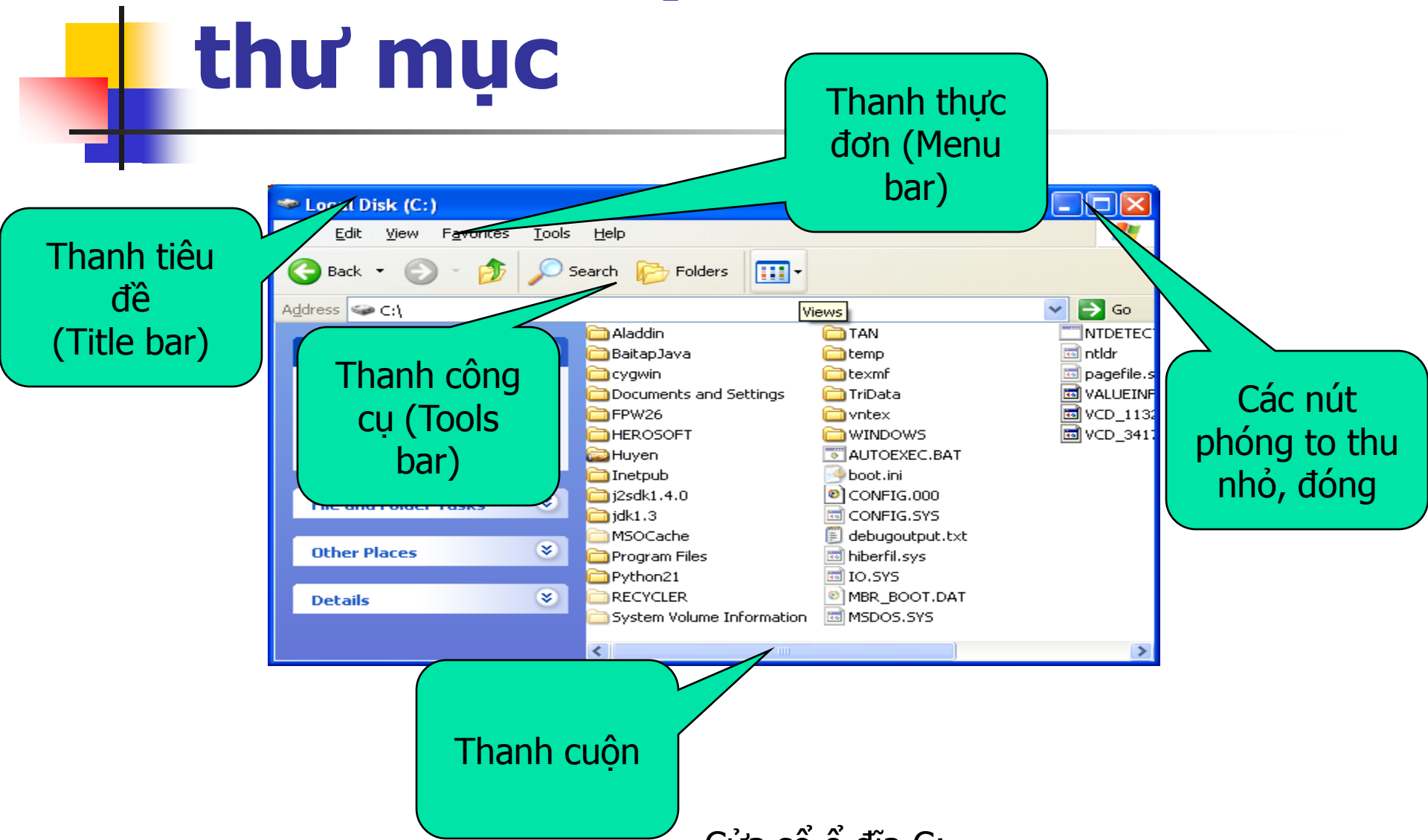


Thao tác nhấp đúp chuột lên biểu tượng tệp tin bất kỳ có trên màn hình nền để mở cửa sổ làm việc với tệp tin.

Các thành phần của cửa sổ thư mục

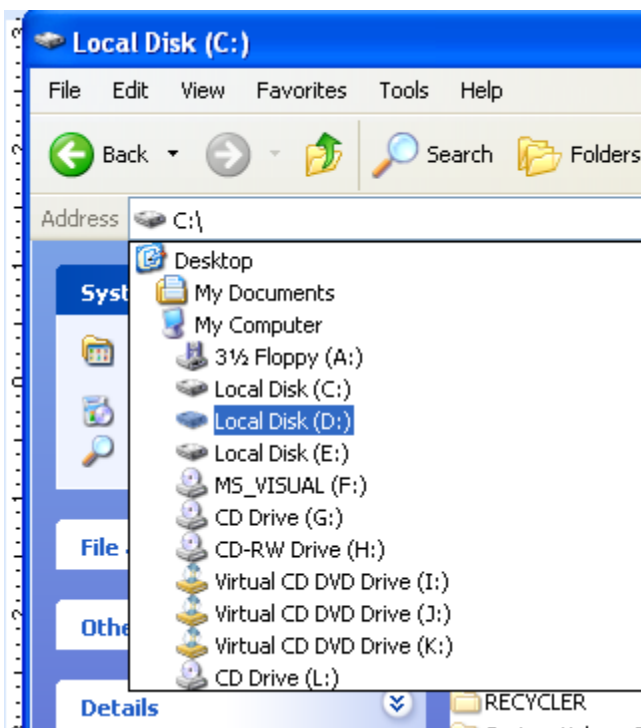
- Thanh tiêu đề nằm phía trên cửa sổ. Khi nhấp đúp chuột trên thanh tiêu đề làm phóng to hoặc thu nhỏ cửa sổ.
- Dưới thanh tiêu đề là thanh thực đơn lệnh gồm thực đơn lệnh FILE, EDIT,... Tất cả các lệnh để điều khiển cửa sổ và để điều khiển các đối tượng khác trong cửa sổ đều có mặt trong thanh thực đơn lệnh.
- Thanh công cụ chứa các nút gắn sẵn chức năng (ví dụ nút  để xoá tệp tin hay thư mục đã chọn) giúp cho người sử dụng thao tác dễ dàng hơn, chính xác hơn và nhanh hơn so với việc chọn các lệnh có trên thanh thực đơn.

Các thành phần của cửa sổ thư mục



Cửa sổ ổ đĩa C:

Thanh công cụ địa chỉ Address



Hộp chọn hỗ trợ khả năng hiển thị sơ đồ thư mục dạng cây

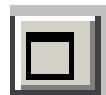
- Thanh công cụ địa chỉ (tiếng Anh là Address) có hộp chọn hỗ trợ khả năng hiển thị thư mục theo sơ đồ dạng cây giúp người dùng dễ hình dung cách tổ chức thư mục trên bộ nhớ ngoài mỗi khi làm thao tác chọn và mở các cửa sổ tiếp theo.

Thu nhỏ, phóng to, đóng

Tại góc trên, bên phải cửa sổ có các nút:



- là nút thu nhỏ cửa sổ. Chỉ còn một nút bấm nhanh trên thanh tác vụ cho phép mở lại cửa sổ;



- là nút chức năng phóng to cửa sổ chiếm toàn bộ màn hình;

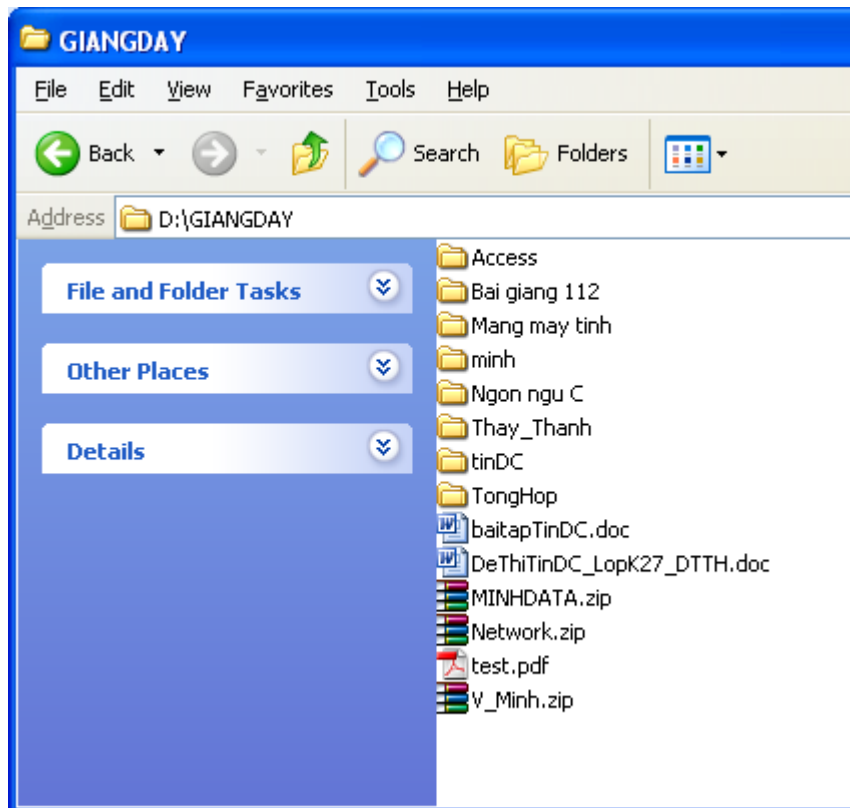


- là nút biến đổi cửa sổ về trạng thái có thể co giãn được;



- là nút đóng cửa sổ.

Cửa sổ hiện tại



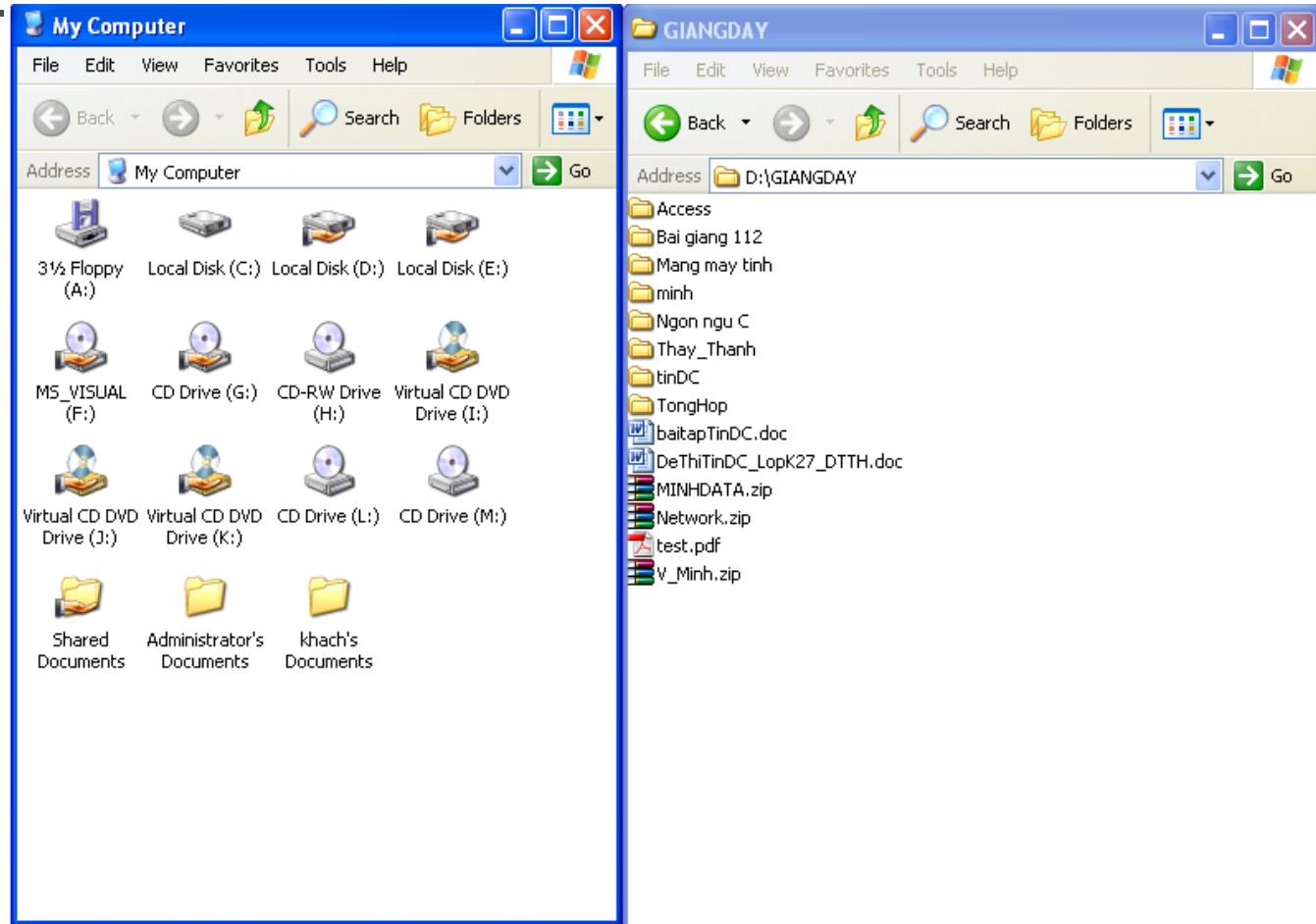
- Hệ điều hành MS-Windows cho phép mở nhiều thư mục trong cùng một cửa sổ.
Theo thứ tự mở thư mục chúng ta có các khái niệm sau:
- Cửa sổ hiện tại, là cửa sổ thư mục đang mở.
- cửa sổ ngay trước là cửa sổ thư mục xuất hiện ngay trước cửa sổ hiện tại.
- cửa sổ liền sau là cửa sổ đã được mở sau cửa sổ hiện tại.



Cửa sổ hoạt động

- Hệ điều hành MS-Windows cho phép mở nhiều cửa sổ khác nhau.
- Hệ điều hành cho phép mở nhiều cửa sổ nhưng tại một thời điểm nhất định chỉ có một cửa sổ thư mục cho phép thao tác, và được gọi là cửa sổ hoạt động. Theo hình minh họa dưới đây, cửa sổ hoạt động là cửa sổ My Computer. Thanh tiêu đề cửa sổ này hiển thị sáng màu.

Cửa sổ hoạt động



Hai cửa sổ đang mở, nhưng chỉ có cửa sổ My Computer là cửa sổ hoạt động

Sử dụng thanh công cụ cơ bản - Standard Buttons




- Nút được sử dụng để quay trở lại cửa sổ ngay trước cửa sổ hiện tại.
- Nút được sử dụng khi chúng ta đã có sử dụng nút Back. Nút này cho phép quay trở về cửa sổ đã có trước khi bấm nút Back.
- Nhấp nút để chuyển đến cửa sổ thư mục mẹ của cửa sổ thư mục hiện tại.
- Nhấp nút làm xuất hiện vùng tìm kiếm thông tin (Search). Nhấp lần thứ hai để đóng vùng tìm kiếm.
- Nhấp nút làm xuất hiện vùng hiển thị thư mục dạng cây. Nhấp lần thứ hai để đóng vùng hiển thị thư mục dạng cây.

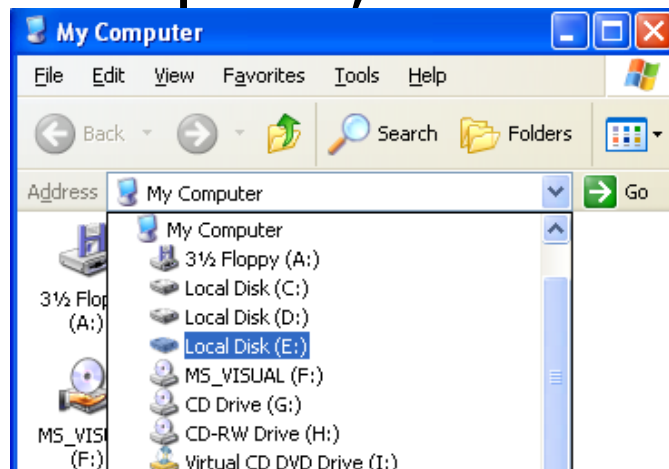
Ý nghĩa các mục trong hộp lệnh View

Thực hiện nhấp chuột lên dấu mũi tên làm xuất hiện nút Views bị che dấu. Trong hộp lệnh View có thể chọn các cách hiển thị tệp tin, thư mục trong cửa sổ:

- Mục Large Icons: khi chọn mục này, biểu tượng được hiển thị ở kích thước lớn.
- Mục Small Icons: khi chọn mục này, biểu tượng được hiển thị ở kích thước bé.
- Mục List: để hiển thị biểu tượng theo danh sách.
- Mục Details: để hiển thị biểu tượng với đầy đủ thông tin gồm: tên, kích thước,..
- Mục Thumbnails: cho phép hiển thị các tệp ảnh.

Thanh địa chỉ - Address

Trên thanh Address, bấm chọn mũi tên  làm xuất hiện danh sách biểu tượng để chọn lựa và mở các cửa sổ khác như: Desktop, My Documents, My Computer,...



Trên thanh địa chỉ cho phép chọn nhanh các ổ đĩa

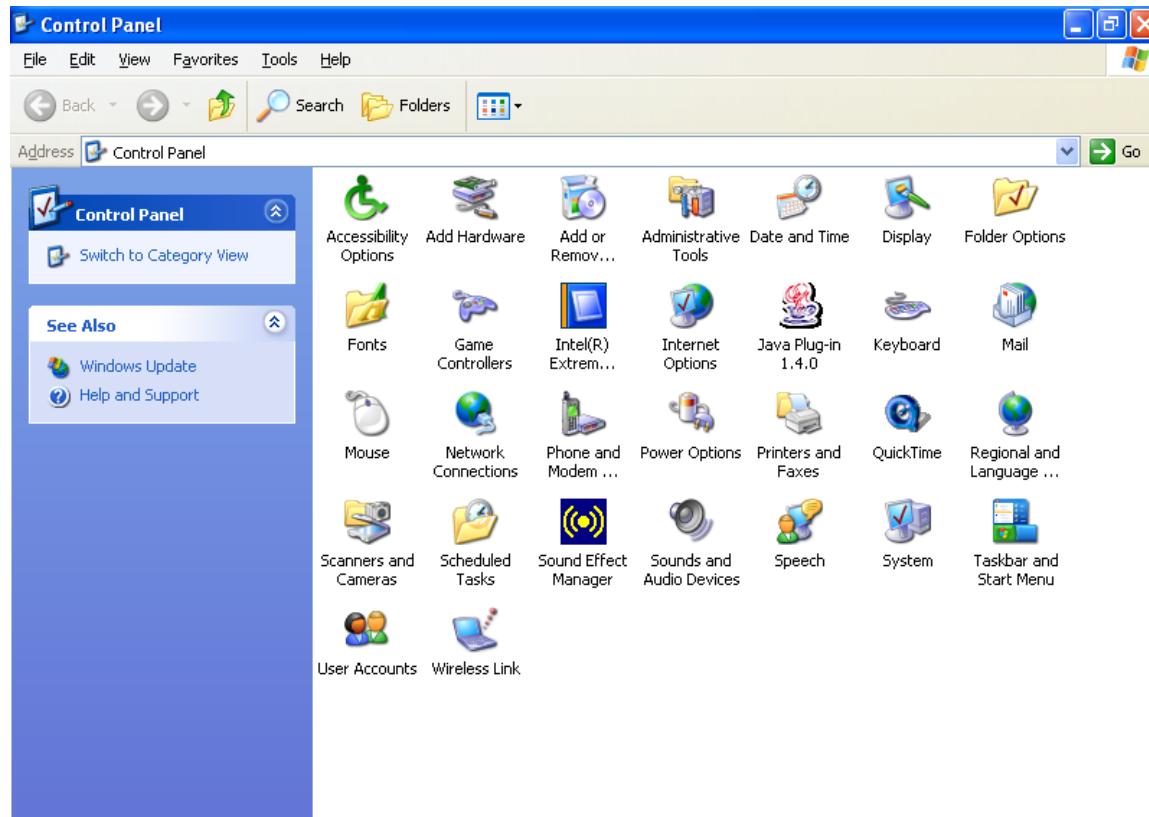
Di chuyển (hoán đổi) nhiều cửa sổ



- Mở ít nhất 3 cửa sổ: My computer, Recycle Bin, Winword.
- Để duy chuyển cửa sổ ta làm 3 cách sau:
 - **Cách 1:** Sử dụng các nút Minimize để thu nhỏ cửa sổ
 - **Cách 2:** Bấm chuột vào cửa sổ nằm trên thanh tác vụ
 - **Cách 3:** Sử dụng phím Alt + Tab

Cửa sổ Control Panel

- Nhấp chuột lên nút **Start** → **Settings** → **Control Panel** để mở cửa sổ Control Panel.



2010

TRƯỜNG ĐẠI HỌC
BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ
THÔNG TIN VÀ
TRUYỀN THÔNG

[BÀI GIẢNG TIN HỌC ĐẠI CƯƠNG - IT1110]

[Type the abstract of the document here. The abstract is typically a short summary of the contents of the document. Type the abstract of the document here. The abstract is typically a short summary of the contents of the document.]

THÔNG TIN MÔN HỌC

Trình độ:

Sinh viên đại học các ngành

Khối lượng: 4(3-1-1-8)

- Lý thuyết: 45 tiết
- Bài tập/BTL: 15 tiết (bài tập, thảo luận)
- Thí nghiệm: 5 bài (x 3 tiết)(thực hành)

Mô tả vắn tắt nội dung:

Khái niệm thông tin, biểu diễn thông tin trong máy tính. Hệ thống máy tính: phần cứng, phần mềm, hệ điều hành, mạng máy tính. Giải quyết bài toán bằng máy tính. Các cấu trúc lập trình cơ bản..

MỤC LỤC

PHẦN 1. TIN HỌC CĂN BẢN	8
I.1. Thông tin và biểu diễn thông tin (5T lý thuyết)	8
I.1.1. Các khái niệm cơ bản về thông tin và tin học (1T lý thuyết).....	8
I.1.1.1. Thông tin và xử lý thông tin.....	8
a. Thông tin - Dữ liệu – Tri thức.....	8
b. Quy trình xử lý thông tin.....	9
I.1.1.2. Máy tính điện tử và phân loại.....	9
a. Lịch sử phát triển của máy tính điện tử.....	9
b. Phân loại máy tính điện tử.....	10
I.1.1.3. Tin học và các ngành công nghệ liên quan.....	10
a. Tin học.....	10
b. Công nghệ thông tin (Information Technology - IT).....	11
c. Công nghệ thông tin và truyền thông.....	11
I.1.2. Biểu diễn dữ liệu trong máy tính (4T lý thuyết).....	11
I.1.2.1. Biểu diễn số trong các hệ đếm.....	11
a. Hệ đếm cơ số b.....	11
b. Hệ đếm thập phân (Decimal system, b=10).....	12
c. Hệ đếm nhị phân (Binary system, b=2).....	13
d. Hệ đếm bát phân (Octal system, b=8).....	13
e. Hệ đếm thập lục phân (Hexa-decimal system, b=16).....	13
f. Chuyển đổi một số từ hệ thập phân sang hệ đếm cơ số b.....	14
I.1.2.2. Biểu diễn dữ liệu trong máy tính và đơn vị thông tin.....	14
a. Nguyên tắc chung.....	14
b. Đơn vị thông tin.....	15
I.1.2.3. Biểu diễn số nguyên.....	16
a. Số nguyên không dấu.....	16
b. Số nguyên có dấu.....	16
c. Tính toán số học với số nguyên.....	17
Cộng/ trừ số nguyên.....	17
Nhân/ chia số nguyên.....	17
I.1.2.4. Biểu diễn số thực.....	18
a. Nguyên tắc chung.....	18
b. Chuẩn IEEE754/85.....	19
I.1.2.5. Biểu diễn ký tự.....	20
a. Nguyên tắc chung.....	20
b. Bộ mã ASCII.....	20
c. Bộ mã Unicode.....	22
I.2. HỆ THỐNG MÁY TÍNH (7 tiết)	23
I.2.1. Hệ thống máy tính (3T lý thuyết).....	23

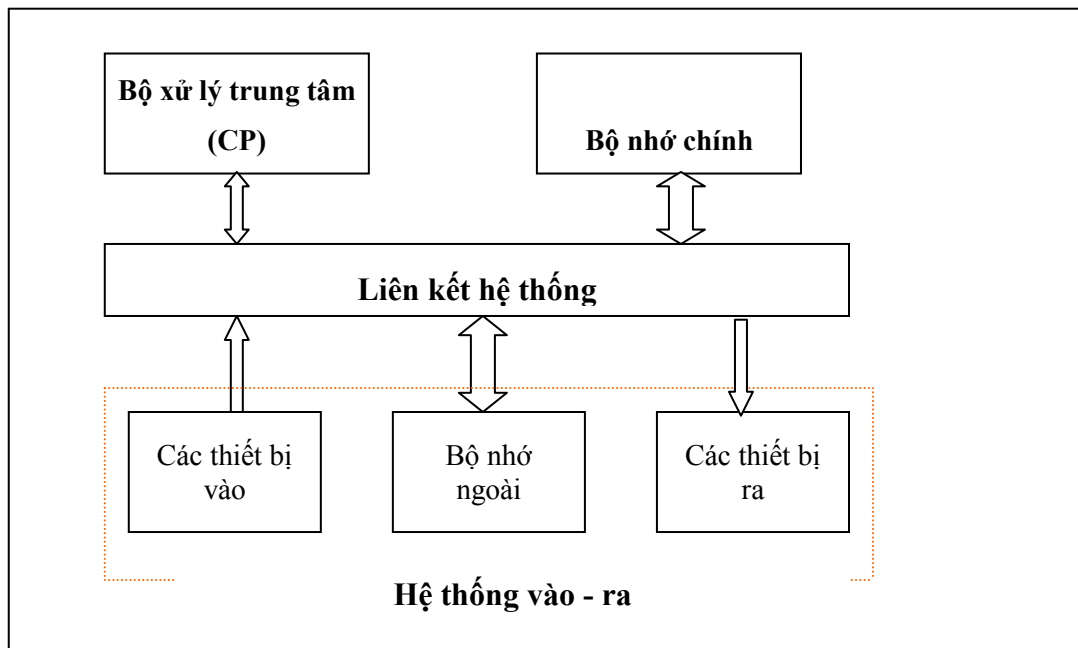
a. Mô hình cơ bản của máy tính.....	23
b. Bộ xử lý trung tâm – CPU	25
c. Bộ nhớ.....	26
d. Hệ thống vào-ra	27
e. Liên kết hệ thống (buses).....	29
I.2.1.2. Phần mềm máy tính	30
a. Dữ liệu và giải thuật.....	30
b. Chương trình và ngôn ngữ lập trình	35
c. Phân loại phần mềm máy tính.....	37
I.2.2. Mạng máy tính (2T lý thuyết).....	37
I.2.2.1. Lịch sử phát triển của mạng máy tính.....	37
I.2.2.2. Phân loại mạng máy tính.....	38
I.2.2.3. Các thành phần cơ bản của một mạng máy tính	38
I.2.2.4. Mạng Internet.....	40
I.2.3. Giới thiệu hệ điều hành (2T lý thuyết).....	41
I.2.3.1. Các khái niệm cơ bản.....	41
a. Khái niệm hệ điều hành	41
b. Tập (File)	41
c. Quản lý tệp của hệ điều hành	42
I.2.3.2. Hệ lệnh của hệ điều hành	43
I.2.3.3. Hệ điều hành Windows	44
a. Sự ra đời và phát triển.....	44
b. Khởi động và thoát khỏi Windows XP	44
c. Một số thuật ngữ và thao tác thường sử dụng.....	45
d. Cấu hình Windows (Control Panel).....	47
e. Windows Explorer	52
I.3. Các hệ thống ứng dụng (4T lý thuyết).....	56
I.3.1. Các hệ thống quản lý thông tin.....	56
I.3.1.1. Các khái niệm về thông tin và Hệ thống quản lý thông tin.....	56
I.3.1.2. Phân loại.....	56
a. Phân loại theo cấp bậc quản lý.....	56
b. Phân loại theo chức năng nghiệp vụ	59
c. Phân loại theo quy mô tích hợp	60
I.3.2. Hệ thông tin bảng tính.....	62
I.3.3. Hệ quản trị cơ sở dữ liệu	67
I.3.3.1. Cơ sở dữ liệu	67
a. Khái niệm về Cơ Sở Dữ Liệu.....	67
b. Hệ quản trị cơ sở dữ liệu.....	72
I.3.4. Các hệ thống thông minh	76
PHẦN II. GIẢI QUYẾT BÀI TOÁN.....	77

II.1. Thuật toán(4 tiết LT)	77
II.1.1. Định nghĩa thuật toán.....	77
II.1.2. Biểu diễn thuật toán.....	78
II.1.2.1. Ngôn ngữ lưu đồ.....	79
II.1.2.2. Mã giả.....	83
II.1.3. Thuật toán đệ qui.....	84
II.1.4. Một số thuật toán thông dụng.....	85
II.1.4.1. Thuật toán số học.....	85
II.1.4.2. Thuật toán về dãy.....	86
II.1.5. Thuật giải heuristic.....	87
II.1.5.1. Thuật giải – Sự mở rộng khái niệm của thuật toán.....	87
II.1.5.2. Thuật giải heuristic.....	87
II.2. Giải quyết bài toán (4 tiết LT, 2 tiết BT)	88
II.2.1. Khái niệm về bài toán.....	88
II.2.2. Các bước giải quyết bài toán bằng máy tính.....	89
Bài tập về Thuật toán (2 tiết BT)	90
II.2.3. Các phương pháp giải quyết bài toán bằng máy tính.....	90
II.2.3.1. Giải quyết bài toán theo hướng xác định trực tiếp lời giải:.....	90
II.2.3.2. Giải quyết bài toán theo hướng tìm kiếm lời giải.....	91
II.2.4. Phân loại bài toán.....	91
II.2.4.1. Độ phức tạp thuật toán.....	91
II.2.4.2. Phân loại bài toán.....	93
PHẦN 3. LẬP TRÌNH	95
III.1. Tổng quan về ngôn ngữ C (3 tiết LT)	95
III.1.1. Lịch sử phát triển.....	95
III.1.2. Các phần tử cơ bản của ngôn ngữ C.....	96
1.2.1. Tập kí tự.....	96
1.2.2. Từ khóa.....	96
1.2.3. Định danh.....	97
1.2.4. Các kiểu dữ liệu.....	98
1.2.5. Hằng.....	99
1.2.6. Biến.....	100
1.2.7. Hàm.....	100
1.2.8. Biểu thức.....	101
1.2.9. Câu lệnh.....	101
1.2.10. Chú thích.....	102
III.1.3. Cấu trúc cơ bản của một chương trình C.....	102

III.1.4. Biên dịch chương trình C	105
III.1.5. Trình biên dịch Turbo C++.....	105
III.1.6. Cài đặt và sử dụng Turbo C++ 3.0	105
III.2. Kiểu dữ liệu và biểu thức trong C (4 tiết LT).....	106
III.2.1. Các kiểu dữ liệu chuẩn trong C (1 tiết LT).....	107
III.2.2. Các biểu thức	109
III.2.3. Các phép toán	110
III.2.3.1. Phép toán số học.....	110
III.2.3.2. Phép toán quan hệ	111
III.2.3.3. Các phép toán logic	112
III.2.3.4. Phép toán gán	112
III.2.4. Thứ tự ưu tiên các phép toán	114
III.2.5. Một số toán tử đặc trưng trong C	115
III.3. Cấu trúc lập trình trong C (6 tiết LT).....	118
III.3.1. Vào/ra	118
III.3.1.1. Các lệnh vào ra dữ liệu với các biến (printf, scanf)	118
III.3.1.2. Các lệnh nhập xuất khác.....	123
III.3.2. Cấu trúc lệnh khối	124
III.3.3. Cấu trúc if.....	126
III.3.4. Cấu trúc lựa chọn switch	127
III.3.5. Vòng lặp for.....	130
III.3.6. Vòng lặp while và do – while.....	132
III.3.7. Các lệnh thay đổi cấu trúc lập trình.....	135
III.3.7.1. continue.....	136
III.3.7.2. break	137
III.4. Mảng và chuỗi ký tự (5 tiết LT)	138
III.4.1. Mảng.....	138
III.4.1.1. Khái niệm mảng	138
III.4.1.2. Khai báo và sử dụng mảng	138
III.4.1.3. Các thao tác cơ bản trên mảng	139
a. Nhập dữ liệu cho mảng.....	139
b. Xuất dữ liệu chứa trong mảng.....	140
c. Tìm phần tử có giá trị lớn nhất, phần tử có giá trị nhỏ nhất.....	142
III.4.1.4. Tìm kiếm trên mảng	142
III.4.1.5. Sắp xếp mảng	144
III.4.2. Chuỗi ký tự (2 tiết LT).....	146
III.4.2.1. Khái niệm chuỗi ký tự	146

III.4.2.2. Khai báo và sử dụng xâu	147
a. Khai báo xâu kí tự	147
b. Truy nhập vào một phần tử của xâu	147
III.4.2.3. Các hàm xử lý ký tự	147
III.4.2.4. Các hàm xử lý xâu	149
a. Vào ra dữ liệu	149
b. Một số hàm xử lí xâu kí tự khác	149
III.5. Cấu trúc (2 tiết LT)	151
III.5.1. Khái niệm cấu trúc	151
III.5.2. Khai báo và sử dụng cấu trúc	152
III.5.2.1. Khai báo kiểu dữ liệu cấu trúc	152
III.5.2.2. Khai báo biến cấu trúc:	152
III.5.2.3. Định nghĩa kiểu dữ liệu cấu trúc với typedef	153
III.5.3. Xử lý dữ liệu cấu trúc	154
III.5.3.1. Truy nhập các trường dữ liệu của cấu trúc	154
III.5.3.2. Phép gán giữa các biến cấu trúc	156
III.6. Hàm (2 tiết LT)	157
III.6.1. Khái niệm hàm	157
III.6.1.1. Khái niệm chương trình con	157
III.6.1.2. Phân loại chương trình con:	158
III.6.2. Khai báo và sử dụng hàm	158
III.6.2.1. Khai báo hàm	158
III.6.2.2. Sử dụng hàm	161
III.6.3. Phạm vi của biến	163

5	♣ 5	NAK 21	% 37	5 53	E 69	U 85	e 101	u 117
6	♠ 6	SYN 22	& 38	6 54	F 70	V 86	f 102	v 118
7	BEL 7	ETB 23	' 39	7 55	G 71	W 87	g 103	w 119
8	BS 8	CAN 24	(40	8 56	H 72	X 88	h 104	x 120
9	HT 9	EM 25) 41	9 57	I 73	Y 89	I 105	y 121
A	LF 10	SUB 26	* 42	: 58	J 74	Z 90	j 106	z 122
B	VT 11	ESC 27	+ 43	; 59	K 75	[91	k 107	{ 123
C	FF 12	FS 28	, 44	< 60	L 76	\ 92	l 108	 124
D	CR 13	GS 29	- 45	= 61	M 77] 93	m 109	} 125
E	SO 14	RS 30	. 46	> 62	N 78	^ 94	n 110	~ 126
F	SI 15	US 31	/ 47	? 63	O 79	<u>o</u> 95	o 111	DEL 127



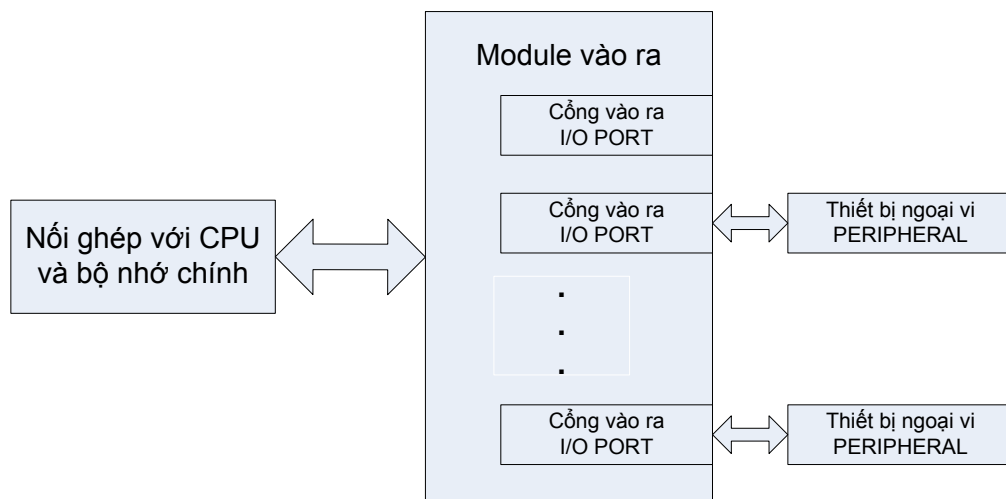
Hình I.2.1.1.a. Các thành phần chính của hệ thống máy tính

- **Bộ xử lý trung tâm – CPU:** Điều khiển các hoạt động của máy tính và thực hiện xử lý dữ liệu.
- **Bộ nhớ chính (Main Memory):** lưu trữ chương trình và dữ liệu.
- **Hệ thống vào ra (Input-Output System):** trao đổi thông tin giữa thế giới bên ngoài với máy tính.
- **Liên kết hệ thống (System Interconnection):** kết nối và vận chuyển thông tin giữa CPU, bộ nhớ chính và hệ thống vào ra của máy tính với nhau.

Hoạt động của máy tính.

Hoạt động cơ bản của máy tính là thực hiện chương trình. Chương trình gồm một tập các lệnh được lưu trữ trong bộ nhớ. Việc thực hiện chương trình là việc lặp lại *chu trình lệnh* bao gồm các bước sau:

- CPU phát địa chỉ từ con trỏ lệnh đến bộ nhớ nơi chứa lệnh cần nhận.
- CPU nhận lệnh từ bộ nhớ đưa về thanh ghi lệnh
- Tăng nội dung con trỏ lệnh để trỏ đến nơi lưu trữ lệnh kế tiếp
- CPU giải mã lệnh để xác định thao tác của lệnh
- Nếu lệnh sử dụng dữ liệu từ bộ nhớ hay công vào ra thì cần phải xác định địa chỉ nơi chứa dữ liệu.
- CPU nạp các dữ liệu cần thiết vào các thanh ghi trong CPU
- Thực thi lệnh
- Ghi kết quả vào nơi yêu cầu
- Quay lại bước đầu tiên để thực hiện lệnh tiếp theo.



Thiết bị vào ra

Mỗi thiết bị vào-ra làm nhiệm vụ chuyển đổi thông tin từ một dạng vật lý nào đó về dạng dữ liệu phù hợp với máy tính hoặc ngược lại. các thiết bị ngoại vi thông dụng như bàn phím, màn hình, máy in hay một máy tính khác. Người ta có thể phân các thiết bị ngoại vi ra nhiều loại:

- *Thiết bị thu nhận dữ liệu:* Bàn phím, chuột, máy quét ảnh,...
- *Thiết bị hiển thị dữ liệu:* màn hình, máy in,...
- *Thiết bị nhớ:* các loại ổ đĩa
- *Thiết bị truyền thông:* modem
- *Thiết bị hỗ trợ đa phương tiện:* hệ thống âm thanh, hình ảnh,...

Các thiết bị vào:

- **Bàn phím** (Keyboard, thiết bị nhập chuẩn): là thiết bị nhập dữ liệu và câu lệnh, bàn phím máy vi tính phổ biến hiện nay là một bảng chứa 104 phím có các tác dụng khác nhau.

Có thể chia làm 3 nhóm phím chính:

- Nhóm phím đánh máy: gồm các phím chữ, phím số và phím các ký tự đặc biệt (~, !, @, #, \$, %, ^, &, ?, ...).
- Nhóm phím chức năng (function keypad): gồm các phím từ F1 đến F12 và các phím như ← ↑ → ↓ (phím di chuyển từng điểm), phím PgUp (lên trang màn hình), PgDn (xuống trang màn hình), Insert (chèn), Delete (xoá), Home (về đầu), End (về cuối)
- Nhóm phím số (numeric keypad) như NumLock (cho các ký tự số), CapsLock (tạo các chữ in), ScrollLock (chế độ cuộn màn hình) thể hiện ở các đèn chỉ thị.

- Đầu ra: số nguyên lớn nhất của dãy a_k , k trong khoảng $[1 \dots N]$

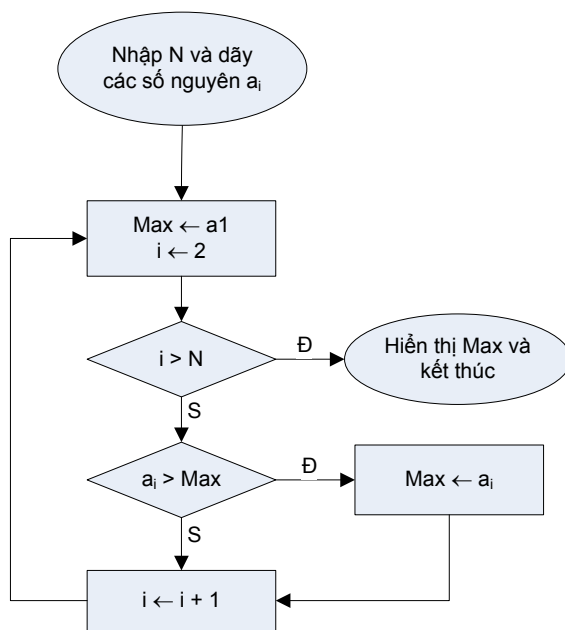
Ý tưởng

- Khởi tạo giá trị $Max = a_1$
- Lần lượt so sánh Max với $i=2, 3, \dots, N$, nếu $a_i > Max$ ta gán giá trị mới cho Max .

Thuật toán

- B1: Nhập dãy số a_i .
- B2: $Max \leftarrow a_1$.
- B3: Nếu $i > N$, thuật toán kết thúc và Max là giá trị lớn nhất của dãy cần tìm
- B4: Nếu $a_i > Max$, gán a_i cho Max .
- B5: Tăng i lên 1 đơn vị.
- B6: Quay lên b4.
- B7: Kết thúc.

Vì thuật toán là chỉ ra dãy thao tác và trình tự thao tác để đạt mục đích và dùng cho người dùng con người, nên ngoài cách liệt kê trên, người ta có thể dùng sơ đồ khối để minh họa (biểu diễn). Với thuật toán trên, cách biểu diễn theo sơ đồ khối như sau:



Qui ước:

- Hình thoi biểu diễn thao tác so sánh
- Hình chữ nhật thể hiện 1 thao tác tính toán đơn giản hay phức tạp
- Hình ô van thể hiện thao tác bắt đầu hay kết thúc.

Thí dụ 2: Sắp xếp bằng phương pháp tráo đổi (Exchange Sort)

- Đầu vào: Dãy A gồm N số nguyên a_1, a_2, \dots, a_N
- Đầu ra: Dãy A được sắp lại theo thứ tự không giảm

Ý tưởng

- Với mỗi cặp số liên tiếp trong dãy , nếu số trước không lớn hơn số sau ta đổi chỗ chúng cho nhau.
- Việc đó được lặp cho đến khi không có sự đổi chỗ nào cho nhau.

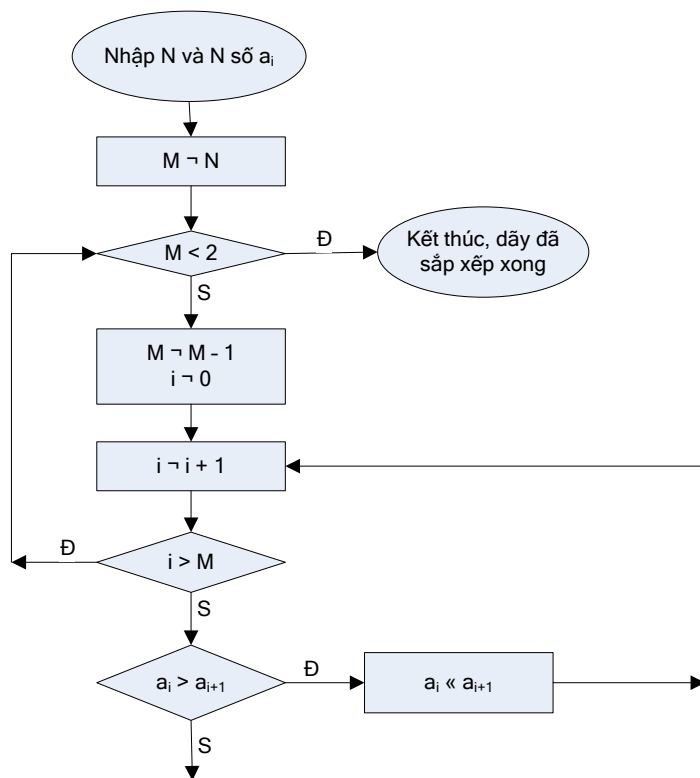
Thuật toán

- B1: Nhập số N và dãy số a_1, a_2, \dots, a_N
- B2: $M \leftarrow N$.
- B3: Nếu $M < 2$ thì thuật toán kết thúc và hiển thị dãy đó.
- B4: $M \leftarrow M - 1, i \leftarrow 0$.
- B5: Tăng i lên 1 đơn vị.
- B6: Nếu $i > M$ thì quay lại bước 3.
- B7: Nếu $a_i > a_{i+1}$ thì trao đổi hai số đó cho nhau .
- B8: Quay lên bước 5.

Chú ý:

- Thuật toán này tạo ra sau mỗi lượt sắp một phần tử đúng vị trí và phần tử này không còn tham gia vào quá trình sắp nữa. Nó giống như bọt nước nổi lên mặt nước: bóng nhẹ sẽ được đẩy dần lên trên. Cũng chính vì thế mà sắp xếp trao đổi còn có tên “*nổi bọt*” (Bubble Sort).
- Quá trình được lặp lại với dãy sau khi đã bỏ phần tử cuối dãy, do vậy lúc đầu M được gán với giá trị N và dừng khi $M < 2$.
- Trong thuật toán trên, i được khởi tạo giá trị 0 và tiến tới $M + 1$

Với thuật toán trên, cách biểu diễn theo sơ đồ khối như sau:



Ngoài cách biểu diễn theo kiểu liệt kê hay sơ đồ khối, còn nhiều cách biểu diễn khác. Xu hướng chung hiện nay là sử dụng các ngôn ngữ lập trình như Pascal, C/C++ hay Java. Tuy nhiên, không nhất thiết phải sử dụng đúng ký pháp của các ngôn ngữ đó mà có thể được bỏ một số ràng buộc.

Cũng có khi người ta dùng một giả ngữ gọi là ngôn ngữ mô phỏng chương trình PDL (Programming Description Language).

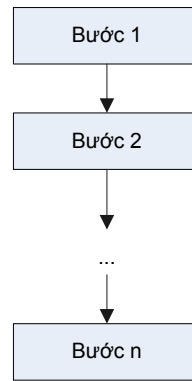
Điều quan trọng của giải thuật mà chúng ta công nhận ở đây là giải thuật phải thỏa mãn 3 tiêu chí sau:

1. Tính hữu hạn: giải thuật phải dừng sau một thời gian hữu hạn.
2. Khi kết thúc, giải thuật phải cung cấp kết quả đúng đắn.
3. Tính hiệu quả: thời gian tính toán nhanh, sử dụng ít tài nguyên không gian như bộ nhớ, thiết bị,... Hơn nữa, giải thuật cần mang tính phổ dụng, dễ hiểu, dễ cài đặt và mở rộng cho các lớp bài toán khác.

Theo phong cách lập trình cấu trúc, người ta khuyến cáo, giải thuật được xây dựng với 3 cấu trúc cơ bản sau:

1. Cấu trúc tuần tự:

Các bước được thực hiện theo 1 trình tự tuyến tính, hết bước này đến bước khác



2. Cấu trúc rẽ nhánh

Việc thực hiện bước nào phụ thuộc vào điều kiện xác định. Thí dụ tìm max của 2 số a, b . Nếu $a > b$ thì max là a , ngược lại max sẽ là b . Theo cách biểu diễn giải thuật ta có thể mô tả như sau:

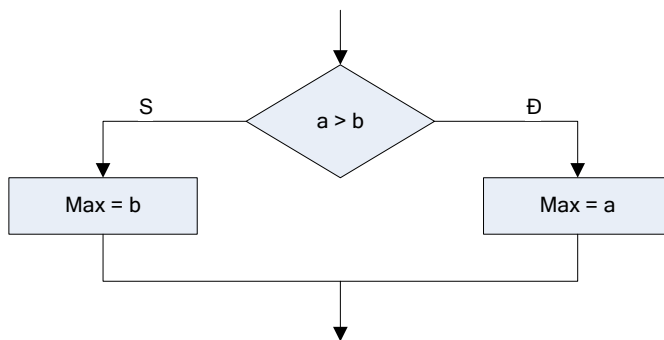
B1: Nhập 2 số a, b .

B2: Nếu $a > b$ thì $\max = a$ và đi đến bước kết thúc (B4).

B3: $\max = b$ ($a < b$).

B4: Kết thúc.

Cấu trúc rẽ nhánh này thường được biểu diễn dưới dạng sơ đồ như sau:

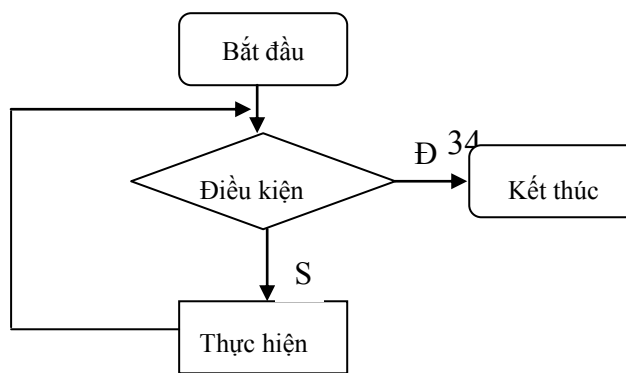


Cấu trúc rẽ nhánh (2 nhánh) có thể mở rộng cho nhiều nhánh

3. Cấu trúc lặp

Một tác động/ nhiệm vụ có thể được thực hiện lặp nhiều lần. Tuy nhiên số lần lặp phải hữu hạn, và số lần lặp có thể biết trước hoặc không biết trước. Thí dụ tìm số lớn nhất của 1 dãy có n số. Để thực hiện yêu cầu này, ta lần lượt phải so sánh số max tạm thời (lúc đầu max được gán bằng phần tử thứ nhất, a_1) với a_i , với i từ 2, 3, ..., n . Việc so sánh này được thực hiện lặp nhiều lần giữa max và a_i . Khi kết thúc quá trình lặp, ta sẽ thu được số max, số lớn nhất của dãy n số.

Sơ đồ chung của cấu trúc lặp:



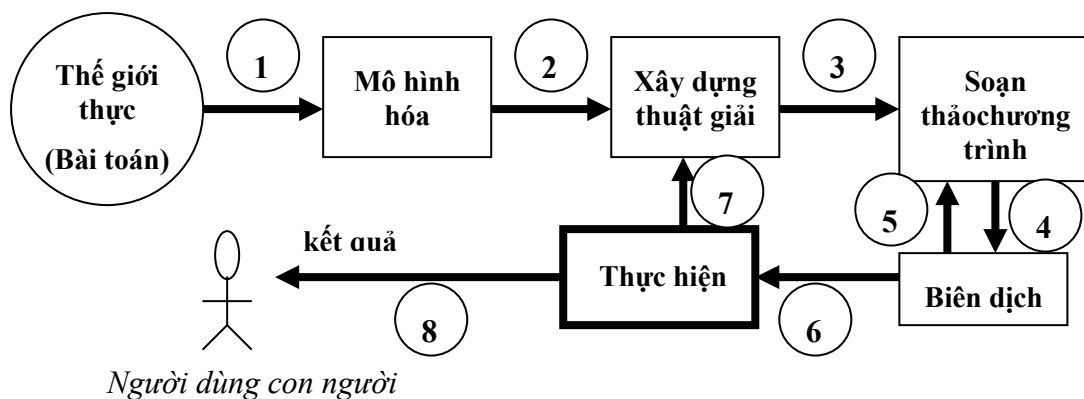
của chương trình đến lệnh cuối cùng nếu không có lỗi. Bộ thông dịch này giống như vai trò của 1 thông dịch viên (translator).

- *Biên dịch (Compiler)*: Khác với thông dịch, trình biên dịch dịch toàn bộ chương trình nguồn sang ngôn ngữ đích. Với chương trình đích này, máy đã có thể hiểu được và biết cách thực thi. Quá trình biên dịch sẽ tạo ra chương trình đích chỉ khi các lệnh trong chương trình nguồn không có lỗi.

Cách ứng xử với lỗi cũng khác nhau. Có chương trình dịch cứ gặp lỗi thì ngưng lại như Turbo Pascal, song có chương trình dịch cứ dịch cho đến hết kể cả khi gặp lỗi như C hay 1 số ngôn ngữ khác.

Hiện tại có khá nhiều ngôn ngữ lập trình bậc cao. Ngôn ngữ đầu tiên phải kể đến là FORTRAN4 xuất hiện vào năm 1959 sau đó cải tiến thành phiên bản FORTRAN5 năm 1977. Tiếp theo là COBOL 1960, ngôn ngữ sử dụng cho nghiệp vụ quản lý. Vào những năm tiếp theo là ALGOL60, BASIC. Có những ngôn ngữ đến nay đã không còn được sử dụng hay đã được hoàn thiện hơn như Visual Basic có nguồn gốc từ Basic. Nhiều ngôn ngữ mới ra đời như C, C++, Visual C, Java, C#, Python,...

Quá trình giải quyết một bài toán trên máy tính có thể biểu diễn theo một sơ đồ chung sau:



Hình 3.5 Quy trình giải quyết một bài toán trên máy tính

Chú ý:

- Bước 5: quay lại soạn thảo khi quá trình dịch phát hiện lỗi cú pháp trong chương trình nguồn
- Bước 7: xem lại giải thuật khi kết quả thực hiện không đúng (lỗi lô gíc).

Theo cách tiếp cận về phần mềm, các nhà khoa học và các nhà phát triển phần mềm đã định ra qui trình phát triển phần mềm máy tính bao gồm các bước sau:

- B1 Xác định bài toán: Thuật ngữ mới cho bước này là xác định yêu cầu người dùng, người mong muốn có phần mềm để sử dụng.
- B2 Phân tích bài toán: Tìm hiểu nhiệm vụ (chức năng) mà phần mềm cần xây dựng phải có và các dữ liệu cần thiết. Qua đó xây dựng các giải pháp khả thi. Nói một cách ngắn gọn, bước này tìm hiểu hệ thống là gì? Và làm gì?
- B3 Thiết kế hệ thống: thực hiện thiết kế kiến trúc hệ thống, thiết kế các mô đun chương trình, thiết kế giao tiếp, thiết kế an toàn,... Như vậy, nhiệm vụ thiết kế mô đun chính là xây dựng giải thuật cho mô đun đó và cách diễn tả giải thuật.
- B4 Xây dựng chương trình : Viết code cho các mô đun theo ngôn ngữ lập trình đã xác định.

- Các thiết bị đầu cuối (*terminal*). Ví dụ: ...?
- Các phụ kiện mạng

Các phụ kiện mạng khác gồm giắc cắm, ổ cắm,

- Hệ điều hành mạng

Hệ điều hành mạng là một phần mềm điều khiển sự hoạt động của mạng.

- Các phần mềm mạng cho máy tính

Hiện nay nói chung các hệ điều hành đều sẵn có khả năng kết nối mạng. Trong trường hợp hệ điều hành của máy tính không có sẵn khả năng kết nối mạng thì các phần mềm này là cần thiết.

- Các ứng dụng trên mạng.

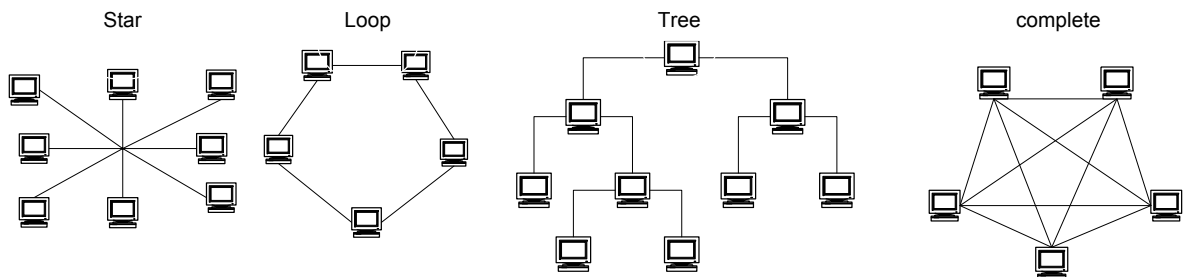
Ví dụ như Email, hệ quản trị cơ sở dữ liệu.

- Kiến trúc mạng máy tính

Kiến trúc mạng máy tính (network architecture) thể hiện cách kết nối máy tính với nhau và qui ước truyền dữ liệu giữa các máy tính như thế nào. Cách nối các máy tính với nhau gọi là *hình trạng* (*topology*) của mạng. Tập các qui ước truyền thông gọi là *giao thức* (*protocol*).

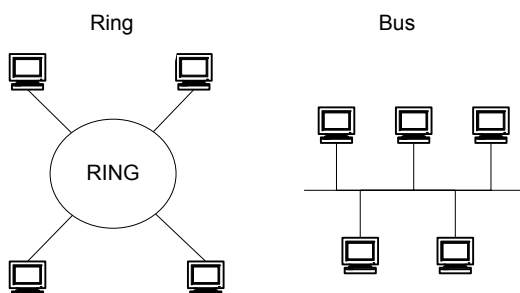
- Có hai kiểu nối mạng chủ yếu là *điểm-điểm* (point to point) và *quảng bá* (broadcast). Trong kiểu điểm-điểm các đường truyền nối các nút thành từng cặp. Như vậy một nút sẽ gửi dữ liệu đến nút lân cận nó (nút được nối trực tiếp với nó). Nút lân cận sẽ chuyển tiếp dữ liệu như vậy cho đến khi dữ liệu đến đích.

Kiểu nối mạng điểm- điểm có ba dạng chính là : hình sao (*star*), chu trình (*loop*), cây (*tree*) và đầy đủ (*complete*).



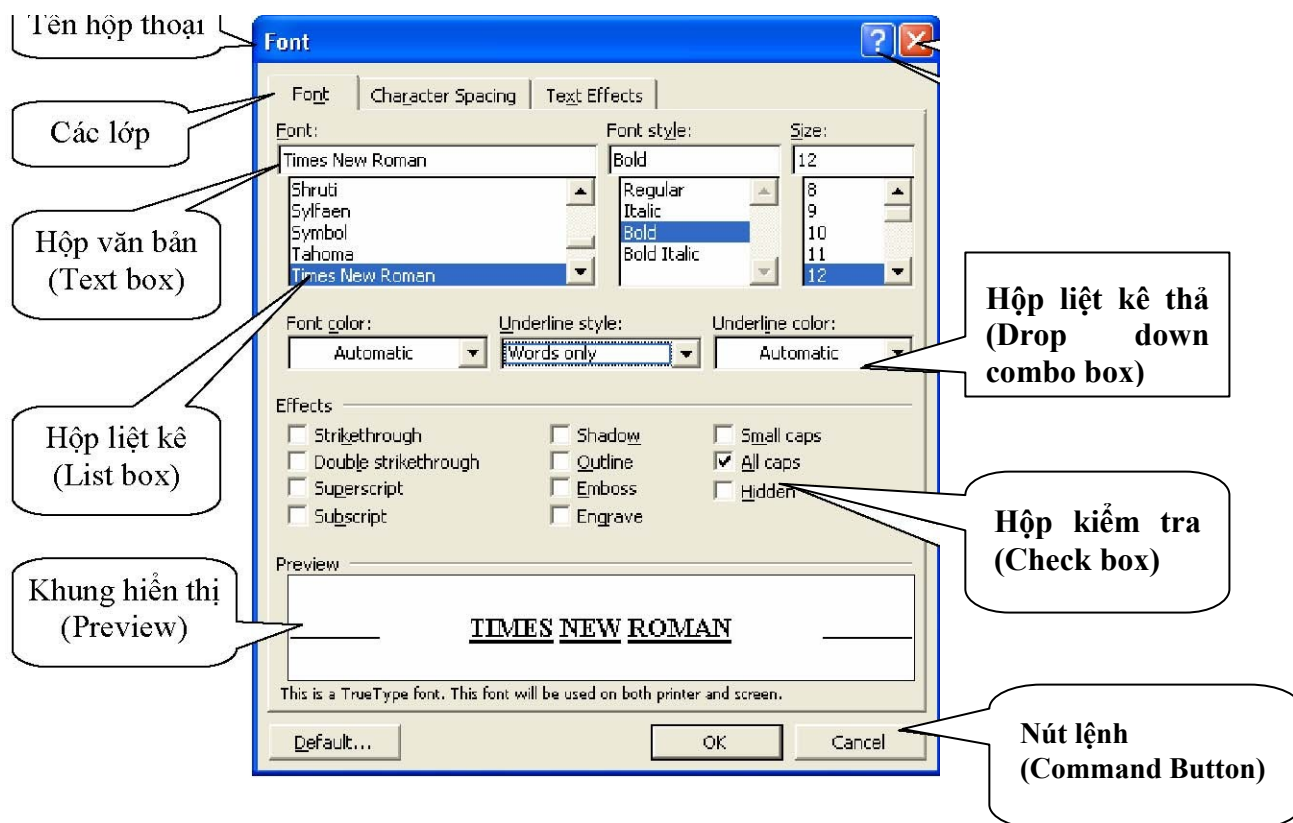
Hình 4.5 : Một số topo mạng kiểu điểm- điểm

Trong kiểu quảng bá các nút nối vào đường truyền chung. Như vậy khi một nút gửi dữ liệu các nút còn lại đều nhận được. Do đó dữ liệu gửi đi cần có địa chỉ đích. Khi một nút nhận được dữ liệu nó sẽ kiểm tra địa chỉ đích xem có phải gửi cho mình không.



Hộp hội thoại (Dialogue box)

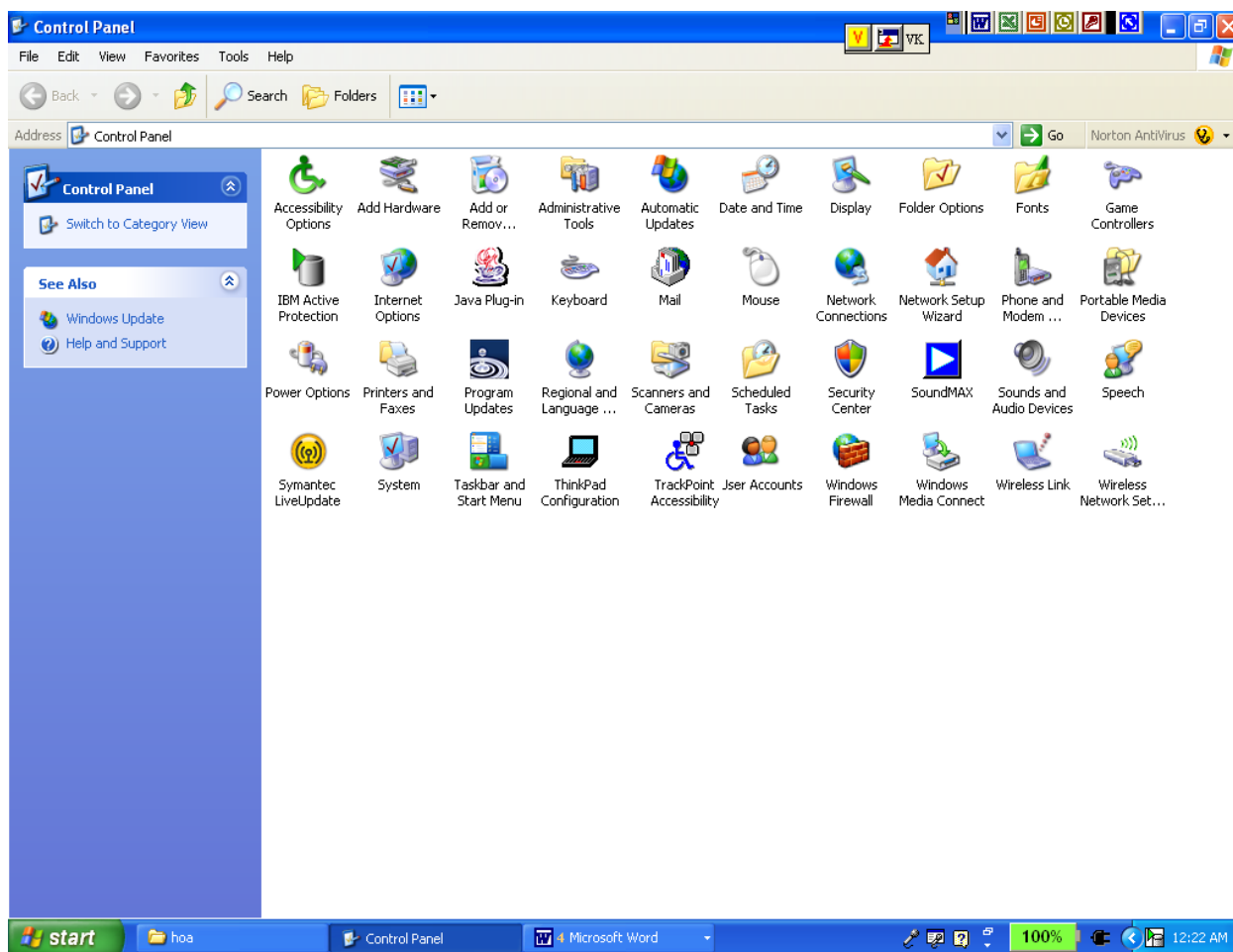
Trong khi làm việc với Windows và các chương trình ứng dụng chạy dưới môi trường Windows bạn thường gặp những hộp hội thoại. Các hộp thoại này xuất hiện khi nó cần thêm những thông số để thực hiện lệnh theo yêu cầu của bạn. Hình dưới đây giới thiệu các thành phần của hộp hội thoại



Thông thường, trên một hộp hội thoại sẽ có các thành phần sau:

- Hộp văn bản (Text box): dùng để nhập thông tin.
- Hộp liệt kê (List box): liệt kê sẵn một danh sách có các mục có thể chọn lựa, nếu số mục trong danh sách nhiều không thể liệt kê hết thì sẽ xuất hiện thanh trượt để cuộn danh sách.
- Hộp liệt kê thả (Drop down list box/ Combo box): khi nhấp chuột vào nút thả thì sẽ bung xuống một danh sách.
- Hộp kiểm tra (Check box): có 2 dạng, dạng hình vuông thể hiện việc cho phép không chọn, chọn 1 hoặc nhiều mục không loại trừ lẫn nhau. Dạng ô tròn (Option button): bắt buộc phải chọn một trong số các mục. Đây là những lựa chọn loại trừ lẫn nhau.
- Nút lệnh (Command Button): lệnh cần thực thi. Các loại nút lệnh thường gặp có:
 - + **OK (hoặc bấm phím Enter)**: thực hiện lệnh (chấp nhận)
 - + **Close**: giữ lại các thông số đã chọn và đóng cửa sổ
 - + **Cancel (hay nhấn phím Esc)**: không thực hiện lệnh (từ chối thực hiện)
 - + **Apply**: áp dụng các thông số đã chọn.
 - + **Default**: đặt mặc định theo các thông số

ii. Cửa sổ làm việc của Control Panel:



iii. Cài đặt và loại bỏ Font chữ

Để cài đặt thêm những Font chữ khác hoặc loại bỏ các Font chữ, ta chọn chương trình **Fonts**

- **Loại bỏ font chữ.** Từ cửa sổ Fonts
 - Chọn những Font cần loại bỏ
 - Chọn File/ Delete (hoặc nhấn phím Delete).
- **Thêm font chữ mới** Từ cửa sổ Fonts, chọn lệnh File/Install New Font, xuất hiện hộp thoại Add Fonts. Trong hộp thoại này **chỉ ra nơi chứa các Font nguồn muốn thêm bằng cách chọn tên ổ đĩa** chứa các tập tin Font chữ, sau đó chọn các tên Font và Click OK.

iv. Thay đổi dạng hiện màn hình desktop

Chọn lệnh **Start/ Settings/ Control Panel/ Display** hoặc **R_Click trên màn hình nền (Desktop)**, chọn Properties. Xuất hiện cửa sổ Display Properties với các thành phần như sau:

Desktop: Chọn ảnh nền cho Desktop bằng cách Click chọn các ảnh nền có sẵn hoặc Click vào nút Browse để chọn tập tin ảnh không có trong danh sách những ảnh có sẵn.

Screen Saver: xác lập màn hình nghỉ

Settings: Thay đổi chế độ màu và độ phân giải của màn hình. - Chế độ màu càng cao thì hình ảnh càng đẹp và rõ nét. Các chế độ màu: 64.000 màu (16 bits) , 16 triệu màu (24 bits). Chế độ màu trên mỗi máy tính có thể khác nhau tùy thuộc vào dung lượng bộ nhớ của card màn hình. Độ phân giải càng lớn thì màn hình càng hiển thị được nhiều thông tin .

v. *Cài đặt và loại bỏ chương trình:*

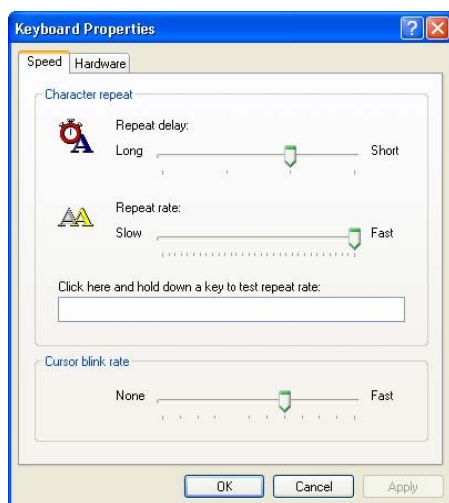
Để cài đặt các chương trình mới hoặc loại bỏ các chương trình không còn sử dụng bạn nhấn đúp chuột vào biểu tượng **Add or Remove Programs** trong cửa sổ Control Pane xuất hiện hộp thoại và thao tác theo chỉ dẫn.

vi. *Cấu hình ngày, giờ cho hệ thống*

Bạn có thể thay đổi ngày giờ của hệ thống bằng cách Click lên biểu tượng đồng hồ trên thanh Taskbar hoặc vào Control Panel, chọn nhóm **Date/Time - Date & Time**: thay đổi ngày, tháng, năm, giờ, phút, giây.

vii. *Thay đổi thuộc tính của bàn phím và chuột*

Lệnh Start/ Settings/ Control Panel, rồi chọn biểu tượng Mouse



Thay đổi thuộc tính của bàn phím:

Repeat delay: thay đổi thời gian trễ cho phím.

Repeat rate: thay đổi tốc độ lặp lại khi nhấn phím

Thay đổi thuộc tính của thiết bị chuột:

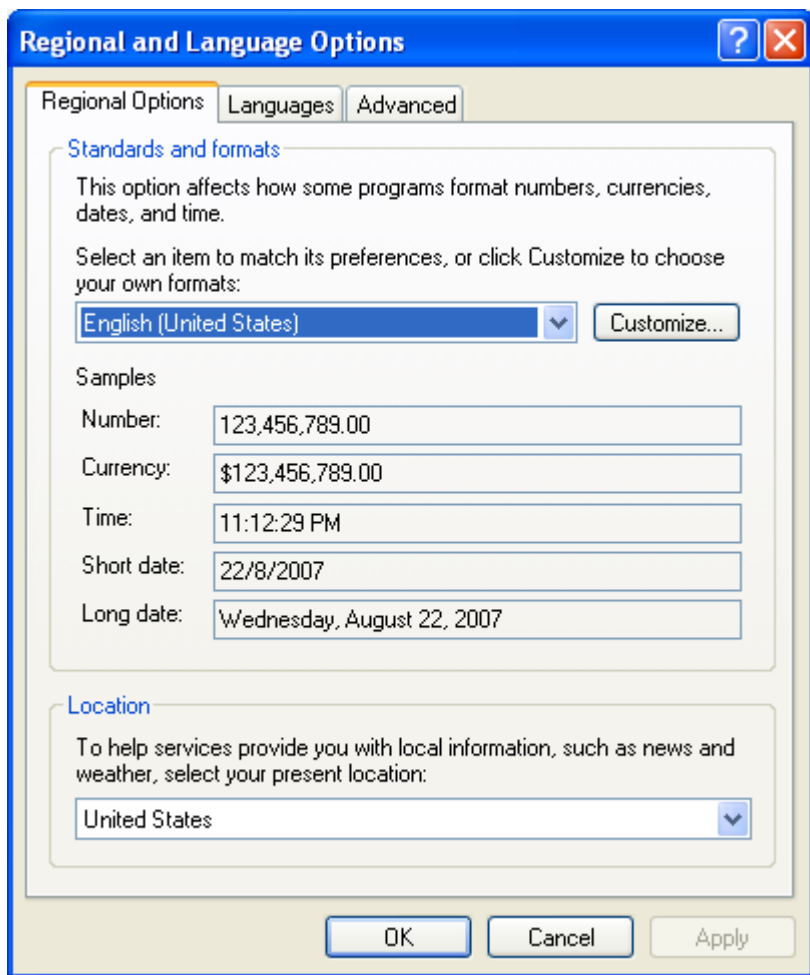
Lệnh Start/ Settings/ Control Panel, rồi chọn biểu tượng Keyboard

Lớp Buttons: thay đổi phím trái và phím chuột phải (thuận tay trái hay phải) và tốc độ nhấp đúp chuột.

Lớp Pointers: cho phép chọn hình dạng trỏ chuột trong các trạng thái làm việc.

viii. *Thay đổi thuộc tính vùng (Regional Settings)*

Bạn có thể thay đổi các thuộc tính như định dạng tiền tệ, đơn vị

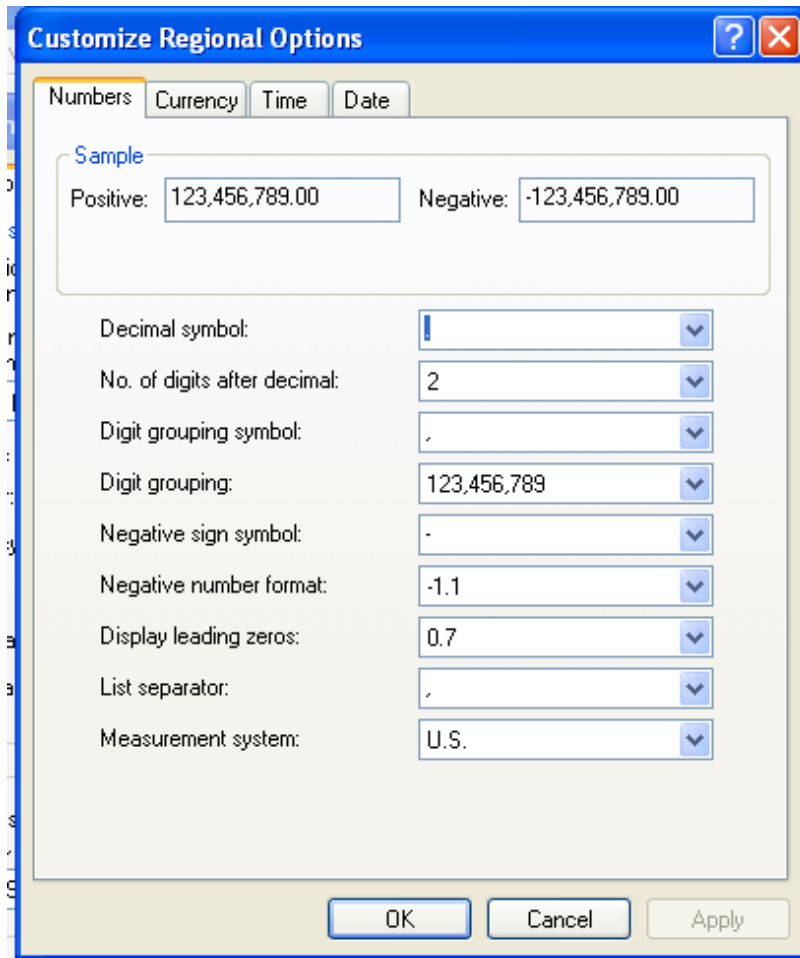


Lệnh Start/ Settings/ Control Panel/ Regional and Language Options

Lớp Regional Options: Thay đổi thuộc tính vùng địa lý, sau đó sẽ kéo theo sự thay đổi các thuộc tính của Windows. Click chọn **Customize**, cửa sổ Customize để thay đổi qui ước về dạng số, tiền tệ, thời gian, ngày tháng.

- **Number:** Thay đổi định dạng số, cho phép định dạng việc hiển thị
 - + Decimal symbol: Thay đổi ký hiệu phân cách hàng thập phân.
 - + No. of digits after decimal: Thay đổi số các số lẻ ở phần thập phân.
 - + Digit grouping symbol: Thay đổi ký hiệu phân nhóm hàng ngàn.
 - + Digit grouping: Thay đổi số ký số trong một nhóm (3 số / 4 số/ ...)
 - + Negative sign symbol: Thay đổi ký hiệu của số âm.
 - + Negative number format: Thay đổi dạng thể hiện của số âm.
 - + Display leading zeros: **0.7** hay **.7**.
 - + Measurement system: Chọn hệ thống đo lường như cm, inch, ...
 - + List separator: Chọn dấu phân cách giữa các mục trong danh sách
- **Currency:** Thay đổi định dạng tiền tệ (\$, VND, ...)
- **Time:** Thay đổi định dạng giờ theo chế độ 12 giờ hay 24 giờ

- **Date:** Thay đổi định dạng ngày tháng (Date), cho phép chọn cách thể hiện ngày.



ix. Cài đặt / loại bỏ máy in

Cài đặt thêm máy in:

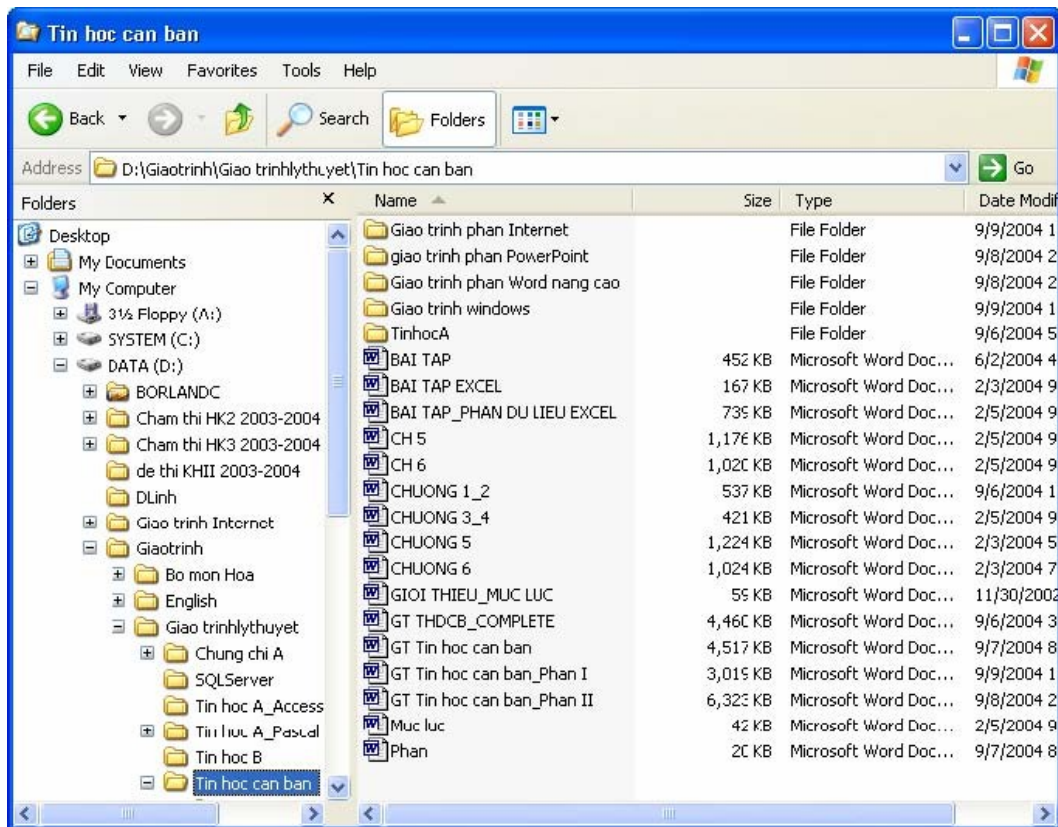
Với một số máy in thông dụng Windows đã tích hợp sẵn chương trình điều khiển (driver) của các máy in, tuy nhiên cũng có những máy in mà trong Windows chưa có chương trình điều khiển. Muốn sử dụng những máy in này ta cần phải gọi thực hiện chương trình **Printers and Faxes** trong Control Panel.

Các bước cài đặt máy in:

- Chọn lệnh Start/ Settings/ Printers and Faxes
- Click chọn Add a Printer, xuất hiện hộp thoại Add
- Làm theo các bước hướng dẫn của hệ thống

Loại bỏ máy in đã cài đặt

- Chọn lệnh Start/ Settings/ Printers and Faxes
- Click chuột chọn máy in muốn loại bỏ
- Nhấn phím Delete, sau đó chọn Yes



Hình 4.1: Cửa sổ Windows Explorer

Thư mục (Folder)

	- Back: Chuyển về thư mục trước đó.
	- Up: Chuyển lên thư mục cha.
	- Forward: Chuyển tới thư mục vừa quay về (Back).
	- Search: Tìm kiếm tập tin/ thư mục.
	- Folder: Cho phép ẩn/ hiện cửa sổ Folder bên trái.
<ul style="list-style-type: none"> Thumbnails Tiles Icons List ● Details 	- Views: Các chế độ hiển thị các đối tượng (tập tin/ thư mục/ ổ đĩa)

Nội dung trong cửa sổ có thể được sắp xếp thể hiện theo thứ tự. Đối với kiểu thể hiện Details, bạn có thể thực hiện bằng cách luân phiên nhấn chuột lên cột tương ứng (Name, Size, Type, Date Modified).

Trong các kiểu thể hiện khác bạn có thể thực hiện bằng lệnh **View/ Arrange Icons By** và lựa chọn tiếp một trong các khóa sắp xếp (theo tên, kích cỡ tập tin, kiểu tập tin, hoặc ngày tháng cập nhật).

Trong kiểu thể hiện bằng các biểu tượng lớn và biểu tượng nhỏ bạn có thể để Windows sắp xếp tự động bằng lệnh **View/ Arrange Icons By / Auto Arrange**. Tùy chọn Auto Arrange chỉ áp dụng cho cửa sổ của thư mục hiện hành.

Thao tác với thư mục và tệp

i. Mở tập tin/ thư mục:

Có ba cách thực hiện :

- Cách 1: D_Click lên biểu tượng của tập tin/ thư mục.
- Cách 2: R_Click lên biểu tượng của tập tin/ thư mục và chọn mục Open.
- Cách 3: Chọn tập tin/ thư mục và nhấn phím Enter.

Nếu tập tin thuộc loại tập tin văn bản thì chương trình ứng dụng kết hợp sẽ được khởi động và tài liệu sẽ được nạp vào.

Trong trường hợp chương trình ứng dụng không được cài đặt trong máy tính thì Windows sẽ mở hộp thoại Open With và cho chọn chương trình kết hợp. Nếu tập tin thuộc dạng chương trình ứng dụng thì chương trình tương ứng sẽ được khởi động.

ii. Chọn tập tin/ thư mục:

- Chọn một tập tin/ thư mục: Click lên biểu tượng tập tin/ thư mục.
- Chọn một nhóm tập tin/ thư mục: có thể thực hiện theo 2 cách:
 - Các đối tượng cần chọn là một danh sách gồm các đối tượng liên tục: Click lên đối tượng đầu danh sách để chọn, sau đó nhấn giữ phím Shift và Click lên đối tượng ở cuối danh sách.
 - Các đối tượng cần chọn nằm rời rạc nhau: nhấn giữ phím Ctrl và Click chọn các đối tượng tương ứng.

iii. Tạo thư mục

- Chọn nơi chứa thư mục cần tạo (thư mục/ ổ đĩa ở cửa sổ bên trái).
- Chọn menu **File/ New/ Folder** hoặc **R_Click/ New/ Folder**.
- Nhập tên thư mục mới, sau đó gõ Enter để kết thúc.

iv. Sao chép thư mục và tập tin

Chọn các thư mục và tập tin cần sao chép. Sau đó có thể thực hiện theo một trong hai cách sau:

- Cách 1: Nhấn giữ phím Ctrl và Drag đối tượng đã chọn đến nơi cần chép.
- Cách 2: Nhấn tổ hợp phím Ctrl + C (hoặc Edit/ Copy hoặc R_Click và chọn Copy) để chép vào Clipboard, sau đó chọn nơi cần chép đến và nhấn tổ hợp phím Ctrl + V (hoặc Edit/ Paste hoặc R_Click và chọn Paste).

v. Di chuyển thư mục và tập tin

Chọn các thư mục và tập tin cần di chuyển. Sau đó có thể thực hiện theo một trong hai cách sau:

- Cách 1: Drag đối tượng đã chọn đến nơi cần di chuyển.
- Cách 2: Nhấn tổ hợp phím Ctrl + X (hoặc Edit/ Cut hoặc R_Click và chọn Cut) để chép vào Clipboard, sau đó chọn nơi cần di chuyển đến và nhấn tổ hợp phím Ctrl + V (hoặc Edit/ Paste hoặc R_Click và chọn Paste).

vi. Xoá thư mục và tập tin

- Chọn các thư mục và tập tin cần xoá.
- Chọn File/ Delete
- hoặc: Nhấn phím Delete
- hoặc: R_Click và chọn mục Delete.
- Xác nhận có thực sự muốn xoá hay không (Yes/ No)

vii. Phục hồi thư mục và tập tin

Các đối tượng bị xoá sẽ được đưa vào Recycle Bin. Nếu muốn phục hồi các đối tượng đã xoá, bạn thực hiện các thao tác sau đây:

- D_Click lên biểu tượng Recycle Bin
- Chọn tên đối tượng cần phục hồi.
- Thực hiện lệnh **File/ Restore** hoặc R_Click và chọn mục **Restore**.

Ghi chú: Nếu muốn xoá hẳn các đối tượng, ta thực hiện thao tác xoá một lần nữa đối với các đối tượng ở trong Recycle Bin. Nếu muốn xoá hẳn tất cả các đối tượng trong Recycle Bin, R_Click lên mục Recycle Bin và chọn mục Empty Recycle Bin.

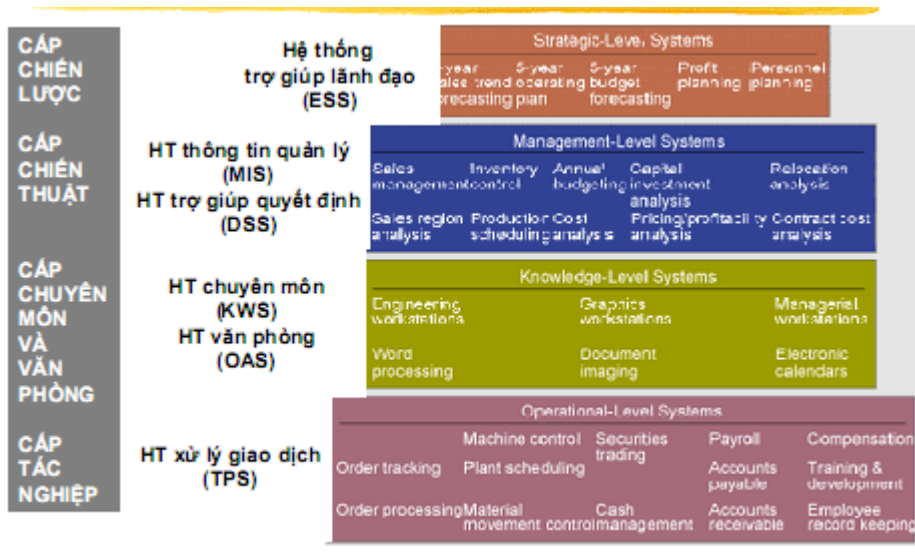
viii. Đổi tên thư mục và tập tin

- Chọn đối tượng muốn đổi tên
- Thực hiện lệnh **File/ Rename** hoặc nhấn phím F2 hoặc R_Click trên đối tượng và chọn mục Rename.
- Nhập tên mới, sau đó gõ Enter để kết thúc.

Ghi chú: với tập tin đang sử dụng thì các thao tác di chuyển, xoá, đổi tên không thể thực hiện được.

ix. Thay đổi thuộc tính tập tin và thư mục

- Nhấn chuột phải lên đối tượng muốn thay đổi thuộc tính và chọn mục **Properties**
- Thay đổi các thuộc tính.
- Chọn **Apply** để xác nhận thay đổi, ngược lại thì nhấn **Cancel**.



Hệ thống xử lý giao dịch (TPS)

Hệ thống xử lý giao dịch là một hệ thống thông tin giúp thi hành và lưu lại các giao dịch thường ngày cần thiết cho hoạt động sản xuất kinh doanh.

Ví dụ: nhập đơn đặt hàng, đặt phòng khách sạn, bảng lương, lưu hồ sơ nhân viên và vận chuyển vật tư.

Đây là HTTTQL ở Cấp tác nghiệp. Trong đó:

- Thu thập: các giao dịch, sự kiện
- Xử lý: cập nhật, sắp xếp, tổng hợp
- Phân phối: các báo cáo chi tiết, danh sách, tóm tắt
- Người dùng: nhân viên tác nghiệp, quản đốc, trưởng nhóm

Hệ thống thông tin văn phòng (OAS)

Hệ thống thông tin văn phòng là hệ thống hỗ trợ các nhân viên văn phòng trong các chức năng phối hợp và liên lạc trong văn phòng.

Đây là HTTTQL ở Cấp chuyên môn và văn phòng. Trong đó:

- Thu thập: văn bản, tài liệu, lịch trình
- Xử lý: quản lý văn bản, lập lịch trình, thông tin liên lạc
- Phân phối: văn bản, lịch biểu, thư điện tử
- Người dùng: nhân viên văn thư, tất cả nhân viên

Hệ thống chuyên môn (KWS)

Hệ thống chuyên môn là hệ thống hỗ trợ lao động có trình độ cao trong công việc chuyên môn hàng ngày của họ.

Đây là HTTTQL ở Cấp chuyên môn và văn phòng. Trong đó:

- Thu thập: các ý tưởng thiết kế, thông số kỹ thuật
- Xử lý: xây dựng mô hình chuyên môn
- Phân phối: bản thiết kế, đồ họa, kế hoạch
- Người dùng: chuyên gia, kỹ thuật viên

Hệ thống trợ giúp ra quyết định (DSS)

Hệ thống trợ giúp ra quyết định hỗ trợ các nhà quản lý ra các quyết định đặc thù, nhanh thay đổi và không có quy trình định trước.

Đây là HTTTQL ở Cấp chiến thuật. Trong đó:

- Thu thập: dữ liệu khối lượng nhỏ
- Xử lý: tương tác
- Phân phối: các báo cáo phân tích, trợ giúp quyết định
- Người dùng: Nhà quản lý bậc trung, chuyên gia

Hệ thống thông tin quản lý (MIS)

Hệ thống thông tin quản lý là hệ thống phục vụ các chức năng lập kế hoạch, giám sát và ra quyết định ở cấp quản lý.

Đây là HTTTQL ở cấp chiến thuật. Trong đó:

- Thu thập: dữ liệu khối lượng lớn, từ Hệ thống xử lý giao dịch
- Xử lý: các quy trình đơn giản
- Phân phối: các báo cáo tổng hợp, tóm tắt
- Người dùng: nhà quản lý bậc trung

Hệ thống trợ giúp lãnh đạo (ESS)

Hệ thống trợ giúp lãnh đạo là môi trường khai thác thông tin tổng thể từ trong và ngoài doanh nghiệp phục vụ việc ra các quyết định đòi sự đánh giá, suy xét và không có quy trình thống nhất.

Đây là HTTTQL ở Cấp chiến lược. Trong đó:

- Thu thập: dữ liệu đã tổng hợp
- Xử lý: tương tác
- Phân phối: các dự báo, phân tích, báo cáo tổng hợp
- Người dùng: lãnh đạo cao cấp



Hệ thống quản lý chuỗi cung ứng (SCM)

Là hệ thống tích hợp giúp quản lý và liên kết các bộ phận sản xuất, khách hàng và nhà cung cấp.



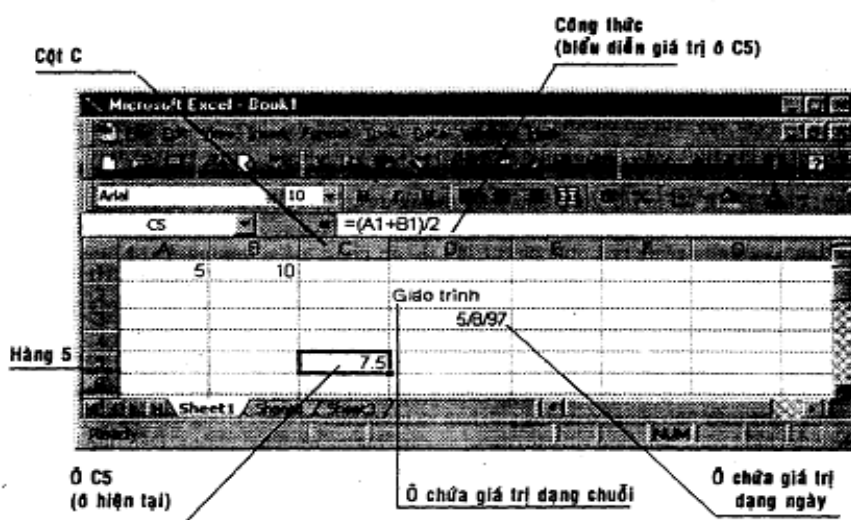
Hệ thống quản lý quan hệ khách hàng (CRM)

Là hệ thống tích hợp giúp quản lý và liên kết toàn diện các quan hệ với khách hàng qua nhiều kênh và bộ phận chức năng khác nhau.



Hệ thống quản lý tri thức (KM)

phần giao của một hàng hoặc một cột được gọi là một ô. Mỗi ô trong bản được xác định bằng một địa chỉ duy nhất tạo bởi số của hàng và chữ cái của cột. Ví dụ, ô nằm ở góc bên trái của bảng là ô A1(cột A hàng 1), ô nằm ngay bên phải là ô B1, kế đến là C1..., ô nằm ngay dưới ô A1 là A2, kế đến là A3,... Đối với một bảng tính rỗng (chưa có dữ liệu) thì tất cả các ô đều rỗng; dĩ nhiên, sau này người dùng sẽ điền nội dung vào các ô. Mỗi ô trong bảng tính có thể chứa một giá trị số, một chuỗi kí tự hoặc một công thức hiển thị mỗi liên hệ giữa các con số trong các ô. Các giá trị số được xem là vật liệu thô dùng để tính toán.



Các con số này có thể biểu diễn cho tiền lương, điểm thi, dữ liệu thời tiết, kết quả bầu cử hoặc bất cứ thứ gì có thể định lượng được. Để giúp người đọc dễ dàng hiểu được ý nghĩa các con số, người ta thường xuyên dùng thêm các chuỗi tiêu đề (label) ở đầu các cột và ở bên trái các hàng, như “Lương tháng”, “ Kết quả học kì I”, “Tốc độ gió trung bình”, và “ Tỷ lệ bầu cử cuối cùng”. Đối với máy tính, các tiêu đề trên chỉ là những chuỗi kí tự vô nghĩa. Một tiêu đề tên là “ Điểm tổng kết” hoàn toàn không yêu cầu máy tính tính toán điểm tổng kết mà hiển thị kết quả tại một ô nào đó; nó chỉ là một dạng hướng dẫn người đọc bảng tính – chứ không phải hướng dẫn máy tính!

Để thực hiện một thao tác tính toán gì đó, như tính điểm trung bình chẳng hạn, chúng ta phải đưa vào một công thức tính toán để hướng dẫn máy tính cách tính ra kết quả cuối cùng. Giả sử, bạn có điểm Toán và điểm Lý, bạn đã đưa điểm Toán (giả sử là 9) vào ô B2 và điểm Lý (giả sử là 7) vào ô B3, và bạn muốn biết kết quả điểm trung bình đó sẽ được hiển thị trong ô B5. bạn sẽ đưa vào ô B5 công thức $= (B2+B3)/2$. Công thức này yêu cầu máy tính cộng con số ở ô B2 và B3, được bao nhiêu chia cho 2 và hiển thị kết quả cuối cùng trong ô B5. Bạn sẽ không thấy công thức nằm ở ô B5 mà chỉ thấy kết quả của nó (là 8). Với máy tính, ý nghĩa của các con số ở ô B2 và B3 không quan trọng, nó chỉ biết tính toán con số trung bình theo công thức đã cho và hiển thị kết quả. Nếu con số ở ô B2 hoặc B3 bị thay đổi thì con số trung bình ở ô B5 cũng sẽ được tính toán lại (chẳng hạn bạn sửa giá trị ô B2 từ 9 thành 5 thì giá trị ở ô B5 sẽ là 6).

Các phần mềm bảng tính khác nhau có thể khác nhau ở một số chức năng hoặc giao diện nhưng đa số đều có những tính năng gần giống nhau. Hiếm có phần mềm bảng tính nào thiếu một trong những chức năng sau đây:

- **Tự động lặp các giá trị, tiêu đề và công thức:** Dữ liệu trong các bảng tính có nhiều điểm bị lặp lại như công thức tính điểm trung bình là giống nhau đối với mọi sinh viên trong cùng một lớp, công thức tính lương nhân viên là không đổi qua nhiều tháng,.. các phần mềm bảng tính thường cung cấp nhiều chức năng giúp đơn giản hóa việc nhập các dữ liệu lặp. Các chức năng này khác nhau tùy theo từng phần mềm bảng tính nhưng tất cả đều có điểm chung là dựa trên sự mở rộng của thao tác cắt - dán (cut – paste) cơ bản. Các phần mềm bảng tính còn khái niệm tham chiếu tương đối đến những ô khác, khi công thức được chép đến một vị trí khác, thì các tham chiếu cũng sẽ đến chỉ đến những ô khác. Chẳng hạn để tính điểm trung bình của nhiều sinh viên, giả sử mỗi sinh viên chỉ có 2 điểm Toán và Lý.

Tất cả điểm của mỗi sinh viên được đặt trên một hàng của bảng tính, trong đó điểm Toán được đặt ở cột A, điểm Lý được đặt ở cột B và điểm trung bình cần tính ở cột C. Giả sử ta có 10 sinh viên, nếu không sử dụng tham chiếu tương đối, bạn sẽ phải nhập 10 lần công thức tính điểm trung bình sẽ là $= (A1 + B1) / 2$. điểm trung bình điểm trung bình của sinh viên thứ 2 là $= (A2 + B2) / 2$, và cứ như thế. Nếu sử dụng tham chiếu tương đối, bạn chỉ cần nhập công thức này cho một lần cho sinh viên 1 (ở ô C1) $= (A1 + B1) / 2$. Trong đó dấu \$ ngụ ý rằng các con số 1 là tham chiếu tương đối theo hàng. Như vậy, khi bạn chép nội dung ô C1 xuống các ô C2, C3, ... C10 thì các công thức sẽ tự động được cập nhật cho phù hợp; cụ thể ở ô C2 công thức sẽ là $= (A2 + B2) / 2$, và cứ thế... Tuy nhiên, do tham chiếu tương đối được dùng quá rộng rãi còn tham chiếu tuyệt đối hầu như không dùng nên các phần mềm bảng tính thường mặc định sử dụng tham chiếu tương đối. Như ở ví dụ tính điểm trung bình trên, ta chỉ cần ghi $= (A1 + B1) / 2$ là đủ.

- **Tự động tính lại:** Tự động tính lại là một trong những chức năng quan trọng của các phần mềm bảng tính. Khi có sự thay đổi xảy ra tại một ô tính, phần mềm tính sẽ tự động tính lại toàn bộ bảng tính (chẳng hạn nhập sai dữ liệu) mà còn cho phép người dùng dò tìm đáp số của bài toán với những dữ liệu vào khác nhau. Đối với các bảng tính lớn, phức tạp, thao tác tự động tính lại có thể rất chậm, vì vậy các phần mềm bảng tính cho phép bạn bật/ tắt chức năng tự động tính lại khi cần thiết.

- **Các hàm thư viện:** với những chiếc máy tính nguyên sơ, người ta phải tính toán căn bậc hai của một con số bằng một loạt các bước buồn chán và dễ nhầm lẫn. Với các máy tính tay hiện đại ngày nay, chỉ cần bấm nút “căn số” là chiếc máy đã tự động làm mọi chuyện để có được căn bậc hai của con số ta đưa vào. Các phần mềm bảng tính có sẵn các hàm thư viện – một khái niệm giống như nút “căn số” trên máy tính tay. Một hàm trong một công thức toán học yêu cầu máy tính thực hiện một loạt thao tác tính toán đã định sẵn. Chẳng hạn công thức $= \text{SQRT}(C5)$ sẽ yêu cầu máy tính tính căn bậc hai của con số nằm ở ô C5. Các phần mềm bảng tính hiện đại đều có một thư viện khổng lồ. Các hàm như Sum, AVERAGE(hoặc AVG), Min, Max là những hàm tính toán đơn giản được sử dụng thường xuyên trong mọi loại bảng tính. Ngoài ra, còn có những hàm rất phức tạp liên quan đến lĩnh vực tài chính, toán học, thống kê... Hàm IF cho phép bảng tính quyết định phải làm gì dựa trên nội dung của một ô nào đó, từ đó cung cấp khả năng suy luận logic. (Ví dụ: IF số giờ làm việc của công nhân ≥ 40 , tăng lương thêm 50 cho công nhân đó). Cũng giống như nút căn bậc hai trên các máy tính tay, các hàm thư viện giúp tiết kiệm thời gian giảm bớt nguy cơ sinh lỗi.

BẢNG ĐIỂM CÁC MÔN HK 1 - HK 2																	
LỚP 25 K19																	
	Hệ số	3	5	5	4	4	4	3	4	3	4	2	2	3	5		
STT	Tên	LSD	A.VĂN	TRIẾT	TCCẤP	ĐL	CS	TKT	KTCT	LKT	TB MÔN	XEPI					
1	BÌNH	5	5	5	5	5	6	5	3	6	2	5	5	5	5	4.745	YẾ
2	CHƯƠNG	4	5	7	5	4	7	8	7	6	4	5	5	6	5	5.588	TE
3	CƯỜNG	5	5	7	4	5	6	7	5	7	7	7	5	5	5	5.667	TE
4	CƯỜNG	3	5	6	4	5	7	4	6	6	4	5	4	6	4	4.98	YẾ
5	ĐÀO	6	7	7	6	7	6	7	8	7	5	5	5	6	6	6.392	TE
6	DŨNG	6	5	6	6	6	8	8	8	6	5	7	7	6	5	6.235	TE
7	DUYÊN	5	5	6	6	6	6	6	5	6	5	3	6	6	5	5.471	TE
8	HẢI	5	5	6	5	5	2	5	5	6	5	5	6	6	5	5.02	TE
9	HIẾU	5	6	6	6	6	7	8	5	6	5	7	6	5	4	5.765	TE

Macro:
 o: Menu các chức năng của phần mềm bảng tính, cũng giống như trong menu u trong một nhà

hàng, thường chỉ liệt kê những chức năng thường dùng nhất. Trong trường hợp, người dùng thường lặp đi lặp lại một thao tác nhất định nào đó, các phần mềm bảng tính sẽ cho phép người dùng “thu” lại các thao tác đó và định nghĩa nó là một macro. Sau này, khi cần thực hiện lại các thao tác đã ghi nhận trong macro, người dùng chỉ cần gọi lại macro đó thay vì phải lặp lại một loạt các thao tác buồn chán. Ta có thể định nghĩa macro bằng ngôn ngữ macro hoặc dùng bộ thu macro dùng để thu lại mọi thao tác đã tiến hành trên bàn phím và chuột.

Bảng tính mẫu: Ngay sau khi đã dùng macro và hàm việc tạo lập một bảng tính phức tạp từ những đống dữ liệu hỗn tạp vẫn còn đây gian nan. Nhiều người dùng đã biết tận dụng lợi điểm của mẫu một bảng tính chỉ gồm các tiêu đề và công thức nhưng không chứa dữ liệu. Khi người dùng sẽ tự điền các dữ liệu cần thiết. Nhờ tính năng tự động tính lại, ngay khi có dữ liệu phần mềm bảng tính sẽ tính toán và cho chúng ta kết quả. Thông thường các phần mềm bảng tính sẽ gắn kèm theo một số bảng tính mẫu; và bán riêng một số bản mẫu khác. Khi không có các bảng tính mẫu người dùng sẽ tự tạo ra nó hoặc nhờ một lập trình viên viết. Dù xuất phát từ đâu thì, một bảng tính mẫu được thiết kế tốt sẽ giúp tiết kiệm được rất nhiều thời gian và công sức.

Liên kết: Khi làm việc người ta sẽ làm việc với nhiều bảng tính liên quan đến nhau cùng lúc nên một thay đổi trong một bảng tính sẽ ảnh hưởng đến một bảng tính khác. Hầu hết các PMBT cho phép bạn tạo ra những liên kết động giữa các bản tính vì vậy khi giá trị của một bảng tính bị thay đổi tất cả các bảng tính liên kết với nó sẽ tự động cập nhật. Một số chương trình như Lotus 1-2-3 có thể tạo được bảng tính 3 chiều bằng cách xếp lớp (stacking) và liên kết với nhiều bảng tính 2 chiều.

Chức năng cơ sở dữ liệu: Các phần mềm cơ sở dữ liệu sẽ được đề cập chi tiết ở chương 8. ở đây chúng ta chỉ cần biết là nhiều phần mềm bảng tính có thể thực hiện được những chức năng cơ sở dữ liệu cơ bản: lưu trữ và truy cập thông tin, tìm kiếm sắp xếp, phát sinh báo cáo, trộn thư (mail merge). Với những chức năng này phần mềm bảng tính có thể đáp ứng những yêu cầu cơ sở dữ liệu phức tạp hơn, phần mềm bảng tính vẫn có thể hữu dụng; nhiều chương trình bảng tính hỗ trợ liên kết tự động hai chiều với các phần mềm cơ sở dữ liệu.

Câu hỏi điều gì sẽ xảy ra nếu...

Một phần mềm bảng tính là một công cụ linh hoạt, nhưng nó đặc biệt hữu dụng cho câu hỏi “điều gì sẽ xảy ra nếu...”; “Điểm trung bình của tôi sẽ ra sao nếu điểm môn A của tôi là”; “điều gì sẽ xảy ra nếu tôi bán giá sản phẩm 100 thay vì 200”. Các phần mềm mô phỏng cho phép bạn thay đổi các con số và ngay lập tức thấy thay đổi của những con số này, và điều này sẽ làm đơn giản hóa quá trình tìm kiếm lời giải đối với những câu hỏi dạng này.

Một vài phần mềm mô phỏng còn có cả một chương trình chuyên giải phương trình. Thay vì bạn phải thay đổi số liệu nhập cho đến khi các công thức cho ra con số mà bạn mong muốn, chương trình cho phép bạn định nghĩa một chương trình, đưa vào kết quả mong muốn và nó sẽ tính toán số liệu cần thiết để tạo ra kết quả mong muốn này. Bạn thử hình dung việc trả lời câu hỏi “Điểm trung bình của tôi sẽ ra sao nếu điểm môn A, môn B là...” sẽ dễ dàng đến mức nào khi có công thức này. Công cụ này còn rất hữu ích để giải những bài toán tối ưu.

Các phần mềm bảng tính tương lai

Với tất cả những gì phần mềm bảng tính đã mang lại cho người dùng, phải chăng nó đã đi đến điểm kết thúc của cuộc cách mạng. Bước tiếp theo là gì ?

Các ấn bản gần đây của Lotus 1-2-3 còn cho phép hỗ trợ người dùng trong công nghệ multimedia. Microsoft Excel là phần mềm bảng tính đã bắt đầu sử dụng công nghệ trí tuệ nhân tạo để hướng dẫn người dùng trong những tác vụ phức tạp. Để giúp người dùng kiểm tra tính toàn vẹn của dữ liệu và công thức của các bảng tính phức tạp, các phần mềm bảng tính trong tương lai có lẽ sẽ cung cấp các chức năng đại loại bộ kiểm hợp lệ validator hoạt động như các bộ kiểm lỗi chính tả và lỗi ngữ pháp trong các trình xử lý từ.

Nhìn xa hơn nữa có lẽ các phần mềm bảng tính sẽ biến mất cùng với rất nhiều các ứng dụng khác. Các phần mềm bảng tính khổng lồ, đầy các chức năng có thể bằng các phần mềm nhỏ hơn để phụ trợ những dạng ứng dụng khổng lồ, bao quát. Người dùng sẽ làm việc với văn bản, các con số và những kiểu dữ liệu khác mà không cần phải nghĩ về những trình xử lý, phần mềm bảng tính riêng biệt. Theo quan điểm của người dùng, dữ liệu chiếm vị trí quan trọng hơn công cụ. Nhưng cho đến lúc điều đó xảy ra, các phần mềm bảng tính vẫn tiếp tục là các công cụ đặc lực cho hàng triệu người dùng trên khắp thế giới.

Đồ thị trong bảng tính: Từ số thành hình vẽ

Đa số phần mềm bảng tính đều có chức năng liên quan đến vẽ đồ thị – một chức năng dùng để chuyển những con số thành hình vẽ, đồ thị một cách tự động. Các chương trình lưu đồ riêng biệt cho phép tạo các biểu đồ từ bất kỳ tập hợp các con số nào, dù được chứa trong các bảng tính làm việc hay không. Quá trình tạo một biểu đồ cũng đơn giản như việc điền một thông tin vào một hộp thoại. Và như vậy thì việc tạo lập các biểu đồ từ con số sẽ trở nên dễ dàng, trực quan hơn.

Sự gia tăng dân số nhanh của một quốc gia sẽ dễ hình dung hơn khi ta trình bày nó dưới dạng một đồ thị ngày càng lên cao dần thay vì chỉ quan sát các con số. Hầu hết các PMBT và chương trình vẽ biểu đồ đều cung cấp nhiều kiểu biểu đồ cơ sở và nhiều chọn lựa để xây dựng đồ thị. Các loại đồ thị khác nhau được dùng không chỉ vì lý do mỹ thuật; mỗi loại đồ thị phù hợp với mỗi kiểu diễn đạt thông tin nhất định nào đó. Đồ thị tròn (pie chart) cho thấy sự tương quan của từng bộ phận với toàn thể. Đồ thị đường (line chart) thường dùng để thể hiện các biến động thông qua thời gian. Đồ thị cột (Bar chart) tương tự đồ thị đường, nhưng phù hợp khi dữ liệu được phân ra thành nhiều loại, nhiều kiểu... Một đồ thị được thiết kế tốt có thể chuyển tải những thông tin rất giá trị, và ngược lại một đồ thị đuwocj thiết kế tồi có thể dẫn đến hiểu sai hoặc thiếu sót.

Trao đổi kinh nghiệm

Tại sao người ta dùng máy tính để thực hiện những thao tác quản lý thông tin trong khi những thao tác này có thể thực hiện được với các sổ sách thông thường? Cơ sở dữ liệu trên máy tính có những ưu điểm vượt trội so với người bạn giấy và bút:

+ Cơ sở dữ liệu làm cho việc lưu trữ một lượng thông tin khổng lồ trở nên dễ dàng. Nếu bạn chỉ có 20 đến 30 đĩa CD, bạn vẫn có thể theo dõi chúng bằng một cuốn sổ. Khi bạn có 2000 đến 3000 đĩa CD (bạn là chủ một cửa hàng bán đĩa CD chẳng hạn!) thì cuốn sổ của bạn sẽ dày cộm và rối tung y như đồng đĩa CD của bạn! Nhưng với một cơ sở dữ liệu trên máy tính, toàn bộ thông tin về đĩa CD của bạn chỉ nằm gọn trên một cái đĩa. Thông tin càng khổng lồ, việc sử dụng một cơ sở dữ liệu càng mang lại nhiều lợi ích.

+ Cơ sở dữ liệu giúp nhanh chóng và mềm dẻo trong việc tra cứu thông tin. Bạn thường phải mất khoảng một phút hoặc hơn để tra một số điện thoại trong một sổ ghi điện thoại, cũng việc đó nhưng với cơ sở dữ liệu chỉ tốn chưa đầy 1 giây. Nếu bạn có đến 200 con số để ghi nhớ mỗi tuần, lợi điểm khi sử dụng Cơ sở dữ liệu đã rõ ràng. Lợi điểm đó còn lớn hơn nữa khi việc tìm kiếm của bạn không hợp với cách tổ chức lưu trữ thông tin của cơ quan bạn. Ví dụ, giả sử bạn thấy một số điện thoại trên mảnh giấy nhỏ, bạn muốn tìm tên và địa chỉ của người có số điện thoại đó. Việc tìm kiếm này sẽ làm bạn mất hàng giờ nếu thông tin trên sổ địa chỉ được sắp xếp theo thứ tự alphabet theo tên. Nếu dùng cơ sở dữ liệu trên máy tính hầu như ngay lập tức, bạn sẽ có kết quả. Và như vậy bạn sẽ không lầy gì làm ngạc nhiên khi tổng đài 116 có thể nhanh chóng cho chúng ta biết tên họ và địa chỉ người có số điện thoại cho trước trong khi họ có đến hàng trăm số điện thoại khác nhau.

+ Cơ sở dữ liệu giúp dễ dàng sắp xếp và tổ chức thông tin, khi có quá nhiều tài liệu hoặc sách vở, bạn thường phải sắp xếp chúng lại theo những trật tự nhất định nào đó. Bạn nên sắp xếp chúng theo tên tác giả, tựa đề, ngày xuất bản hay theo chủ đề? Có rất nhiều thứ sẽ làm bạn phải rất phân vân vì nếu sau này phải sắp xếp lại thì sẽ mất rất nhiều thời gian. Với một cơ sở dữ liệu bạn có thể ngay lập tức chuyển từ kiểu sắp xếp này với kiểu sắp xếp khác khi bạn thích; và bạn sẽ không phải chịu bất cứ sự bực mình nào.

+ Cơ sở dữ liệu giúp in và phân phối thông tin theo nhiều cách, giả sử bạn muốn gửi các lá thư đến hàng trăm người bạn để mời họ đến ăn mừng ngày bạn ra trường. Bạn sẽ phải chỉ đường đi đến nơi của bạn cho từng người bạn ở thành phố khác. Quá mất công? Một cơ sở dữ liệu, khi được dùng với một Trình xử lý từ có thể in những dạng thư từ cá nhân và đưa vào đó những thông tin chỉ dẫn cho những người cần nó, và in địa chỉ của từng người lên bì thư chỉ tốn chút thời gian và hầu như không bao giờ sai.

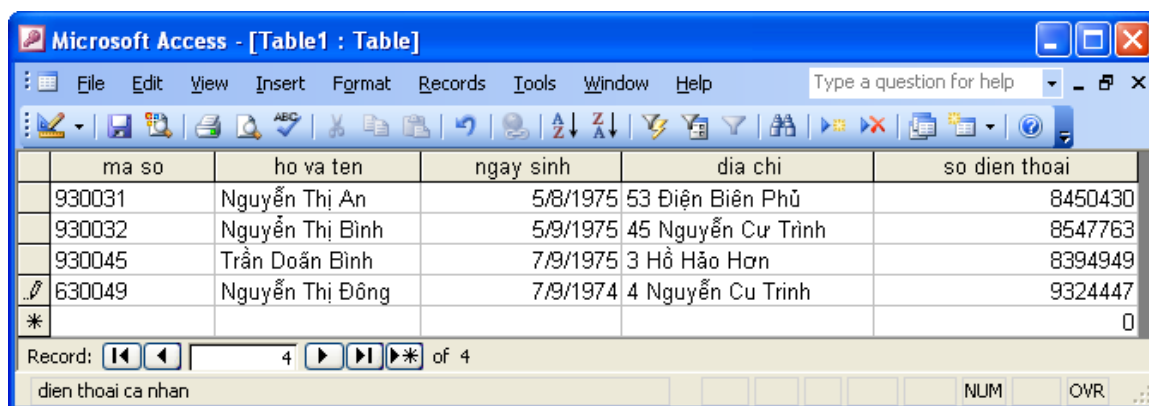
Bên trong cơ sở dữ liệu

Hiểu một cách nôm na, một cơ sở dữ liệu là một tập hợp thông tin được lưu trữ trên máy tính theo một dạng tổ chức nào đó, và một chương trình cơ sở dữ liệu là một công cụ phần mềm để tổ chức, lưu trữ và truy cập các thông tin đó. Có nhiều loại chương trình thỏa mãn định nghĩa chung này, từ các chương trình tra cứu sổ địa chỉ đến các hệ thống theo dõi hàng tồn kho khổng lồ. Chúng ta sẽ bàn về sự khác nhau giữa các kiểu chương trình cơ sở dữ liệu sau, nhưng bây giờ chúng ta sẽ xem chúng như là một.

Một cơ sở dữ liệu được hình thành từ các file. Một File chứa một tập các thông tin có liên quan; giống như một ngăn kéo đựng hồ sơ thông thường. nếu cơ sở dữ liệu được dùng để ghi nhận thông tin về các hàng bán cho một công ty, mỗi file có thể sẽ chứa dữ liệu hàng bán có liên quan cho mỗi năm. Đối với các cơ sở dữ liệu địa chỉ, các file riêng có thể sẽ lưu trữ các thông tin cá

nhân và đối tác. Tùy theo người thiết kế cơ sở dữ liệu để quyết định có nên để các thông tin thuộc các dạng khác nhau vào các file trên máy tính khác nhau hay không.

Từ file thường gây nhầm lẫn do tính nhiều nghĩa của nó. Một đĩa cứng có thể chứa chương trình ứng dụng, chương trình hệ thống, chương trình tiện ích và các văn bản, tất cả mọi thứ, nếu đứng từ góc nhìn máy tính đều là các file. Nhưng đối với những người sử dụng cơ sở dữ liệu, từ file thường mang nghĩa là một phần của cơ sở dữ liệu. Trong chương này, từ file mang ý nghĩa là các file dữ liệu tạo bởi một phần mềm cơ sở dữ liệu.



ma so	ho va ten	ngay sinh	dia chi	so dien thoai
930031	Nguyễn Thị An	5/8/1975	53 Điện Biên Phủ	8450430
930032	Nguyễn Thị Bình	5/9/1975	45 Nguyễn Cư Trinh	8547763
930045	Trần Doãn Bình	7/9/1975	3 Hồ Hảo Hớn	8394949
630049	Nguyễn Thị Đông	7/9/1974	4 Nguyễn Cư Trinh	9324447
*				0

Một file cơ sở dữ liệu bao gồm nhiều bản ghi (record). Một bản ghi là thông tin liên quan đến một người, một sản phẩm hoặc một sự kiện nào đó. Trong cơ sở dữ liệu lưu các thẻ thư viện, một record tương đương với một thẻ thư viện. Trong một cơ sở dữ liệu chứa địa chỉ, một record chứa thông tin về một người. Một cơ sở dữ liệu liên quan đến đĩa CD sẽ có một record cho mỗi một đĩa CD.

Mỗi một đoạn thông tin riêng rẽ trong một record được gọi là một trường (field). Một record trong cơ sở dữ liệu thư viện sẽ có những field cho tác giả, tựa đề sách, nhà xuất bản, địa chỉ, ngày tháng và mã số sách. Cơ sở dữ liệu về CD có thể chia record thành các field như tựa đề CD, tác giả....

Kiểu thông tin một field được xác định bằng kiểu field(field type). Ví dụ: field tên tác giả trong cơ sở dữ liệu thư viện có kiểu chữ (text field) để có thể chứa được các chữ cái tên của tác giả. Field cho biết số bản sách hiện có thể sẽ được định nghĩa là field kiểu số để nó chỉ có thể chứa các con số. Field cho biết ngày nhập sách vào thư viện sẽ là field kiểu ngày. Bên cạnh những kiểu field cơ bản này, nhiều chương trình cơ sở dữ liệu cho phép các field chứa các hình ảnh, âm thanh, và thậm chí cả những đoạn video. Các field tính toán (computed) chứa các công thức tương tự như công thức trong các phần mềm bảng tính; chúng hiển thị giá trị được tính toán từ các giá trị của những field khác. Ví dụ, một field tính toán có tên là DTB sẽ chứa công thức tính điểm trung bình với các điểm nằm các field khác.

Hầu hết các phần mềm cơ sở dữ liệu cho phép bạn xem dữ liệu theo nhiều cách khác nhau, thường là kiểu xem đọc(form view) chỉ cho thấy mỗi record một lần và kiểu xem ngang (list view) hiển thị nhiều record trong một danh sách trông giống như một bảng tính. Trong bất cứ kiểu xem nào, các field cũng có thể được sắp xếp lại và không làm thay đổi dữ liệu bên dưới.

Một khi cấu trúc của cơ sở dữ liệu đã được định nghĩa, việc đưa thông tin vào trở nên dễ dàng; chỉ là vấn đề gõ phím mà thôi. Và thậm chí việc gõ phím cũng sẽ không cần thiết nếu dữ liệu đã được lưu đâu đó trên máy tính. Hầu hết các chương trình cơ sở dữ liệu có thể dễ dàng hiểu và

chấp nhận dữ liệu dưới dạng các file văn bản tạo bởi các trình xử lý từ, phần mềm bảng tính hoặc những phần mềm khác. Khi thông tin bị thay đổi hoặc khi phát hiện lỗi, các record có thể được điều chỉnh, thêm hoặc xóa.

Phần thử thách của việc sử dụng một Cơ sở dữ liệu là truy cập thông tin theo cách nào sao cho ít tốn thời gian và phù hợp. Các thông tin sẽ mất nhiều giá trị nếu ta không truy cập đến nó được. Một cách để tìm thông tin là dò (browse) từng record của file Cơ sở dữ liệu giống như dò trên giấy tờ. Hầu hết các chương trình Cơ sở dữ liệu cung cấp được các lệnh bàn phím, các nút trên màn hình và những công cụ khác để qua lại nhanh chóng giữa các record. Những kiểu làm việc này chẳng mang lại lợi lộc gì so với dung giấy; thật là kinh khủng khi gặp phải những file quá lớn.

Nhưng may thay, đa số các chương trình Cơ sở dữ liệu thường có nhiều lệnh và chức năng làm cho việc tìm kiếm thông tin của bạn muốn trở nên dễ dàng hơn.

Truy vấn cơ sở dữ liệu (query)

Một phương pháp thay vì dò từng record là hỏi Cơ sở dữ liệu về những thông tin nào đó. Trong các thuật ngữ liên quan đến Cơ sở dữ liệu, một yêu cầu về thông tin được gọi là query. Một query có thể là một yêu cầu tìm kiếm đơn giản một record nào đó (chẳng hạn như tìm một sinh viên có mã số ABCD01) hoặc là tìm tất cả record thỏa mãn một tập điều kiện nào đó (chẳng hạn, tìm tất cả sinh viên có điểm trung bình dưới 5 và còn nợ trên 2 môn học). Một khi đã chọn xong một nhóm các record, bạn có thể dò tìm trên nhóm record này, tạo ra 1 bảng báo cáo; nói chung là bất cứ thao tác nào bạn có thể thực hiện trên một file thì bạn cũng có thể thực hiện trên 1 nhóm các record có được từ 1 query.

Sắp xếp dữ liệu

Thỉnh thoảng cũng phải sắp xếp lại các record để sử dụng dữ liệu hữu hiệu hơn. Giả sử danh sách sinh viên nên sắp xếp theo alphabet để dễ tra cứu theo tên, nhưng để phân loại kết quả học tập lại cần phải sắp xếp danh sách này theo điểm trung bình. Lệnh sort cho phép bạn sắp xếp các record theo trình tự alphabet hoặc theo giá trị số tùy theo kiểu dữ liệu của file được sắp xếp.

In các báo cáo, tiêu đề và các dạng thư từ

Để hiển thị thông tin lên màn hình, các chương trình Cơ sở dữ liệu có thể tạo ra rất nhiều báo cáo – một danh sách các record được chọn lựa và sắp xếp theo một trình tự dễ đọc. Đa số các bảng báo cáo thương mại đều sắp xếp theo dữ liệu thành các bác với càng hàng là các record và các cột cho các file được chọn; ngoài ra chúng thường có các hàng tổng cộng chứa tổng và trung bình giá trị của một nhóm record.

Các chương trình dữ liệu còn có thể được dùng để tạo ra các nhãn thư từ và các dạng thư từ theo ý muốn. Nhiều chương trình Cơ sở dữ liệu không thực sự in ra các lá thư; chúng chỉ đơn giản cung cấp các dữ liệu cần thiết cho các Trình xử lý từ có khả năng trộn thư (mail merge) để chúng tiến hành công việc in thư.

Các query phức tạp

Các query có thể đơn giản hay phức tạp, nhưng theo cách nào đi nữa kết quả vẫn phải chính xác và không được mờ nghĩa. Với các Cơ sở dữ liệu phù hợp, các query có được xây dựng để:

- Trong một Cơ sở dữ liệu về bệnh nhân trong một bệnh viện, xác định tên và vị trí của tất cả bệnh nhân trên tầng 5 và 6 của bệnh viện.

- Trong Cơ sở dữ liệu của lịch trình chuyến bay, xác định lịch trình rẻ tiền nhất để bay từ Boston đến San Francisco vào trưa thứ 3.

- Trong một Cơ sở dữ liệu về chính trị, xác định tất cả cử tri đã đóng góp tài chính cho cuộc vận động bầu cử và tất cả những cử tri đã bày tỏ ý kiến về bộ luật lao động mới của chính phủ.

Những điều này đều là những mục tiêu đúng đắn của query, nhưng chưa được diễn tả theo một ngôn ngữ chung mà tất cả chương trình Cơ sở dữ liệu có thể hiểu được. Phương pháp chính xác để thực hiện một query phụ thuộc vào giao diện người dùng của Cơ sở dữ liệu. Hầu hết chương trình cho phép người dùng chỉ ra các luật tìm kiếm bằng cách điền thông tin vào các hộp thoại. Một vài phần mềm yêu cầu người dùng gõ các yêu cầu bằng cách dùng một ngôn ngữ đặc biệt, ngôn ngữ của query, chính xác hơn tiếng Anh. Ví dụ để xác định tất cả record của những người nam giới trong độ tuổi từ 18 đến 35, bạn sẽ gõ

Select * from Population Where Sex = M and Age > 18 and Age < 35

(Chọn tất cả công dân trong file <Danh sách công dân> sao cho giới tính là nam và tuổi từ 18 đến 35)

Nhiều chương trình Cơ sở dữ liệu có sử dụng ngôn ngữ lập trình để các query đưa vào chương trình và được thực hiện tự động khi chương trình được thi hành. Dù chi tiết của quá trình này là khác nhau, những điểm cơ bản là thống nhất với mọi chương trình.

Hầu hết các chương trình quản trị Cơ sở dữ liệu đều hỗ trợ SQL (một ngôn ngữ chuẩn để diễn đạt các câu query phức tạp, ví dụ ở trên dùng SQL). Vì SQL làm việc trên nhiều hệ quản trị Cơ sở dữ liệu khác nhau, các lập trình viên và những người dùng tinh tế không cần phải học lại khi họ làm việc với các hệ phân cứng và phần mềm khác nhau. Người dùng thường được tách biệt khỏi sự phức tạp của ngôn ngữ query phức tạp bằng các giao diện người dùng bằng đồ họa cho phép thực hiện các query bằng trỏ và click.

Các chương trình Cơ sở dữ liệu chuyên biệt

Các phần mềm Cơ sở dữ liệu chuyên biệt được lập trình sẵn cho một kiểu dữ liệu xác định và một cách truy cập xác định. Các Cơ sở dữ liệu về CD-ROM được sử dụng trong các thư viện là một ví dụ của các chương trình Cơ sở dữ liệu chuyên biệt. Người dùng của các Cơ sở dữ liệu chuyên biệt không cần phải định nghĩa các cấu trúc file hoặc định nghĩa các màn hình giao diện vì những chi tiết này đã được người thiết kế phần mềm làm sẵn. Trên thực tế, nhiều chương trình Cơ sở dữ liệu chuyên biệt không được bán dưới hình thức một Cơ sở dữ liệu; chúng được đặt tên chính xác hơn, phản ánh mục đích sử dụng của chúng .

Ví dụ:

- a. Danh bạ điện thoại điện tử có thể chứa hàng triệu tên và số điện thoại trên một đĩa CD-ROM. Dùng một danh bạ điện thoại điện tử của Mỹ quốc ngay cả khi bạn không biết họ ở đâu. Bạn có thể tra tên riêng của một người nếu bạn có số điện thoại hoặc địa chỉ. Bạn có thể in một bảng báo cáo cho biết tất cả các nha sĩ trong bất kỳ thành phố nào. Sau đó dùng thêm một Cơ sở dữ liệu nữa, một bảng đồ điện tử, bạn sẽ xác định được chính xác các vị trí các vị trí bạn quan tâm.

Ví dụ, xét bài toán quản lý thông tin sinh viên. Bạn cũng sẽ dễ thấy cách dùng cơ sở dữ liệu để chứa các thông tin: một file chứa một record cho một sinh viên, với các field cho tên, mã số sinh viên, địa chỉ, số điện thoại và nhiều thứ khác. Thông thường, một file sẽ không thể chứa hết mọi thông tin của sinh viên.

Hầu hết các trường chọn cách dùng vài file để chứa thông tin sinh viên: một cho các dữ liệu về tài chính, một theo dõi việc đăng ký môn học, điểm và nhiều thứ khác. Mỗi file chứa một record cho một sinh viên. Bên cạnh đó, nhà trường còn phải theo dõi danh sách các sinh viên đăng ký vào các lớp với một record cho một lớp và các field cho các thông tin trên mỗi sinh viên đăng ký vào lớp. 3 file này được sắp xếp như sau:

Thông tin sinh viên

Mã số sinh viên

Tên

Địa chỉ

Chuyên ngành

...

Học kỳ 1

Trung bình

Số tín chỉ đã đạt

...

Học kỳ 2

Trung bình

Số tín chỉ đã đạt

Thông tin tài chính

Mã số sinh viên

Tên

Địa chỉ

Chuyên ngành

...

Học phí

Phí ký túc xá

Học bổng

Danh sách lớp

Mã môn học

Phòng học

GV hướng dẫn

Tổng số sinh viên

...

Sinh viên 1

MSSV

Họ tên

Ngành

...

Sinh viên 2

MSSV

Họ tên

Ngành

Trong cơ sở dữ liệu ở trên, mỗi file trong 3 file đều chứa những thông tin cơ bản về mỗi sinh viên (họ tên, địa chỉ). Kiểu dư thừa dữ liệu này không chỉ làm tốn không gian lưu trữ mà còn làm khó khăn trong việc đảm bảo các thông tin sinh viên là chính xác và cập nhật. Nếu một sinh viên được chuyển sang lớp khác, 3 file phải được cập nhật lại để phản ánh sự thay đổi này. Càng thay đổi nhiều, khả năng sai sót càng lớn.

Với một hệ quản trị cơ sở dữ liệu, không cần phải chứa tất cả thông tin này trong mỗi file. Cơ sở dữ liệu có thể dùng một file chứa các thông tin cơ sở về sinh viên – những thông tin duy nhất cho mỗi sinh viên. Vì các thông tin này được chứa trên một file riêng biệt nên nó không cần đưa vào các file theo dõi tài chính, file theo dõi điểm, và lớp hay bất kỳ một file nào khác. Các mã số sinh viên được đưa vào trong các file hoạt động như là một key file; dùng để lấy ra các thông tin sinh viên liên quan trong file sinh viên khi cần thiết ở một nơi nào khác. Field mã số sinh viên được dùng chung trong tất cả các file dùng dữ liệu từ file này. Nếu sinh viên đổi nhà, việc thay đổi địa chỉ chỉ cần ghi nhận ở một chỗ. Các cơ sở dữ liệu được tổ chức theo kiểu này được gọi là Cơ sở dữ liệu quan hệ.

Theo dõi học tập

Mã số sinh viên

...

Học kỳ 1

Điểm TB

...

Học kỳ 2

Thông tin sinh viên

Mã số sinh viên

Tên

Địa chỉ

Chuyên ngành

...

Học kỳ 1

Trung bình

Số tín chỉ đã đạt

...

Học kỳ 2

Trung bình

Số tín chỉ đã đạt

Danh sách lớp

Mã môn học

Phòng học

GV hướng dẫn

Tổng số sinh viên

...

Sinh viên 1

MSSV

Họ tên

Ngành

...

Sinh viên 2

MSSV

Họ tên

Ngành

Điều gì làm cho một cơ sở dữ liệu có tính quan hệ?

Đối với hầu hết người dùng, một chương trình cơ sở dữ liệu quan hệ là chương trình cho phép các file có liên hệ với nhau để các thay đổi trên một file sẽ được tự động phản ánh trên những file khác. Đối với các lập trình viên, từ cơ sở dữ liệu quan hệ có một định nghĩa kỹ thuật liên quan đến cấu trúc bên dưới của dữ liệu và các luật xác định cách thao tác dữ liệu.

Cấu trúc của một cơ sở dữ liệu quan hệ dựa trên mô hình quan hệ (relational model).

Một mô hình toán học phối hợp dữ liệu giữa các mảng. Những hệ quản trị cơ sở dữ liệu loại khác sẽ dựa trên những mô hình lý thuyết khác với những lợi điểm và yếu điểm kỹ thuật riêng. Nhưng những hệ quản trị cơ sở dữ liệu chính yếu được dùng ngày nay, kể cả những hệ quản trị cơ sở dữ liệu dựa trên máy PC cũng dùng mô hình quan hệ. Vì vậy, từ góc nhìn của một người dùng trung bình, sự phân biệt giữa định nghĩa kỹ thuật và phổ thông của từ quan hệ có vẻ hoàn toàn lý thuyết.

Tính nhiều mặt của cơ sở dữ liệu

Các cơ sở dữ liệu không lồ có thể chứa hàng trăm file với vô số mối quan hệ bên trong. Kiểu mê cung thông tin này có thể rất khó hiểu đối với người dùng nếu họ bị buộc phải làm việc trực tiếp với nó. May thay, một hệ quản trị cơ sở dữ liệu có thể “che” người dùng khỏi những công việc phức tạp bên trong của hệ thống, chỉ cung cấp thông tin và các lệnh cần thiết để hoàn thành công việc. Trên thực tế, một cơ sở dữ liệu tốt sẽ có nhiều “bộ mặt” khác nhau đối với những người dùng khác nhau.

Nhân viên bán lẻ không cần (và nhiều lúc không được quyền) truy cập đến mọi thông tin trong cơ sở dữ liệu được lưu trữ của công ty; họ chỉ cần ghi nhận các giao dịch mua bán từ các máy tính trạm đặt ở điểm bán lẻ. Cơ sở dữ liệu được thiết kế cho những người bán lẻ thông thường chỉ cần một giao diện đơn giản, cung cấp cho người bán các thông tin cần thiết giúp họ thực hiện việc bán hàng. Nhà quản lý kế toán, những chuyên gia truy cập dữ liệu và các khách hàng sẽ nhìn cơ sở dữ liệu dưới những góc nhìn khác nhau vì họ cần làm việc với những thông tin khác nhau.

Xu hướng client/server

Các tiến bộ trong hai thập kỷ vừa qua đã thay đổi cách hầu hết các tổ chức làm việc với dữ liệu. Các chương trình quản lý file đầu tiên chỉ có thể thực hiện xử lý theo lô (batch processing), yêu cầu người dùng tích trữ các giao dịch và đưa chúng vào máy tính một lần. Những hệ thống xử lý theo lô này không thể cũng cấp các câu trả lời ngay lập tức như chúng ta vẫn thấy ngày nay. Các câu hỏi đại loại như “số dư của tài khoản của tôi là bao nhiêu?” hoặc là “có chuyến bay nào đến Denver thứ 3 tới không?” sẽ được trả lời bằng cách “chờ đã, những thông tin này sẽ được cập nhật vào tối nay, ngày mai chúng tôi sẽ cho bạn biết! “

Các ổ đĩa, bộ nhớ và những phần mềm tinh tế và rẻ tiền ngày nay đã cho xử lý tương tác (interactive processing) thay thế xử lý theo lô đối với hầu hết ứng dụng. Người dùng bây giờ có thể tương tác với dữ liệu thông qua các trạm làm việc, xem và thay đổi giá trị ngay lập tức theo thời gian thực (nghĩa là sự thay đổi sẽ được cập nhật và những người khác sẽ lập tức thấy sự thay đổi này). Xử lý theo lô vẫn được dùng để in các hóa đơn và báo cáo định kỳ và để tạo các

Được biểu diễn bởi hình ovan có ghi chữ bên trong như:

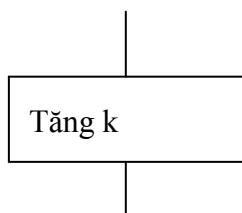


Các nút trên còn được gọi là nút đầu và nút cuối của lưu đồ.

b. Nút thao tác:

Là một hình chữ nhật có ghi các lệnh cần thực hiện.

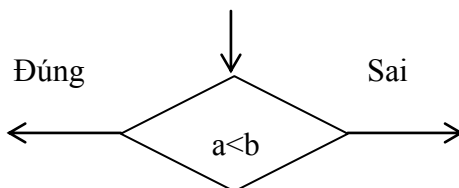
Ví dụ:



c. Nút điều kiện:

Thường là một hình thoi có ghi điều kiện cần kiểm tra. Trong các cung nối với nút này có 2 cung ra chỉ hướng đi theo 2 trường hợp: điều kiện đúng và điều kiện sai.

Ví dụ:



d. Cung:

Là các đường nối từ nút này đến nút khác của lưu đồ.

Hoạt động của thuật toán theo lưu đồ được bắt đầu từ nút đầu tiên. Sau khi thực hiện các thao tác hoặc kiểm tra điều kiện ở mỗi nút thì bộ xử lý sẽ theo một cung đến một nút khác. Quá trình thực hiện thuật toán sẽ dừng khi gặp nút kết thúc hay nút cuối.

Ví dụ:

1: Tìm giá trị lớn nhất của một dãy số nguyên có N số

- Đầu vào: số số nguyên dương N và N số nguyên a_1, a_2, \dots, a_N
- Đầu ra: số nguyên lớn nhất của dãy a_k , k trong khoảng $[1 \dots N]$

Ý tưởng

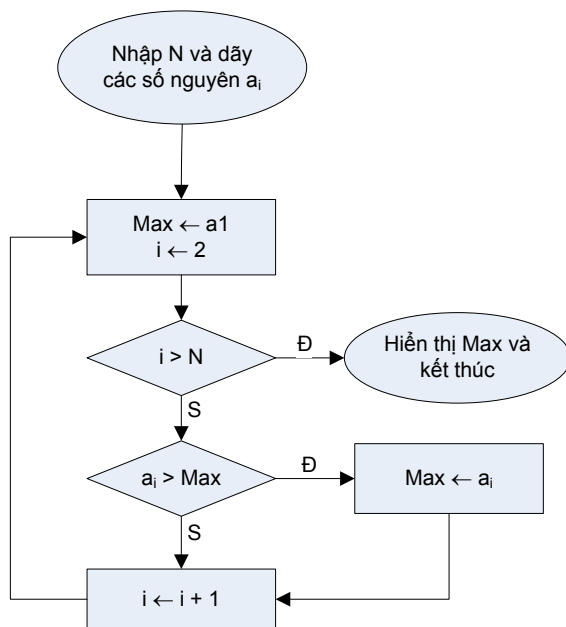
- Khởi tạo giá trị $Max = a_1$

- Lần lượt so sánh Max với $i=2, 3, \dots, N$, nếu $a_i > \text{Max}$ ta gán giá trị mới cho Max.

Thuật toán

- B1: Nhập dãy số a_i .
- B2: $\text{Max} \leftarrow a_1$.
- B3: Nếu $i > N$, thuật toán kết thúc và Max là giá trị lớn nhất của dãy cần tìm
- B4: Nếu $a_i > \text{Max}$, gán a_i cho Max.
- B5: Tăng i lên 1 đơn vị.
- B6: Quay lên b4.
- B7: Kết thúc.

Vì thuật toán là chỉ ra dãy thao tác và trình tự thao tác để đạt mục đích và dùng cho người dùng con người, nên ngoài cách liệt kê trên, người ta có thể dùng sơ đồ khối để minh họa (biểu diễn). Với thuật toán trên, cách biểu diễn theo sơ đồ khối như sau:



Qui ước:

- Hình thoi biểu diễn thao tác so sánh
- Hình chữ nhật thể hiện 1 thao tác tính toán đơn giản hay phức tạp
- Hình ô van thể hiện thao tác bắt đầu hay kết thúc.

2: Sắp xếp bằng phương pháp trao đổi (Exchange Sort)

- Đầu vào: Dãy A gồm N số nguyên a_1, a_2, \dots, a_N
- Đầu ra: Dãy A được sắp lại theo thứ tự không giảm

Ý tưởng

- Với mỗi cặp số liên tiếp trong dãy , nếu số trước không lớn hơn số sau ta đổi chỗ chúng cho nhau.
- Việc đó được lặp cho đến khi không có sự đổi chỗ nào cho nhau.

Thuật toán

- B1: Nhập số N và dãy số a_1, a_2, \dots, a_N
- B2: $M \leftarrow N$.
- B3: Nếu $M < 2$ thì thuật toán kết thúc và hiển thị dãy đó.
- B4: $M \leftarrow M - 1, i \leftarrow 0$.
- B5: Tăng i lên 1 đơn vị.
- B6: Nếu $i > M$ thì quay lại bước 3.
- B7: Nếu $a_i > a_{i+1}$ thì trao đổi hai số đó cho nhau .
- B8: Quay lên bước 5.

Chú ý:

- Thuật toán này tạo ra sau mỗi lượt sắp một phần tử đúng vị trí và phần tử này không còn tham gia vào quá trình sắp nữa. Nó giống như bọt nước nổi lên mặt nước: bóng nhẹ sẽ được đẩy dần lên trên. Cũng chính vì thế mà sắp xếp trao đổi còn có tên “*nổi bọt*” (Bubble Sort).
 - Quá trình được lặp lại với dãy sau khi đã bỏ phần tử cuối dãy, do vậy lúc đầu M được gán với giá trị N và dừng khi $M < 2$.
 - Trong thuật toán trên, i được khởi tạo giá trị 0 và tiến tới $M + 1$
- Với thuật toán trên, cách biểu diễn theo sơ đồ khối như sau:

thực hiện thuật toán trong trường hợp tốt nhất đối với dữ liệu có dữ liệu nhập có độ lớn n . Tương tự ta cũng thường đề cập đến thời gian thực hiện thuật toán lớn nhất đối với các dữ liệu nhập có độ lớn n và gọi là thời gian thực hiện thuật toán trong trường hợp xấu nhất đối với dữ liệu có độ lớn n . Ngoài ra, đối với thuật toán có dữ liệu nhập có độ lớn n trong một tập hữu hạn nào đó, ta còn muốn tính ra thời gian trung bình để thực hiện thuật toán.

Ví dụ: Thuật toán tìm giá trị lớn nhất trong dãy gồm n số nguyên. (đã nêu trong các phần trước)

Trong thuật toán này nếu ta xem thời gian thực hiện thuật toán là số lần thực hiện phép so sánh hay phép gán thì thời gian thực hiện thuật toán trong trường hợp xấu nhất là:

$$t(n) = 1 + 2(n-1) = 2n + 1$$

Và thời gian thực hiện thuật toán trong trường hợp lớn nhất là:

$$T(n) = 1 + (n-1) = n$$

Thông thường ta ít chú ý tới con số chính xác về thời gian thực hiện thuật toán trong trường hợp xấu nhất và trong trường hợp tốt nhất. Điều mà chúng ta thường quan tâm hơn khi đánh giá độ phức tạp thời gian của thuật toán là mức độ tăng lên của thời gian thực hiện thuật toán khi độ lớn của dữ liệu nhập tăng lên. Chẳng hạn, một thuật toán đang được xem xét nào đó có thời gian thực hiện trong trường hợp xấu nhất và tốt nhất lần lượt là:

$$t(n) = 20n^2 + 5n + 1$$

$$T(n) = n^2 + 10n + 1$$

Như thế nếu như n rất lớn thì ta có thể xấp xỉ $t(n)$ và $T(n)$ với $20n^2$ và n^2 . Có thể nói rằng $t(n)$ tăng giống như n^2 khi n tăng, hay nói $t(n)$ có “cấp cao nhất” là n^2 và ta viết $t(n) = O(n^2)$.

Tương tự $T(n) = O(n^2)$. Cuối cùng ta nói độ phức tạp của thuật toán là $O(n^2)$ (độ phức tạp của thuật toán có cấp cao nhất là n^2).

Định nghĩa: Cho 2 hàm f và g có miền xác định trong tập số tự nhiên N . Ta viết:

$$f(n) = O(g(n))$$

Và nói $f(n)$ có cấp cao nhất là $g(n)$ khi có một hằng số dương C sao cho:

$$|f(n)| \leq Cg(n)$$

Với hầu hết n thuộc miền xác định của các hàm f và g . Từ “hầu hết” ở đây ý nói là “với mọi chỉ trừ một số hữu hạn”.

Định lý. Nếu $f(n)$ là một đa thức bậc k theo n thì ta có $f(n) = O(n^k)$.

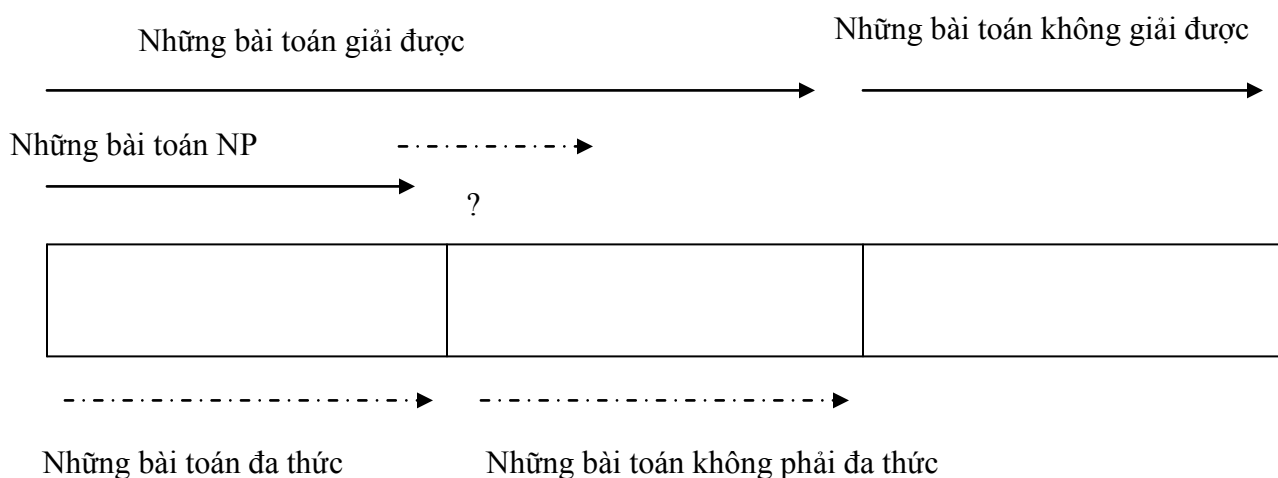
Liên quan đến độ phức tạp của thuật toán ta có một số thuật ngữ thông dụng thường dùng trong một số trường hợp sau đây:

- Độ phức tạp hằng: $O(1)$
- Độ phức tạp logarith: $O(\log n)$
- Độ phức tạp tuyến tính: $O(n)$

Ví dụ: bài toán người bán hàng rong?

Ta gọi một bài toán mà có thể giải được trong thời gian đa thức bằng một giải thuật không tự quyết là bài toán đa thức không tự quyết hay bài toán NP. Dĩ nhiên lớp bài toán NP chứa những bài toán đa thức thực sự, bởi vì nếu một bài toán có thể giải bằng giải thuật tự quyết trong thời gian đa thức, lúc đó chắc chắn nó được giải trong thời gian đa thức bằng một giải thuật không tự quyết. Hơn nữa có những gợi ý rằng những bài toán NP mà không phải bài toán đa thức. (ví dụ bài toán người bán hàng rong). Phân loại bài toán NP hiện vẫn còn là một điều bí mật trong ngành máy tính ngày nay.

Tóm lại, ta thấy rằng những bài toán được phân loại có thể giải quyết (có một lời giải giải thuật) hay không thể giải (không có lời giải giải thuật) như hình sau:



Hơn nữa, nội tại trong lớp của bài toán có thể giải có 2 lớp con: tập hợp các bài toán đa thức được xem là có lời giải thực tế; tập hợp bài toán không đa thức mà lời giải của nó được xem là thực tế chỉ cho những số liệu đưa vào có chọn lựa cẩn thận và tương đối nhỏ. Cuối cùng, có những bài toán NP được xếp vào lớp bài toán có thể giải và chứa những bài toán đa thức. Có hay không những lời giải đa thức đúng đắn cho tất cả các bài toán NP vẫn còn là câu hỏi mở.

Khai báo tệp tiêu đề

```
#include
```

Định nghĩa kiểu dữ liệu

```
typedef ...
```

Khai báo các hàm nguyên mẫu

Khai báo các biến toàn cục

Định nghĩa hàm **main()**

```
main()
```

```
{
```

```
    ...
```

```
}
```

Định nghĩa các hàm đã khai báo nguyên mẫu

- Phần 1: Phần khai báo các tệp tiêu đề. Phần này có chức năng thông báo cho chương trình dịch biết là chương trình có sử dụng những thư viện nào (mỗi tệp tiêu đề tương ứng với một thư viện).
- Phần 2: Định nghĩa các kiểu dữ liệu mới dùng cho cả chương trình.
- Phần 3: Phần khai báo các hàm nguyên mẫu. Phần này giúp cho chương trình dịch biết được những thông tin cơ bản (gồm tên hàm, danh sách các tham số và kiểu dữ liệu trả về) của các hàm sử dụng trong chương trình.
- Phần 4: Phần khai báo các biến toàn cục.
- Phần 5: Phần định nghĩa hàm **main()**. Hàm **main()** là một hàm đặc biệt trong C. Khi thực hiện, chương trình sẽ gọi hàm **main()**, hay nói cách khác chương trình sẽ bắt đầu bằng việc thực hiện các lệnh trong hàm **main()**. Trong hàm **main()** ta mới gọi tới các hàm khác.
- Phần 6: Phần định nghĩa các hàm đã khai báo nguyên mẫu. Ở phần 3 ta đã khai báo nguyên mẫu (*prototype*) của các hàm, trong đó chỉ giới thiệu các thông tin cơ bản về hàm như tên hàm, danh sách các tham số và kiểu dữ liệu trả về. Nguyên mẫu hàm không cho ta biết cách thức cài đặt và hoạt động của các hàm. Ta sẽ làm việc đó ở phần định nghĩa các hàm.

Trong 6 phần trên, thì phần 5 định nghĩa hàm **main()** bắt buộc phải có trong mọi chương trình C. Các phần khác có thể có hoặc không.

Dưới đây là ví dụ một chương trình viết trên ngôn ngữ C.

```
1. // Chương trình sau sẽ nhập vào từ bàn phím 2 số nguyên
2. // và hiển thị ra màn hình tổng, hiệu tích của 2 số nguyên vừa nhập vào
3. #include <stdio.h>
4. #include <conio.h>
5. void main()
6. {
7.     // khai báo các biến trong chương trình
8.     int a, b
9.     int tong, hieu, tich;
10.    // Nhập vào từ bàn phím 2 số nguyên
11.    printf("\n Nhập vào số nguyên thứ nhất: ");
12.    scanf("%d",&a);
13.    printf("\n Nhập vào số nguyên thứ hai: ");
14.    scanf("%d",&b);
15.    // Tính tổng, hiệu, tích của 2 số vừa nhập
16.    tong = a+b;
17.    hieu = a - b;
18.    tich = a*b;
19.    // Hiển thị các giá trị ra màn hình
20.    printf("\n Tổng của 2 số vừa nhập là %d", tong);
21.    printf("\n Hiệu của 2 số vừa nhập là %d", hieu);
22.    printf("\n Tích của 2 số vừa nhập là %d", tich);
23.    // Cho người sử dụng ấn phím bất kỳ để kết thúc
24.    getch();
25. }
```

Trong chương trình trên chỉ có 2 phần là khai báo các thư viện và định nghĩa hàm **main()**. Các phần khai báo hàm nguyên mẫu, khai báo biến toàn cục và định nghĩa hàm nguyên mẫu không có trong chương trình này.

Các dòng 1, 2 là các dòng chú thích mô tả khái quát công việc chương trình sẽ thực hiện.

Dòng thứ 3 và thứ 4 là khai báo các tệp tiêu đề. Bởi vì trong chương trình ta sử dụng các hàm printf() (nằm trong thư viện **stdio** – *standard input/output*, thư viện chứa các hàm thực hiện các thao tác vào ra chuẩn) và getch() (nằm trong thư viện **conio** – *console input/output*, thư viện chứa các hàm thực hiện các thao tác vào ra qua bàn phím, màn hình...) nên ta phải khai báo với chương trình dịch gộp các thư viện đó vào chương trình. Nếu ta không gộp thư viện vào trong chương trình thì ta sẽ không thể sử dụng các hàm có trong thư viện đó.

Để gộp một thư viện vào trong chương trình (nhờ đó ta có thể sử dụng các hàm của thư viện đó), ta khai báo tệp tiêu đề tương ứng với thư viện đó ở đầu chương trình bằng chỉ thị có mẫu sau:

```
#include <tên_tệp_tieu_de>
```

Ví dụ để gộp thư viện conio vào chương trình ta dùng chỉ thị

```
#include <conio.h>
```

Lưu ý: Các tệp tiêu đề có tên là tên của thư viện, có phần mở rộng là .h (viết tắt của từ *header*).

Các dòng tiếp theo (từ dòng thứ 5 đến dòng thứ 25) là phần định nghĩa hàm **main()**, trong đó các dòng 7, 10, 15, 19, 23 là các dòng chú thích mô tả công việc mà các câu lệnh sau đó sẽ thực hiện.

Sau khi khai báo, biến có thể nhận giá trị thuộc kiểu dữ liệu đã khai báo. Chúng ta có thể khởi tạo giá trị đầu cho biến nếu muốn với cú pháp:

kiểu_dữ_liệu_tên_biến = giá_trị_đầu;

Ví dụ:

```
int a = 3;           // sau lệnh nay bien a se co gia tri bang 3
float x = 5.0, y = 2.6; // sau lệnh nay x co gia tri 5.0, y co gia tri 2.6
```

Biến dùng để lưu giữ giá trị, dùng làm toán hạng trong biểu thức, làm tham số cho hàm, làm biến chỉ số cho các cấu trúc lặp (**for**, **while**, **do while**), làm biểu thức điều kiện trong các cấu trúc rẽ nhánh (**if**, **switch**)...

Khai báo hằng

Có 2 cách để khai báo hằng trong C là dùng chỉ thị **#define** hoặc khai báo với từ khóa **const**.

Dùng chỉ thị **#define**

Cú pháp khai báo:

#define tên_hằng giá_trị

Lưu ý không có dấu chấm phẩy ở cuối dòng chỉ thị.

Với cách khai báo này, mỗi khi chương trình dịch gặp tên_hằng trong chương trình, nó sẽ tự động thay thế bằng giá_trị. Ở đây kiểu dữ liệu của hằng tự động được chương trình dịch xác định dựa theo nội dung của giá_trị.

Ví dụ:

```
#define MAX_SINH_VIEN 50           // hằng kiểu số nguyên
#define CNTT "Cong nghe thong tin" // hằng kiểu xâu kí tự (string)
#define DIEM_CHUAN 23.5           // hằng kiểu số thực
```

Dùng từ khóa **const** để khai báo với cú pháp:

const kiểu_dữ_liệu_tên_hằng = giá_trị;

Khai báo này giống với khai báo biến có khởi tạo giá trị đầu, tuy nhiên cần lưu ý:

- Do có từ khóa **const** ở đầu cho nên giá trị của đối tượng tên_hằng sẽ không được phép thay đổi trong chương trình. Những lệnh nhằm làm thay đổi giá trị của tên_hằng trong chương trình sẽ dẫn tới lỗi biên dịch.
- Trong khai báo biến thông thường, người lập trình có thể khởi tạo giá trị cho biến ngay từ khi khai báo hoặc không khởi tạo cũng được. Nhưng trong khai báo hằng, giá trị của tất cả các hằng cần được xác định ngay trong lệnh khai báo.

Các khai báo hằng ở ví dụ trước giờ có thể viết lại theo cách khác như sau:

```
const int MAX_SINH_VIEN = 50;
const char CNTT[20] = "Cong nghe thong tin";
const float DIEM_CHUAN = 23.5;
```

Trong phép toán gán nếu ta bỏ dấu ; ở cuối đi thì ta sẽ thu được biểu thức gán. Biểu thức gán là biểu thức có dạng

$$\text{tên_biến} = \text{biểu_thức}$$

Biểu thức gán là biểu thức nên nó cũng có giá trị. Giá trị của biểu thức gán bằng giá trị của biểu_thức, do đó ta có thể gán giá trị của biểu thức gán cho một biến khác hoặc sử dụng như một biểu thức bình thường. Ví dụ

```
int a, b, c;  
a = b = 2007;  
c = (a = 20) * (b = 30);
```

Trong câu lệnh thứ 2, ta đã gán giá trị 2007 cho biến b , sau đó ta gán giá trị của biểu thức $b = 2007$ cho biến a . Giá trị của biểu thức $b = 2007$ là 2007, do đó kết thúc câu lệnh này ta có a bằng 2007, b bằng 2007.

Trong câu lệnh thứ 3, ta gán giá trị 20 cho a , gán giá trị 30 cho b . Sau đó ta tính giá trị của biểu thức tích $(a = 20) * (b = 30)$ từ giá trị các biểu thức con $a = 20$ (có giá trị là 20) và $b = 30$ (có giá trị là 30). Cuối cùng ta gán giá trị của biểu thức tích thu được (600) cho biến c ;

Phép toán gán thu gọn

Xét lệnh gán sau

$$x = x + y;$$

Lệnh gán này sẽ tăng giá trị của biến x thêm một lượng có giá trị bằng giá trị của y . Trong C ta có thể viết lại lệnh này một cách gọn hơn mà thu được kết quả tương đương

$$x += y;$$

Dạng lệnh gán thu gọn này còn áp dụng được với các phép toán khác nữa.

Lệnh gán thông thường

$$x = x + y$$

$$x = x - y$$

$$x = x * y$$

$$x = x / y$$

$$x = x \% y$$

$$x = x \gg y$$

$$x = x \ll y$$

$$x = x \& y$$

Lệnh gán thu gọn

$$x += y$$

$$x -= y$$

$$x *= y$$

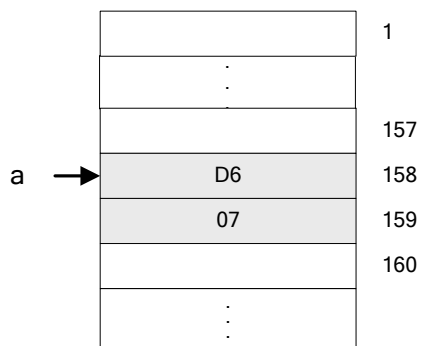
$$x /= y$$

$$x \% = y$$

$$x \gg = y$$

$$x \ll = y$$

$$x \& = y$$



Trong C để xác định địa chỉ của một biến ta sử dụng toán tử một ngôi & đặt trước tên biến, cú pháp là

& <tên biến>;

Ví dụ

`&a;` // có giá trị là 158 hay 9E

Phép toán chuyển đổi kiểu bắt buộc

Chuyển đổi kiểu là chuyển kiểu dữ liệu của một biến từ kiểu dữ liệu này sang kiểu dữ liệu khác. Cú pháp của lệnh chuyển kiểu dữ liệu là như sau:

<kiểu dữ liệu mới> <biểu thức>;

Có những sự chuyển đổi được thực hiện hết sức tự nhiên, không có khó khăn gì, thậm chí đôi khi chương trình dịch sẽ tự động chuyển đổi kiểu hộ cho ta, ví dụ chuyển một dữ liệu kiểu số nguyên **int** sang một số nguyên kiểu **long int**, hay từ một số **long int** sang một số thực **float**... Đó là vì một số nguyên kiểu **int** thực ra cũng là một số nguyên kiểu **long int**, một số nguyên kiểu **long int** cũng chính là một số thực kiểu **float**, một số thực kiểu **float** cũng là một số thực kiểu **double**.

Tuy nhiên điều ngược lại thì chưa chắc, ví dụ số nguyên **long int** 50,000 không phải là một số nguyên kiểu **int** vì phạm vi biểu diễn của kiểu **int** là từ (-32,768 đến 32,767). Khi đó nếu phải chuyển kiểu dữ liệu thì ta phải cẩn thận nếu không sẽ bị mất dữ liệu. Ví dụ một số thực **float** khi chuyển sang kiểu số nguyên **int** sẽ bị loại bỏ phần thập phân, còn một số nguyên kiểu **long int** khi chuyển sang kiểu **int** sẽ nhiều khả năng thu được một giá trị xa lạ. Ví dụ 1,193,046 (0x123456) là một số kiểu **long int**, khi chuyển sang kiểu **int** sẽ thu được 13,398 (0x3456). Đoạn chương trình sau minh họa điều đó.

```
#include <stdio.h>
#include <conio.h>
void main()
{
    long int li;
    int i;
    float f;
    clrscr();
    li = 0x123456;
    f = 123.456;
    i = (int) li;
```

```
printf("\n li = %ld",li);
printf("\n i = %d",i);
i = (int) f;
printf("\n f = %f",f);
printf("\n i = %d",i);
getch();
}
```

Kết quả

```
li = 1193046
i = 13398
f = 123.456001
i = 123
```

C hỗ trợ chuyển kiểu tự động trong những trường hợp sau

char → int → long int → float → double → long double

Khả năng chuyển kiểu dữ liệu là một điểm mạnh của C, tạo sự linh hoạt cho biến trong chương trình. Tuy nhiên nếu các bạn mới bắt đầu lập trình thì các bạn không nên chuyển kiểu dữ liệu trong chương trình vì có thể sinh ra những kết quả không như ý nếu bạn chưa có kinh nghiệm. Ngoài ra hình dung trước kiểu dữ liệu cho một biến và duy trì kiểu dữ liệu đó trong suốt chương trình cũng là một thói quen lập trình tốt.

Biểu thức điều kiện

Là biểu thức có dạng

biểu_thức_1 ? biểu_thức_2 : biểu_thức_3

Giá trị của biểu thức điều kiện sẽ là giá trị của biểu_thức_2 nếu biểu_thức_1 có giá trị khác 0 (tương ứng với giá trị logic ĐÚNG), và trái lại giá trị của biểu thức điều kiện sẽ là giá trị của biểu_thức_3 nếu biểu_thức_1 có giá trị bằng 0 (tương ứng với giá trị logic SAI).

Ví dụ sau sẽ cho ta xác định được giá trị nhỏ nhất của 2 số nhờ sử dụng biểu thức điều kiện

```
float x, y, z;           // khai báo biến
x = 3.8; y = 2.6;       // gán giá trị cho các biến x, y
z = (x < y) ? x : y;    // z sẽ có giá trị bằng giá trị
                        // nhỏ nhất trong 2 số x và y
```

Lệnh dãy

Lệnh dãy là lệnh gồm một dãy các biểu thức phân cách nhau bằng dấu phẩy và kết thúc lệnh là dấu chấm phẩy. Nó có dạng

biểu_thức_1, biểu_thức_2, ..., biểu_thức_n;

Trong lệnh dãy các biểu thức được tính toán độc lập với nhau.

và %f, với %d dùng để báo cho máy biết rằng cần phải hiển thị tham số tương ứng (biến a) theo định dạng số nguyên và %f dùng để báo cho máy cần hiển thị tham số tương ứng (biến x) theo định dạng số thực.

Lưu ý là mỗi nhóm kí tự định dạng chỉ dùng cho một kiểu dữ liệu, còn một kiểu dữ liệu có thể hiển thị theo nhiều cách khác nhau nên có nhiều nhóm kí tự định dạng khác nhau. Nếu giữa nhóm kí tự định dạng và tham số tương ứng không phù hợp với nhau thì sẽ hiển thị ra kết quả không như ý. Phần sau đây giới thiệu một số nhóm kí tự định dạng hay dùng trong C và ý nghĩa của chúng.

Nhóm kí tự định dạng	Áp dụng cho kiểu dữ liệu	Ghi chú
%d	int, long, char	Hiển thị tham số tương ứng dưới dạng số nguyên có dấu hệ đếm thập phân
%i	int, long, char	Hiển thị tham số tương ứng dưới dạng số nguyên có dấu hệ đếm thập phân
%o	int, long, char	Hiển thị tham số tương ứng dưới dạng số nguyên không dấu trong hệ đếm cơ số 8.
%u	int, long, char	Hiển thị tham số tương ứng dưới dạng số nguyên không dấu.
%x	int, long, char	Hiển thị tham số tương ứng dưới dạng số nguyên hệ đếm 16 (không có 0x đứng trước), sử dụng các chữ cái a b c d e f
%X	int, long, char	Hiển thị tham số tương ứng dưới dạng số nguyên hệ đếm 16 (không có 0x đứng trước), sử dụng các chữ cái A B C D E F
%e	float, double	Hiển thị tham số tương ứng dưới dạng số thực dấu phẩy động
%f	float, double	Hiển thị tham số tương ứng dưới dạng số thực dấu phẩy tĩnh
%g	float, double	Hiển thị tham số tương ứng số thực dưới dạng ngắn gọn hơn trong 2 dạng dấu phẩy tĩnh và dấu phẩy động
%c	int, long, char	Hiển thị tham số tương ứng dưới dạng kí tự
%s	char * (xâu kí tự)	Hiển thị tham số tương ứng dưới dạng xâu kí tự

Để trình bày dữ liệu được đẹp hơn, C cho phép đưa thêm một số thuộc tính định dạng dữ liệu khác vào trong xâu định dạng như độ rộng tối thiểu, căn lề trái, căn lề phải.

Độ rộng tối thiểu

Thông thường khi hiển thị dữ liệu, C tự động xác định số chỗ cần thiết sao cho hiển thị vừa đủ nội dung dữ liệu.

Nếu ta muốn C hiển thị dữ liệu của ta trên một số lượng vị trí xác định bất kể nội dung dữ liệu đó có điền đầy số chỗ được cung cấp hay không, ta có thể chèn một số nguyên vào trong nhóm kí tự định dạng, ngay sau dấu %.

Ví dụ khi hiển thị số nguyên

```
a = 1234;
printf("\n%5d",a); // dành 5 chỗ để hiển thị số nguyên a
printf("\n%5d",34); // dành 5 chỗ để hiển thị số nguyên 34
```

Kết quả

```
□1234
□□□34
```

Ở đây □ kí hiệu thay cho dấu trắng (space).

Như vậy với nhóm kí tự định dạng **%md**, m dùng để báo số chỗ cần dành để hiển thị dữ liệu, còn d báo rằng hãy hiển thị dữ liệu đó dưới dạng một số nguyên. Tương tự với các nhóm kí tự định dạng **%mc** khi hiển thị kí tự, và **%ms** khi hiển thị xâu kí tự.

Ví dụ

```
printf("\n%3d %15s %3c", 1, "nguyen van a", 'g');
printf("\n%3d %15s %3c", 2, "tran van b", 'k');
```

Kết quả

```
□□1□□□□nguyen van a□□g
□□2□□□□□□tran van b□□k
```

Nếu nội dung dữ liệu không điền đầy số chỗ được cấp thì những chỗ không dùng đến sẽ được điền bởi dấu trắng.

Khi số chỗ cần thiết để hiển thị nội dung dữ liệu lớn hơn m thì C tự động cung cấp thêm chỗ mới để hiển thị chứ không cắt bớt nội dung của dữ liệu để cho vừa m vị trí.

Với dữ liệu là số thực ta sử dụng mẫu nhóm kí tự định dạng **%m.nf** để báo rằng cần dành m vị trí để hiển thị số thực, và trong m vị trí đó dành n vị trí để hiển thị phần thập phân.

Ví dụ:

```
printf("\n%f",12.345);
printf("\n%.2f",12.345);
printf("\n%8.2f",12.345);
```

Kết quả

```
12.345000
12.35
□□□12.35
```

Căn lề trái

Khi hiển thị dữ liệu, mặc định C căn lề phải. Nếu muốn căn lề trái khi hiển thị dữ liệu ta chỉ cần thêm dấu trừ - vào ngay sau dấu %.

Ví dụ

```
printf("\n%-3d %-15s %-4.2f %-3c", 1, "nguyen van a", 8.5, 'g');
printf("\n%-3d %-15s %-4.2f %-3c", 2, "tran van b", 6.75, 'k');
```

Kết quả

```
1  nguyen van a   8.50 g
2  tran van b    6.75 k
```

Dễ thấy các thuộc tính định dạng độ rộng tối thiểu, căn lề... giúp cho việc hiển thị dữ liệu được thẳng, đều và đẹp hơn.

Thuộc tính	Quy cách kí tự định dạng (m, n là các số nguyên)	Ví dụ
Độ rộng tối thiểu	%md, %ms, %mc	printf("%3d",10); printf("%4s","CNTT"); printf("%2c",'A');
Độ rộng dành cho phần thập phân	%m.nf	printf("%5.1f",1234.5);
Căn lề trái	%-md, %-ms, %-mc	printf("%-3d",10); printf("%-4s","CNTT"); printf("%-2c",'A');

Hàm scanf()

Cú pháp:

scanf(xâu_định_dạng, [danh_sách_địa_chỉ]);

Hàm scanf() dùng để nhập dữ liệu từ bàn phím. Cụ thể nó sẽ đọc các kí tự được nhập từ bàn phím, sau đó căn cứ theo xâu_định_dạng sẽ chuyển những thông tin đã nhập được sang kiểu dữ liệu phù hợp. Cuối cùng sẽ gán những giá trị vừa nhập được vào các biến tương ứng trong danh_sách_địa_chỉ.

xâu_định_dạng trong hàm scanf() xác định khuôn dạng của các dữ liệu được nhập vào. Trong xâu_định_dạng có chứa các nhóm kí tự định dạng xác định khuôn dạng dữ liệu nhập vào.

Địa chỉ của một biến được viết bằng cách đặt dấu & trước tên biến. Ví dụ giả sử ta có các biến có tên là a, x, ten_bien thì địa chỉ của chúng lần lượt sẽ là &a, &x, &ten_bien.

danh_sách_địa_chỉ phải phù hợp với các nhóm kí tự định dạng trong xâu_định_dạng về số lượng, kiểu dữ liệu và thứ tự. Số nhóm kí tự định dạng bằng số địa chỉ của các biến trong danh_sách_địa_chỉ. Dưới đây là một số nhóm kí tự định dạng hay dùng và ý nghĩa

Nhóm kí tự định dạng	Ghi chú
%d	Định khuôn dạng dữ liệu nhập vào dưới dạng số nguyên kiểu int

%o	Định khuôn dạng dữ liệu nhập vào dưới dạng số nguyên kiểu int hệ cơ số 8
%x	Định khuôn dạng dữ liệu nhập vào dưới dạng số nguyên kiểu int hệ cơ số 16
%c	Định khuôn dạng dữ liệu nhập vào dưới dạng kí tự kiểu char
%s	Định khuôn dạng dữ liệu nhập vào dưới dạng xâu kí tự
%f	Định khuôn dạng dữ liệu nhập vào dưới dạng số thực kiểu float
%ld	Định khuôn dạng dữ liệu nhập vào dưới dạng số nguyên kiểu long
%lf	Định khuôn dạng dữ liệu nhập vào dưới dạng số thực kiểu double

Ví dụ:

```
#include <conio.h>
#include <stdio.h>
void main()
{
    // khai bao bien
    int a;
    float x;
    char ch;
    char* str;
    // Nhap du lieu
    printf("Nhap vao mot so nguyen");
    scanf("%d",&a);
    printf("\n Nhap vao mot so thuc");
    scanf("%f",&x);
    printf("\n Nhap vao mot ki tu");
    fflush(stdin); scanf("%c",&ch);
    printf("\n Nhap vao mot xau ki tu");
    fflush(stdin); scanf("%s",str);
    // Hien thi du lieu vua nhap vao
    printf("\n Nhung du lieu vua nhap vao");
    printf("\n So nguyen: %d",a);
    printf("\n So thuc : %.2f",x);
    printf("\n Ki tu: %c",ch);
    printf("\n Xau ki tu: %s",str);
}
```

Kết quả:

```
Nhap vao mot so nguyen: 2007
Nhap vao mot so thuc: 18.1625
Nhap vao mot ki tu: b
Nhap vao mot xau ki tu: ngon ngu lap trinh C
Nhung du lieu vua nhap vao
So nguyen: 2007
So thuc: 18.16
```



```
{  
lệnh;  
    {  
        lệnh;  
        ...  
    }  
    ...  
}
```

C cho phép khai báo biến trong lệnh khối. Ràng buộc duy nhất là phần khai báo phải nằm trước phần câu lệnh.

Ví dụ:

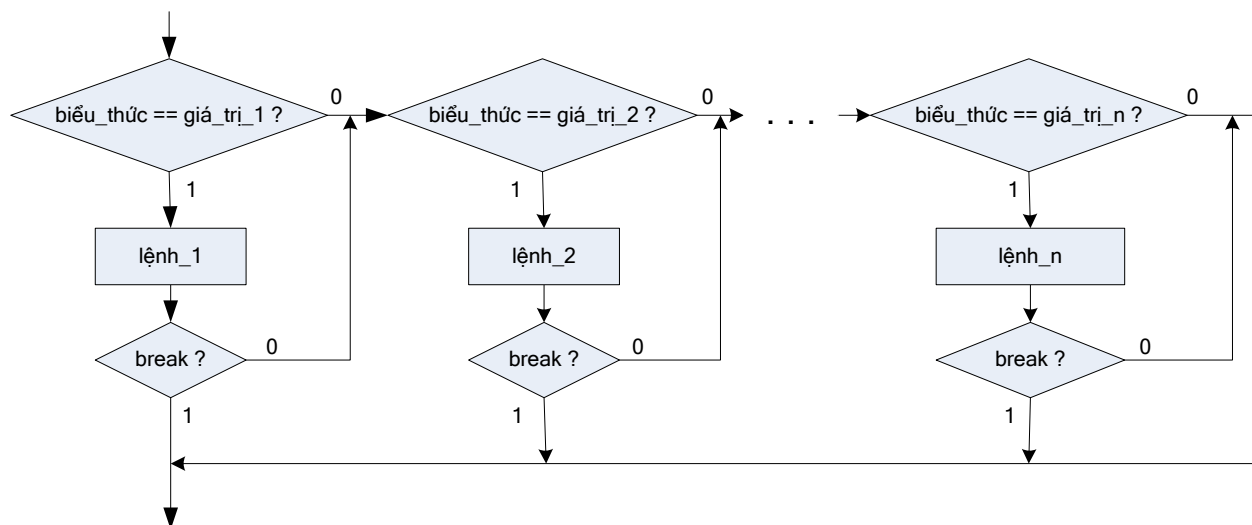
```
#include <conio.h>  
#include <stdio.h>  
void main()  
// Nội dung của hàm main() cũng là một khối lệnh  
{  
    // khai báo biến  
    int c;  
    c = 10;  
    printf(" Gia tri cua c = %d day la c ngoai",c);  
    // bat dau mot khoi lenh khac  
    {  
        int c;  
        c = 10;  
        printf("\n Gia tri cua c = %d day la c trong",c);  
        printf("\n Tang gia tri cua c them 10 don vi");  
        c = c + 10;  
        printf("\n Gia tri cua c = %d day la c trong",c);  
    }  
    printf("\n Gia tri cua c = %d day la c ngoai",c);  
    getch();  
}
```

Kết quả:

```
Gia tri cua c = 10 day la c ngoai  
Gia tri cua c = 10 day la c trong  
Tang gia tri cua c them 10 don vi  
Gia tri cua c = 20 day la c trong  
Gia tri cua c = 10 day la c ngoai
```

- Nếu không có nhãn **default**: chương trình chuyển sang thực hiện lệnh tiếp theo sau cấu trúc **switch**.

Sơ đồ:



Ví dụ:

/* Ví dụ sau yêu cầu người dùng nhập vào một số nguyên không âm và đưa ra ngày trong tuần tương ứng với số nguyên đó. Và ở đây ta quy ước những số chia hết cho 7 ứng với Chủ nhật, chia 7 dư 1 ứng với thứ Hai, ..., chia 7 dư 6 ứng với thứ Bảy.*/

```

#include <conio.h>
#include <stdio.h>
void main()
{
    // khai bao bien
    int a;
    do
    {
        printf("\n Nhap mot gia tri so nguyen khong am: ");
        scanf("%d",&a);
        if(a<0)
            printf("\n so vua nhap la so am");
    }while(a<0);
    printf("\n Thu trong tuan tuong ung voi so do la: ");
    switch(a % 7)
    {
        case 0: printf(" Chu nhat"); break;
        case 1: printf(" Thu Hai"); break;
        case 2: printf(" Thu Ba"); break;
        case 3: printf(" Thu Tu"); break;
        case 4: printf(" Thu Nam"); break;
    }
}
  
```

```
        case 5: printf(" Thu Sau"); break;
        case 6: printf(" Thu Bay"); break;
    }
    getch();
}
```

Kết quả:

```
Nhap vao mot gia tri so nguyen: 2356
Thu tuong ung voi so do la Thu Nam
```

Người ta thường dựa trên tính chất tự động chuyển xuống các câu lệnh sau khi không có lệnh break để viết chung mã lệnh cho các trường hợp khác nhau nhưng cùng được xử lý giống nhau. Ví dụ khi viết chương trình hỗ trợ menu dòng lệnh không phân biệt chữ hoa chữ thường hay bài toán in ra số ngày trong các tháng trong năm dưới đây. Trong một năm các tháng có 30 ngày là 4, 6, 9, 11 còn các tháng có 31 ngày là 1, 3, 5, 7, 8, 10, 12. Riêng tháng hai có thể có 28 hoặc 29 ngày.

```
#include <stdio.h>
#include<conio.h>
int main ()
{
    int thang;
    clrscr();
    printf("\n Nhap vao thangs trong nam ");
    scanf("%d",&thang);
    switch(thang)
    {
        case 1:
        case 3:
        case 5:
        case 7:
        case 8:
        case 10:
        case 12:
            printf("\n Thang %d co 31 ngay ",thang);
            break;
        case 4:
            case 6:
        case 9:
        case 11:
            printf("\n Thang %d co 30 ngay ",thang);
            break;
            case 2:
                printf ("\ Thang 2 co 28 hoac 29 ngay");
                break;
        default :
            printf("\n Khong co thang %d", thang);
    }
```

Bước 3: tăng giá trị của i thêm 1 đơn vị.

Bước 4: kiểm tra nếu giá trị của $i \leq 9$ thì quay về bước 2, nếu giá trị của $i > 9$ thì chuyển sang bước 5.

Bước 5: kết thúc.

Chương trình in ra màn hình các số nguyên dương nhỏ hơn 10 sử dụng vòng lặp for:

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int i;
    for(i = 1; i <= 9; i++)
        printf("%5d", i); // ta dành 5 vị trí để in mỗi số
    getch();
}
```

Kết quả thực hiện

```
0 1 2 3 4 5 6 7 8 9
```

Ta có thể so sánh cách làm này với việc phải viết một lệnh printf() trong đó phải liệt kê toàn bộ các số nguyên dương lẻ nhỏ hơn 10.

Ví dụ 2: Tính và hiển thị ra màn hình tổng của 100 số tự nhiên lẻ đầu tiên.

Cách làm: 100 số lẻ đầu tiên là 1, 3, 5, ..., 199. Ta cần sử dụng một biến nguyên S để chứa giá trị của tổng và một biến nguyên i.

Bước 1: ban đầu gán cho S giá trị bằng 0, gán cho i giá trị bằng 1;

Bước 2: $S = S + i$;

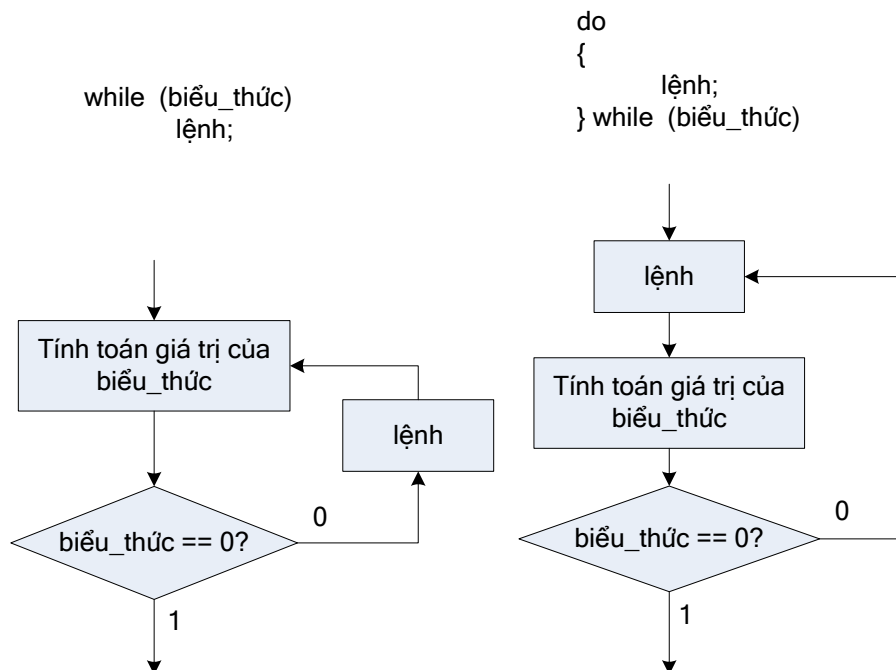
Bước 3: tăng giá trị của i thêm 2 đơn vị

Bước 4: kiểm tra nếu giá trị của $i \leq 199$ thì quay về bước 2, ngược lại nếu $i > 199$ thì chuyển sang bước 5.

Bước 5: kết thúc.

Chương trình:

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int i;
    int S;
    S = 0;
    for(i = 1; i <= 199; i = i+2)
        S = S+i;
    printf("Tổng của 100 số nguyên dương lẻ đầu tiên là %d", S);
}
```



Sự khác nhau giữa **while** và **do{...}while**

- Lệnh **while** kiểm tra điều kiện vòng lặp (tức là giá trị của biểu thức) trước rồi mới thực hiện lệnh, và do vậy sẽ có khả năng lệnh không được thực hiện lần nào.
- Lệnh **do{...}while** thực hiện lệnh trước rồi mới kiểm tra điều kiện của vòng lặp, và do vậy lệnh sẽ luôn được thực hiện ít nhất một lần.

Cấu trúc **while** và **do{...}while** được dùng để thực hiện lặp đi lặp lại một công việc nào đó với số lần lặp không xác định.

Ví dụ: Sau đây là 2 đoạn chương trình có chức năng tương đương nhau nhưng một đoạn sử dụng cấu trúc **while**, một đoạn sử dụng cấu trúc **do{...}while**.

Chức năng của chương trình là yêu cầu người dùng nhập vào giá trị 1 số nguyên, đưa ra thông báo số đó có phải là số hoàn thiện hay không, sau đó hỏi người dùng có muốn nhập lại số nguyên khác và kiểm tra có phải số hoàn thiện hay không.

Đoạn chương trình sử dụng cấu trúc **do{...}while**

```
#include <stdio.h>
#include <conio.h>
void main()
{
    long int n;
    long int tong;
    long int i;
    char ch;

    clrscr();
```

```

do
{
    tong = 0;
    printf("\n Nhap vao mot so nguyen: ");
    scanf("%ld",&n);
    printf("\n Cac uoc so cua %ld la: ",n);
    for(i = 1;i<n;i++)
        if(n % i == 0)
    {
        printf("%5d",i);
        tong = tong + i;
    }
    printf("\n Tong cac uoc so cua %ld bang %ld",n,tong);
    if(tong == n)
        printf("\n %5ld LA so hoan thien");
    else
        printf("\n %5ld KHONG LA so hoan thien");
    printf("\n Ban co muon thuc hien lai(c/k)? ");
    fflush(stdin);
    scanf("%c",&ch);
} while((ch!='k')&&(ch!='K'));
printf("\n An phim bat ki de ket thuc ...");
getch();
}

```

Đoạn chương trình viết bằng cấu trúc **while**

```

#include <stdio.h>
#include <conio.h>
void main()
{
    long int n;
    long int tong;
    long int i;
    char ch;

    clrscr();
    ch = 'c';
    while((ch != 'k')&&(ch != 'K'))
    {
        tong = 0;
        printf("\n Nhap vao mot so nguyen: ");
        scanf("%ld",&n);
        printf("\n Cac uoc so cua %ld la: ",n);
        for(i = 1;i<n;i++)
            if(n % i == 0)
        {
            printf("%5d",i);

```

```

int a[5];
int i, k;
// Nhập giá trị cho các phần tử của mảng a từ bàn phím
for(i = 0; i < 5; i++)
{
    printf("\n a[%d] = ", i);
    scanf("%d", &a[i]);
}
// Hiện thị giá trị của phần tử bất kỳ, giả sử a[3] lên màn hình
printf("\n a[3] = %d", a[3]);
// Hiện thị giá trị của tất cả các phần tử, mỗi phần tử trên một dòng
for(i = 0; i < 5; i++)
    printf("\n%d", a[i]);
// Hiện thị giá trị của tất cả các phần tử trên cùng một dòng, các giá trị cách nhau 2 vị trí
printf("\n"); // Xuống dòng mới
for(i = 0; i < 5; i++)
    printf("%d ", a[i]);
// Hiện thị giá trị của tất cả các phần tử, trong đó k phần tử trên một dòng.
// Các phần tử trên cùng dòng cách nhau 2 vị trí
printf("\n Cho biết giá trị của k = ");
scanf("%d",&k);
for(i = 0; i < 5; i++)
{
    printf("%d ",a[i]);
    if((i+1)%k == 0) // đã hiện thị đủ k phần tử trên một dòng thì phải xuống
dong
        printf("\n");
}
getch();
}

```

Kết quả

```

a[0] = 6
a[1] = 14
a[2] = 23
a[3] = 37
a[4] = 9
a[3] = 37
6
14
23
37
9
6 14 23 37 9
Cho biết giá trị của k = 2
6 14

```

thì ta trả lời là không có phần tử như vậy trong mảng. Phương pháp trên được gọi là phương pháp tìm kiếm tuần tự (*sequential search*).

Dưới đây là cài đặt của thuật toán tìm kiếm tuần tự cho trường hợp mảng dữ liệu là mảng các số nguyên kiểu **int**.

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int m[100], chi_so[100];
    int n; // n là số phần tử trong mảng
    int i, k, kiem_tra;
    clrscr(); // xóa màn hình để tiện theo dõi
    // Nhập giá trị dữ liệu cho mảng m
    // Trước tiên phải biết số phần tử của mảng
    printf(" Cho biet so phan tu co trong mang: ");
    scanf("%d",&n);
    // Rồi lần lượt nhập giá trị cho các phần tử trong mảng
    for(i = 0;i<n;i++)
    {
        int temp;
        printf("\n Cho biet gia tri cua m[%d] = ",i);
        scanf("%d",&temp);
        m[i] = temp;
    }
    // Yêu cầu người sử dụng nhập vào giá trị cho dữ liệu k
    printf("\n Cho biết giá trị của dữ liệu k: ");
    scanf("%d",&k);
    // Bắt đầu quá trình tìm kiếm
    kiem_tra = 0;
    // Duyệt qua tất cả các phần tử
    for(i = 0;i<n;i++)
        if(m[i] == k)//So sánh phần tử đang xét với dữ liệu k
        {
            // Ghi nhận chỉ số của phần tử đang xét
            chi_so[kiem_tra] = i;
            kiem_tra ++; //Tăng biến kiem_tra thêm 1 đơn vị
        }
    // Kết luận
    if(kiem_tra > 0)
    {
        printf("\n Trong mang co %d phan tu co gia tri bang %d",kiem_tra,k);
        printf("\n Chi so cua cac phan tu la: ");
        for(i = 0;i < kiem_tra;i++)
            printf("%3d",chi_so[i]);
    }
    else
```



```

scanf("%d",&n);
// Rời lần lượt nhập giá trị cho các phần tử trong mảng
for(i = 0;i<n;i++)
{
    int temp;
    printf("\n Cho biet gia tri cua m[%d] = ",i);
    scanf("%d",&temp);
    m[i] = temp;
}
// Hiện thị ra màn hình mảng vừa nhập vào
printf("\n Mang truoc khi sap xep:  ");
for(i=0;i<n;i++)
    printf("%3d",m[i]);
// Bắt đầu sắp xếp
for(i = 0; i<n-1;i++)
{
    // Ở lượt sắp xếp thứ i+1
    for(j = i+1;j<n;j++)
    {
        // So sánh m[i] với các phần tử còn lại
        // và đổi chỗ khi tìm thấy phần tử < m[i].
        if(m[j]<m[i])
        {
            int temp;
            temp = m[j]; m[j] = m[i]; m[i] = temp;
        }
    }
    // Hiện thị mảng sau lượt sắp xếp thứ i+1
    printf("\n Mang o luot sap xep thu %d",i+1);
    for(k = 0;k < n ;k++)
        printf("%3d",m[k]);
}
getch(); // Chờ người sử dụng ấn phím bất kì để kết thúc.
}

```

Kết quả thực hiện:

```

Cho biet so phan tu co trong mang: 5
Cho biet gia tri cua m[0]: 34
Cho biet gia tri cua m[1]: 20
Cho biet gia tri cua m[2]: 17
Cho biet gia tri cua m[3]: 65
Cho biet gia tri cua m[4]: 21
Mang truoc khi sap xep:  34 20 17 65 21
Mang o luot sap xep thu 1: 17 34 20 65 21
Mang o luot sap xep thu 2: 17 20 34 65 21

```

int toupper(int ch)

Hàm toupper() dùng để chuyển một kí tự chữ cái thường (các kí tự 'a', 'b', ..., 'z') thành kí tự chữ cái hoa tương ứng ('A', 'B', ..., 'Z').

Hàm tolower()

int tolower(int ch)

Hàm tolower() dùng để chuyển một kí tự chữ cái hoa ('A', 'B', ..., 'Z') thành kí tự chữ cái thường tương ứng ('a', 'b', ...'z').

Hàm isalpha()

int isalpha(int ch)

Hàm isalpha() dùng để kiểm tra một kí tự có phải là chữ cái hay không ('a', 'b', ..., 'z', 'A', 'B', ..., 'Z'). Hàm trả về giá trị khác không nếu đúng là chữ cái, trả về giá trị 0 nếu ngược lại.

Hàm isdigit()

int isdigit(int ch)

Hàm isdigit() dùng để kiểm tra một kí tự có phải là chữ số hay không ('0', '1', ...'9'). Hàm trả về giá trị khác không nếu đúng, trả về giá trị 0 nếu ngược lại.

Hàm islower()

int islower(int ch)

Hàm islower() dùng để kiểm tra một kí tự có phải là chữ cái thường hay không ('a', 'b', ...'z'). Hàm trả về giá trị khác không nếu đúng, trả về giá trị 0 nếu ngược lại.

Hàm isupper()

int isupper(int ch)

Hàm isupper() dùng để kiểm tra một kí tự có phải là chữ cái hoa hay không ('A', 'B', ...'Z'). Hàm trả về giá trị khác không nếu đúng, trả về giá trị 0 nếu ngược lại.

Hàm iscntrl()

int iscntrl(int ch)

Hàm iscntrl() dùng để kiểm tra một kí tự có phải là kí tự điều khiển hay không (là các kí tự không hiển thị được và có mã ASCII từ 0 đến 31). Hàm trả về giá trị khác không nếu đúng, trả về giá trị 0 nếu ngược lại.

Hàm isspace()

int isspace(int ch)

Hàm isspace() dùng để kiểm tra một kí tự có phải là dấu cách (space, mã ASCII là 32), kí tự xuống dòng ('\n', mã ASCII là 10), kí tự về đầu dòng ('\r', mã ASCII là 13), dấu tab ngang ('\t', mã ASCII là 9) hay dấu tab dọc ('\v', mã ASCII là 11) hay không. Hàm trả về giá trị khác không nếu đúng, trả về giá trị 0 nếu ngược lại.

Hàm `strcat()`

char* strcat(char* `xâu_đích`, char* `xâu_nguồn`);

Hàm `strcat` sẽ ghép nối `xâu_nguồn` vào ngay sau `xâu_đích`. Kết quả trả về của hàm `strcat()` là `xâu` mới ghép nối từ 2 `xâu` `xâu_nguồn` và `xâu_đích`.

Hàm `strchr()`

char* strchr(char* `str`, int `ch`);

Hàm `strchr()` dùng để tìm kiếm vị trí của kí tự `ch` trong `xâu str`. Nếu có kí tự `ch` trong `str` thì hàm `strchr()` trả về con trỏ trỏ tới kí tự `ch` đầu tiên trong `str`, ngược lại nó sẽ trả về con trỏ `NULL`.

Hàm `strstr()`

char* strstr(char* `str1`, char* `str2`);

Hàm `strstr()` dùng để tìm kiếm vị trí của `xâu con str2` trong `xâu str1`. Nếu `str2` là `xâu con` của `str1` thì hàm `strstr()` trả về con trỏ trỏ tới kí tự đầu tiên của `xâu con str2` đầu tiên trong `str1`, ngược lại nó sẽ trả về con trỏ `NULL`.

Hàm `atoi()`

int atoi(char* `str`)

Hàm `atoi()` dùng để chuyển một `xâu kí tự` là biểu diễn của một số nguyên thành số nguyên tương ứng. Nếu chuyển đổi thành công, hàm `atoi()` trả về giá trị số nguyên chuyển đổi được, ngược lại trả về giá trị 0.

Hàm `atol()`

long int atol(char* `str`)

Hàm `atol()` dùng để chuyển một `xâu kí tự` là biểu diễn của một số nguyên dài (tương ứng với kiểu dữ liệu **long int**) thành số nguyên dài tương ứng. Nếu chuyển đổi thành công, hàm `atol()` trả về giá trị số nguyên chuyển đổi được, ngược lại trả về giá trị 0.

Hàm `atof()`

double atof(char* `str`)

Hàm `atof()` dùng để chuyển một `xâu kí tự` là biểu diễn của một số thực (ở cả dạng số dấu phẩy tĩnh và động đều được) thành số thực tương ứng. Nếu chuyển đổi thành công, hàm `atof()` trả về giá trị số thực chuyển đổi được, ngược lại trả về giá trị 0.

Các hàm `strcpy()`, `strlen()`, `strcmp()`, `strcat()`, `strchr()`, `strstr()` khai báo trong tệp tiêu đề `string.h`.

Các hàm `atoi()`, `atol()`, `atof()` khai báo trong tệp tiêu đề `stdlib.h`.

Ví dụ minh họa:

```
#include <stdio.h>
#include <conio.h>
#include <string.h> // Phải có thư viện string.h thì mới
                  // sử dụng được các hàm strcpy, strcmp...

void main()
```

từng trường và thực hiện thao tác cần thiết trên từng trường đó. Để truy nhập tới một trường trong cấu trúc ta dùng cú pháp sau:

tên_biến_cấu_trúc.tên_trường

Dấu chấm “.” sử dụng trong cú pháp trên là toán tử truy nhập thành phần cấu trúc, và nếu như trường được truy nhập lại là một cấu trúc thì ta có thể tiếp tục áp dụng toán tử này để truy nhập tới các trường thành phần nằm ở mức sâu hơn.

Giờ đây ta có thể “đổi xử” *tên_biến_cấu_trúc.tên_trường* giống như một biến thông thường có kiểu dữ liệu là kiểu dữ liệu của *tên_trường*, tức là ta có thể nhập giá trị, hiển thị giá trị của biến cấu trúc, sử dụng giá trị đó trong các biểu thức...

Ví dụ:

// dưới đây là một cấu trúc mô tả một điểm trong không gian 2 chiều.

// các trường dữ liệu gồm: tên của điểm và tọa độ của điểm đó.

// tọa độ là một cấu trúc gồm 2 trường: hoành độ và tung độ

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
void main()
```

```
{
```

```
    struct point_2D
```

```
    {
```

```
        char ten_diem;
```

```
        struct
```

```
        {
```

```
            float x, y;
```

```
        } toa_do;
```

```
    } p;
```

```
    float temp_float;
```

```
    char temp_char;
```

```
    printf("\n Hay nhap thong tin ve mot diem");
```

```
    printf("\n Ten cua diem: ");
```

```
    fflush(stdin);
```

```
    scanf("%c",&temp_char);
```

```
    p.ten_diem = temp_char;
```

```
    printf("\n nhap vao hoành do cua diem: ");
```

```
    scanf("%f",&temp_float);
```

```
    p.toa_do.x = temp_float;
```

```
    // giả sử điểm đang xét nằm trên đường thẳng  $y = 3x + 2$ ;
```

```
    p.toa_do.y = 3*p.toa_do.x + 2;
```

```
    printf("\n %c = (%5.2f,%5.2f)",p.ten_diem, p.toa_do.x, p.toa_do.y);
```

```
    getch();
```

```
}
```

Kết quả khi chạy chương trình:

```
Hay nhap thong tin ve mot diem
```

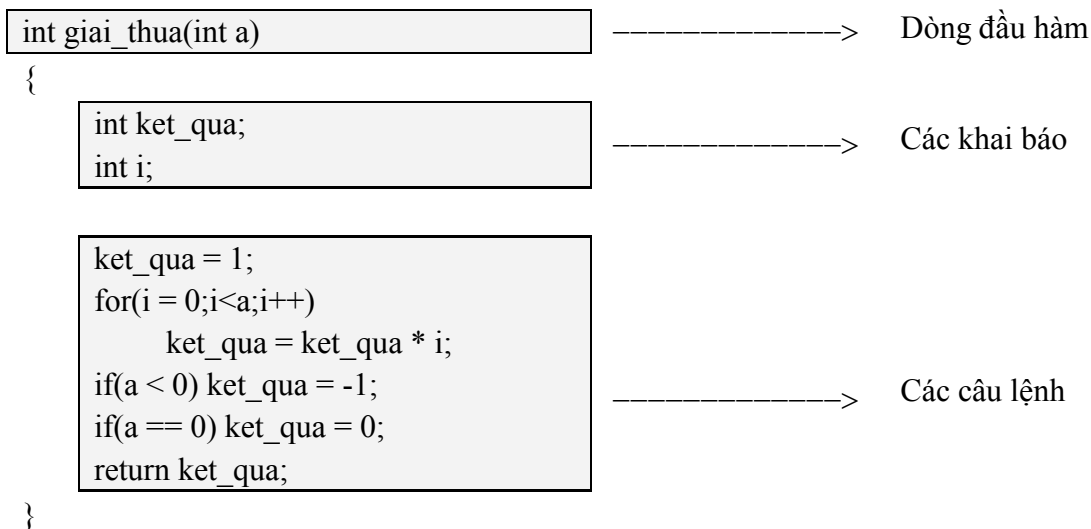
```
Ten cua diem: A
```

<Các câu lệnh>

}

Trong thân hàm có ít nhất một lệnh return.

Ví dụ sau là khai báo và định nghĩa hàm tính giai thừa của một số nguyên dương. Ta quy ước rằng giai thừa của một số âm thì bằng -1, của 0 bằng 0, của một số nguyên dương a là $a! = a \times (a-1) \times \dots \times 1$.



Các thành phần của dòng đầu hàm

Dòng đầu hàm là các thông tin được trao đổi giữa bên trong và bên ngoài hàm. Khi nói tới dòng đầu hàm là ta nói tới tên của hàm, hàm đó cần những thông tin gì từ môi trường để hoạt động (các tham số đầu vào), hàm đó cung cấp những thông tin gì cho môi trường (những tham số đầu ra và giá trị trả về).

Dòng đầu hàm phân biệt các hàm với nhau, hay nói cách khác không được có 2 hàm có dòng đầu hàm giống nhau.

Kiểu dữ liệu trả về của hàm

Thông thường hàm sau khi được thực hiện sẽ trả về một giá trị kết quả tính toán nào đó. Để sử dụng được giá trị đó ta cần phải biết nó thuộc kiểu dữ liệu gì. Kiểu dữ liệu của đối tượng tính toán được hàm trả về được gọi là kiểu dữ liệu trả về của hàm.

Trong C, kiểu dữ liệu trả về của hàm có thể là kiểu dữ liệu bất kì (kiểu dữ liệu có sẵn hoặc kiểu dữ liệu do người dùng tự định nghĩa) nhưng không được là kiểu dữ liệu mảng.

Nếu kiểu dữ liệu trả về là kiểu **void** thì hàm không trả về giá trị nào cả.

Trường hợp ta không khai báo kiểu dữ liệu trả về thì chương trình dịch của C sẽ ngầm hiểu rằng kiểu dữ liệu trả về của hàm là kiểu **int**.

Tên hàm

Tên hàm là có thể là bất kì một định danh hợp lệ nào. Tuy nhiên tên hàm nên mang nghĩa gợi ý chức năng công việc mà hàm thực hiện. Ví dụ một hàm có chức năng tính và trả về bình phương của một số thực x thì nên có tên là *binh_phuong*. Trong C, các hàm không được đặt tên trùng nhau.

Tham số của hàm

Tham số của hàm là các thông tin cần cho hoạt động của hàm và các thông tin, kết quả tính toán được hàm trả lại. Tức là có những tham số chứa dữ liệu vào cung cấp cho hàm, có những tham số chứa dữ liệu ra mà hàm tính toán được.

Các tham số sử dụng trong lời khai báo hàm được gọi là tham số hình thức. Nó là tham số giả định của hàm. Khi khai báo tham số hình thức của hàm phải chỉ ra tên của tham số và kiểu dữ liệu của tham số.

Các tham số được cung cấp cho hàm trong quá trình thực hiện của hàm được gọi là tham số thực. Kiểu dữ liệu của tham số thực cung cấp cho hàm trong chương trình phải giống kiểu dữ liệu của tham số hình thức tương ứng với tham số thực đó, nếu không sẽ có báo lỗi biên dịch.

Một hàm có thể có một, nhiều hoặc không có tham số nào cả. Nếu có nhiều tham số thì chúng phải được phân cách với nhau bằng dấu phẩy. Lưu ý là nếu hàm không có tham số nào cả thì vẫn phải có cặp dấu ngoặc đơn sau tên hàm, ví dụ **main()**.

Lệnh return

Trong chương trình, một hàm được thực hiện khi ta gặp lời gọi hàm của hàm đó trong chương trình. Một lời gọi hàm là tên hàm theo sau bởi các tham số thực trong chương trình. Sau khi hàm thực hiện xong, nó sẽ trở về chương trình đã gọi nó. Có 2 cách để từ hàm trở về chương trình đã gọi hàm:

- Sau khi thực hiện tất cả các câu lệnh có trong thân hàm.
- Khi gặp lệnh **return**.

Cú pháp chung của lệnh **return** là

return biểu_thức;

Khi gặp lệnh này, chương trình sẽ tính toán giá trị của *biểu_thức*, lấy kết quả tính toán được làm giá trị trả về cho lời gọi hàm rồi kết thúc việc thực hiện hàm, trở về chương trình đã gọi nó.

Trong lệnh **return** cũng có thể không có phần *biểu_thức*, khi đó ta sẽ kết thúc thực hiện hàm mà không trả về giá trị nào cả.

Ví dụ và phân tích.

```
#include <stdio.h>
#include <conio.h>
int max(int x, int y, int z)
{
    int max;
    max = x>y?x:y;
    max = max>z?max:z;
```

```
printf("\n Nhap vao 2 so nguyen duong a va b ");
printf("\n a = "); scanf("%d",&a);
printf("\n b = "); scanf("%d",&b);
if(a*b == 0)
{
    printf("\n Khong hop le");
    continue;
}
printf("\n Uoc chung lon nhat cua %d va %d la: %d", a, b, Uscln(a, b));
}while ((a != 0)||(b != 0));
printf("\n An phim bat ki de ket thuc chuong trinh...");
getch();
return 0;
}
```

Kết quả khi thực hiện:

```
Nhap vao 2 so nguyen duong a va b
a = 6
b = 9
Uoc chung lon nhat cua 6 va 9 la: 3
Nhap vao 2 so nguyen duong a va b
a = 15
b = 26
Uoc chung lon nhat cua 15 va 26 la: 1
Nhap vao 2 so nguyen duong a va b
a = 3
b = 0
Khong hop le
Nhap vao 2 so nguyen duong a va b
a = 0
b = 0
Khong hop le
An phim bat ki de ket thuc chuong trinh...
```

- Lưu ý:*
- Nếu có nhiều tham số trong danh sách tham số thì các tham số được phân cách với nhau bằng dấu phẩy
 - Cho dù hàm có một, nhiều hay không có tham số thì vẫn luôn luôn cần cặp dấu ngoặc đơn đứng sau tên hàm

Trong chương trình, khi gặp một lời gọi hàm thì hàm bắt đầu thực hiện bằng cách chuyển các lệnh thi hành đến hàm được gọi. Quá trình diễn ra như sau:

- Nếu hàm có tham số, trước tiên các tham số sẽ được *gán giá trị thực tương ứng*.
- Chương trình sẽ thực hiện tiếp các câu lệnh trong thân hàm bắt đầu từ lệnh đầu tiên đến câu lệnh cuối cùng.

```
        printf("\n a = %d",a);
    }
    printf("\n a = %d",a);
}
{
    int a = 3;
    printf("\n a = %d",a);
}
}
```

Kết quả thực hiện chương trình

```
a = 1
a = 2
a = 1
a = 3
```

Phân loại biến

Theo phạm vi sử dụng, biến chia làm 2 loại: biến cục bộ (biến địa phương – *local variable*) và biến toàn cục (*global variable*).

Biến địa phương

Là các biến được khai báo trong lệnh khởi hoặc trong thân chương trình con. Việc khai báo các biến cục bộ phải được đặt trước phần câu lệnh trong lệnh khởi hoặc trong chương trình con.

Biến toàn cục

Là biến được khai báo trong chương trình chính. Vị trí khai báo của biến toàn cục là sau phần khai báo tệp tiêu đề và khai báo hàm nguyên mẫu.

Lưu ý: Hàm **main()** cũng chỉ là một chương trình con, nhưng nó là chương trình con đặc biệt ở chỗ chương trình được bắt đầu thực hiện từ hàm **main()**.

Biến khai báo trong hàm **main()** không phải là biến toàn cục mà là biến cục bộ của hàm **main()**.

Một số lệnh đặc trưng của C: register, static

Chúng ta biết rằng các thanh ghi có tốc độ truy nhập nhanh hơn so với các loại bộ nhớ khác (RAM, bộ nhớ ngoài), do vậy nếu một biến thường xuyên sử dụng trong chương trình được lưu vào trong thanh ghi thì tốc độ thực hiện của chương trình sẽ được tăng lên. Để làm điều này ta đặt từ khóa **register** trước khai báo của biến đó.

Ví dụ

```
register int a;
```

Tuy nhiên có một lưu ý khi khai báo biến **register** là vì số lượng các thanh ghi có hạn và kích thước của các thanh ghi cũng rất hạn chế (ví dụ trên dòng máy 80x86, các thanh ghi có kích

thước 16 bit = 2 byte) cho nên số lượng biến khai báo **register** sẽ không nhiều và thường chỉ áp dụng với những biến có kích thước nhỏ như kiểu **char**, **int**.

Như ta đã biết, một biến cục bộ khi ra khỏi phạm vi của biến đó thì bộ nhớ dành để lưu trữ biến đó sẽ được giải phóng. Tuy nhiên trong một số trường hợp ta cần lưu giá trị của các biến cục bộ này để phục vụ cho những tính toán sau này, khi đó ta hãy khai báo biến với từ khóa **static** ở đầu.

Ví dụ

```
static int a;
```

Từ khóa **static** giúp chương trình dịch biết được đây là một biến tĩnh, nghĩa là nó sẽ được cấp phát một vùng nhớ thường xuyên từ lúc khai báo và chỉ giải phóng khi chương trình chính kết thúc. Như vậy về thời gian tồn tại biến **static** rất giống với biến toàn cục, chỉ có một sự khác biệt nhỏ là biến toàn cục thì có thể truy cập ở mọi nơi trong chương trình (miễn là ở đó không có biến địa phương nào cùng tên che lấp nó), còn biến **static** thì chỉ có thể truy nhập trong phạm vi mà nó được khai báo mà thôi.

Hãy xét ví dụ sau:

```
# include <stdio.h>
# include <conio.h>
void fct()
{
    static int count = 1;
    printf("\n Day la lan goi ham fct lan thu %2d", count++);
}
void main()
{
    int i;
    for(i = 0; i < 10; i++)
        fct();
    getch();
}
```

Kết quả khi thực hiện

```
Day la lan goi ham fct lan thu 1
Day la lan goi ham fct lan thu 2
Day la lan goi ham fct lan thu 3
Day la lan goi ham fct lan thu 4
Day la lan goi ham fct lan thu 5
Day la lan goi ham fct lan thu 6
Day la lan goi ham fct lan thu 7
Day la lan goi ham fct lan thu 8
Day la lan goi ham fct lan thu 9
Day la lan goi ham fct lan thu 10
```

Bài tập về Lập trình (2 tiết BT)

...



HOA SEN
UNIVERSITY

Giáo trình

TIN HỌC ĐẠI CƯƠNG

Biên soạn: ThS. Phạm Thị Thanh Tâm

ThS. Bùi Xuân Huy

TP. HCM, 09-2011

(Lưu hành nội bộ)

MỤC LỤC

PHẦN WINDOWS	1
I. GIỚI THIỆU	2
1. Các phiên bản của Windows 7	2
2. Yêu cầu phần cứng	2
II. SỬ DỤNG WINDOWS 7 CHO NGƯỜI DÙNG PHỔ THÔNG:.....	3
1. Các thành phần giao diện	3
2. Các tính năng của Taskbar:	4
3. Menu Start	6
4. Một số thành phần khác.....	8
III. CÁC TÍNH NĂNG CƠ BẢN	8
1. Themes	8
2. Desktop Background	9
3. Backup và Restore:.....	10
4. Libraries.....	14
5. Cài đặt font chữ cho Windows	14
6. Quản lý tập tin và thư mục trên Windows Explorer:	14
IV. CONTROL PANEL.....	17
1. Bitlocker drive Encryption.....	18
2. Date and Time	20
3. Region and Language.....	21
4. User Accounts	22
V. CÁC PHÍM TẮT TRÊN WINDOWS 7	23
PHẦN INTERNET	25
Chương 1: CÁC KHÁI NIỆM CƠ BẢN VỀ MẠNG MÁY TÍNH VÀ MẠNG INTERNET	26
I. GIỚI THIỆU VỀ MẠNG MÁY TÍNH.....	26
1. Khái niệm	26
2. Ưu điểm.....	26
3. Phân loại	26
4. Mô hình mạng	26
II. GIỚI THIỆU VỀ MẠNG INTERNET	27

1. Lịch sử	27
2. Khái niệm	27
3. Intranet và Extranet	27
4. Các dịch vụ trên Internet	27
5. Các thuật ngữ liên quan	28
6. Điều kiện để kết nối máy tính đến Internet	29
Chương 2: SỬ DỤNG TRÌNH DUYỆT WEB VÀ THƯ ĐIỆN TỬ	30
I. TRÌNH DUYỆT WEB	30
1. Trình duyệt web là gì ?.....	30
2. Các trình duyệt web hiện nay	30
3. Sử dụng trình duyệt web	30
II. THƯ ĐIỆN TỬ (EMAIL).....	31
1. Các khái niệm cơ bản	31
2. Một số nghi thức khi viết thư điện tử	31
3. Cơ chế phát hiện spam của các hệ thống thư điện tử thường là:.....	32
4. Nguy cơ bị nhiễm virus khi nhận thư điện tử, khi mở tệp đính kèm.....	32
Chương 3: TÌM KIẾM THÔNG TIN TRÊN INTERNET	33
I. MỞ ĐẦU.....	33
II. XÁC ĐỊNH NỘI DUNG TÌM KIẾM	33
III. CHỌN PHƯƠNG PHÁP VÀ CÔNG CỤ TÌM KIẾM	33
1. Đã biết địa chỉ.....	33
2. Tìm kiếm bằng Directory	33
3. Tìm kiếm bằng từ khóa.....	34
IV. TÌM KIẾM THÔNG TIN	35
1. Tìm kiếm chuẩn với Google.....	35
2. Sử dụng chức năng tìm kiếm nâng cao của Google	39
V. LƯU TRỮ THÔNG TIN	40
VI. VẤN ĐỀ BẢN QUYỀN	41
PHẦN MICROSOFT WORD 2010	42
Chương 1: TỔNG QUAN VỀ MICROSOFT WORD 2010.....	43
I. GIỚI THIỆU MICROSOFT WORD 2010	43
II. LÀM QUEN VỚI GIAO DIỆN MICROSOFT WORD 2010.....	43
III. HỆ THỐNG RIBBON	44

3. Tab File.....	44
4. Tab Home	45
5. Tab Insert.....	47
6. Tab Page Layout.....	47
7. Tab Reference.....	48
8. Tab Mailings.....	48
9. Tab Review.....	48
10. Tab View	48
11. Tab Developer	48
12. Tab Add-Ins.....	48
Chương 2: CÁC THAO TÁC CƠ BẢN	50
I. TẠO MỘT TÀI LIỆU MỚI.....	50
II. NHẬP VĂN BẢN.....	50
III. MỞ MỘT TÀI LIỆU CÓ SẴN	51
IV. LƯU TRỮ TÀI LIỆU	53
V. BẢO MẬT TÀI LIỆU.....	55
VI. ĐÓNG LẠI MỘT TÀI LIỆU	55
VII. CÁC CHẾ ĐỘ HIỂN THỊ TÀI LIỆU.....	56
VIII. CÁC THAO TÁC CƠ BẢN TRÊN KHỐI VĂN BẢN	56
1. Chọn khối văn bản.....	56
2. Hủy bỏ chọn khối.....	57
3. Xóa khối văn bản được chọn.....	57
4. Di chuyển khối văn bản (Cut)	57
5. Sao chép khối văn bản (Copy)	58
6. Sử dụng Undo và Redo.....	58
Chương 3: ĐỊNH DẠNG TÀI LIỆU	59
I. GIỚI THIỆU	59
II. ĐỊNH DẠNG KÝ TỰ.....	59
1. Định dạng ký tự.....	59
2. Tạo chữ hiệu ứng (Text Effect).....	63
3. Chèn ký hiệu (Symbol) vào văn bản	64
4. Tạo ký tự hoa lớn đầu đoạn (Drop Cap).....	66
5. Chuyển đổi khối văn bản thành chữ hoa và ngược lại	67

III. ĐỊNH DẠNG ĐOẠN.....	68
6. Thụt lề đoạn văn (Indentation)	68
7. Canh lề đoạn văn (Alignment)	71
8. Thay đổi khoảng cách đoạn và khoảng cách dòng (Paragraph Spacing and Line Spacing).....	73
9. Đánh dấu và đánh số đầu mục (Bullets and Numbering).....	76
10. Tạo danh sách đa cấp (Multilevel List).....	79
11. Sử dụng Tab trong văn bản	81
12. Đóng khung và tô nền cho văn bản (Boder and Shading).....	84
IV. ĐỊNH DẠNG TRANG	87
1. Khổ giấy (Paper Size).....	87
2. Lề trang (Margins).....	88
3. Dàn trang (layout)	90
V. ĐỊNH DẠNG MỘT PHẦN ĐOẠN – PHẦN TRANG.....	91
1. Tạo một phần đoạn:.....	91
2. Xóa bỏ một dấu ngắt phần đoạn.....	92
3. Phần trang.....	92
Chương 4: PHÂN TRANG – TIÊU ĐỀ ĐẦU TRANG, CUỐI TRANG	93
I. PHÂN TRANG	93
II. ĐÁNH SỐ TRANG ĐƠN GIẢN.....	94
III. TIÊU ĐỀ ĐẦU TRANG VÀ TIÊU ĐỀ CUỐI TRANG.....	94
IV. IN TÀI LIỆU.....	96
1. Chọn máy in	96
2. Xem lại văn bản trước khi in.....	97
3. In tài liệu.....	97
Chương 5: VĂN BẢN CỘT BÁO	99
I. MỞ ĐẦU.....	99
II. ĐỊNH DẠNG VĂN BẢN DẠNG NHIỀU CỘT (CỘT BÁO).....	99
III. CHÈN THÊM DẤU NGẮT CỘT.....	103
Chương 6: LÀM VIỆC VỚI BẢNG BIỂU	104
I. TẠO MỚI MỘT BẢNG	104
1. Tạo bảng bằng khung lưới Insert Table	105
2. Tạo bảng bằng lệnh Insert Table	105

3.	Tạo bảng bằng lệnh Draw Table	106
4.	Tạo bảng bằng lệnh Excel Spreadsheet	106
5.	Tạo bảng bằng lệnh Quick Tables	106
II.	CÁC THÀNH PHẦN CỦA BẢNG	107
III.	CÁC THAO TÁC TRÊN BẢNG BIỂU	108
1.	Di chuyển con trỏ trong bảng	108
2.	Nhập dữ liệu vào bảng	108
3.	Các thao tác chọn bảng	108
4.	Điều chỉnh chiều rộng cột và chiều cao dòng	109
5.	Chèn thêm dòng, cột	110
6.	Xóa cột, dòng, bảng	111
7.	Xóa bỏ ô	111
8.	Trộn nhiều ô thành một ô	111
9.	Tách một ô thành nhiều ô	111
10.	Tách một bảng thành hai bảng	112
11.	Hiệu chỉnh bảng	112
12.	Thêm đường viền và tô nền cho bảng	113
13.	Định dạng đoạn bên trong bảng	115
14.	Chuyển đổi qua lại giữa văn bản và bảng	116
IV.	SẮP XẾP THỨ TỰ TRONG BẢNG	117
V.	TÍNH TOÁN TRONG BẢNG	118
Chương 7: LÀM VIỆC VỚI CÁC ĐỐI TƯỢNG ĐỒ HỌA TRONG VĂN BẢN		120
I.	GIỚI THIỆU	120
II.	CHÈN VÀ HIỆU CHỈNH HÌNH VẼ (SHAPES)	121
1.	Chèn hình vẽ	121
2.	Thêm văn bản vào hình vẽ (Add Text)	122
3.	Hiệu chỉnh hình vẽ	122
III.	CHÈN VÀ HIỆU CHỈNH CHỮ NGHỆ THUẬT (WORDART)	129
1.	Chèn WordArt	130
2.	Hiệu chỉnh WordArt	131
IV.	CHÈN VÀ HIỆU CHỈNH HÌNH ẢNH (PICTURE)	137
1.	Chèn hình ảnh	137

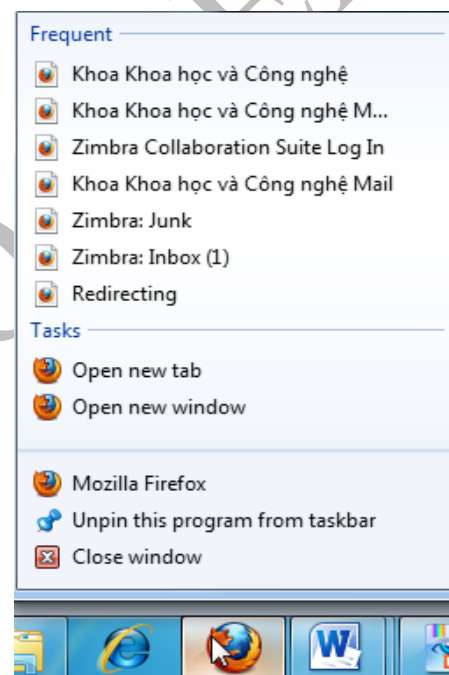
2. Hiệu chỉnh hình ảnh	137
V. CHÈN VÀ HIỆU CHỈNH CLIPT ART.....	140
VI. CHÈN VÀ HIỆU CHỈNH SƠ ĐỒ (SMARTART)	141
1. Chèn sơ đồ.....	142
2. Hiệu chỉnh sơ đồ.....	143
VII. CHÈN VÀ HIỆU CHỈNH BIỂU ĐỒ (CHART)	145
3. Chèn biểu đồ.....	145
4. Hiệu chỉnh biểu đồ	146
VIII. LẬP VÀ HIỆU CHỈNH BIỂU THỨC TOÁN HỌC (EQUATATION)..	147
Chương 8: TRỘN VÀ IN THƯ (MAIL MERGE).....	151
I. GIỚI THIỆU	151
II. TRỘN VÀ IN THƯ VỚI MAIL MERGE WIZARD	152
III. TRỘN VÀ IN THƯ BẰNG TAB MALLINGS	158
Chương 9: MỘT SỐ TÍNH NĂNG MỚI TRÊN WORD 2010	160
I. Lưu dưới dạng file PDF	160
II. Chụp ảnh cửa sổ và chèn vào văn bản (tab Insert > Screenshot).....	161
III. Cải tiến chức năng tìm kiếm	161
IV. Paste Preview (tab Home > Paste)	162
V. Cover Page (tab Page Layout > Cover Page).....	162
VI. Watermark.....	162
VII. Compare và Combine document (tab Review > Compare)	163
VIII. Quick Parts (tab Insert > Quick Parts).....	164
Bài tập thực hành	165

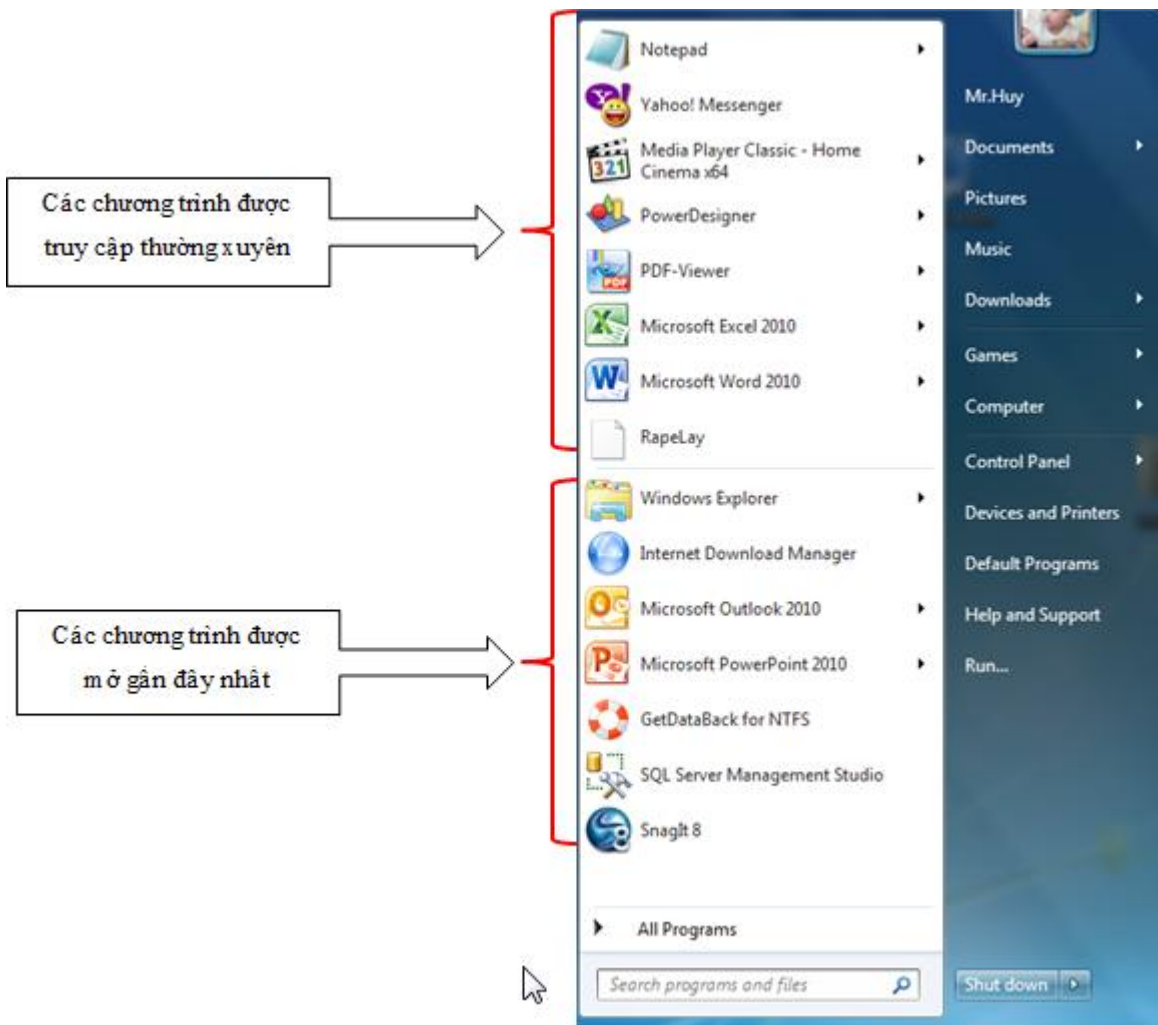


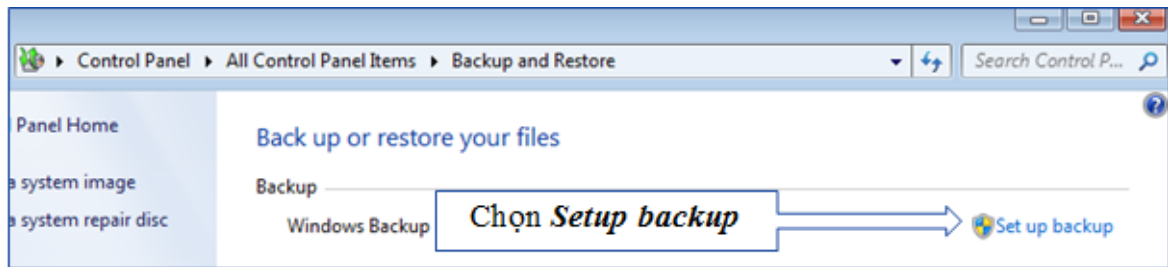
Jumplist: là tính năng mới trong Windows 7. Xem *jumplist* của chương trình bằng cách click phải chuột vào biểu tượng của chương trình đó trên *taskbar*. Các thông tin được hiển thị bao gồm các file được mở gần đây nhất, mở cửa sổ mới, đóng chương trình. Ví dụ: đối với Firefox, *jumplist* sẽ hiển thị các trang web mới mở, đối với Windows media player, *jumplist* sẽ hiển thị các bài hát bạn nghe gần đây nhất. Điều này giúp bạn nhanh chóng tìm được thứ mình vừa mới làm việc với nó ngày hôm qua.

Với tính năng Pin, bạn có thể cố định một file để file đó luôn luôn xuất hiện trong *jumplist*. (Lưu ý: mục *frequent* chỉ hiển thị những file mở gần đây nhất và luôn luôn bị thay đổi, còn mục *pinned* thì cố định cho đến khi bạn gỡ bỏ nó thì thôi)

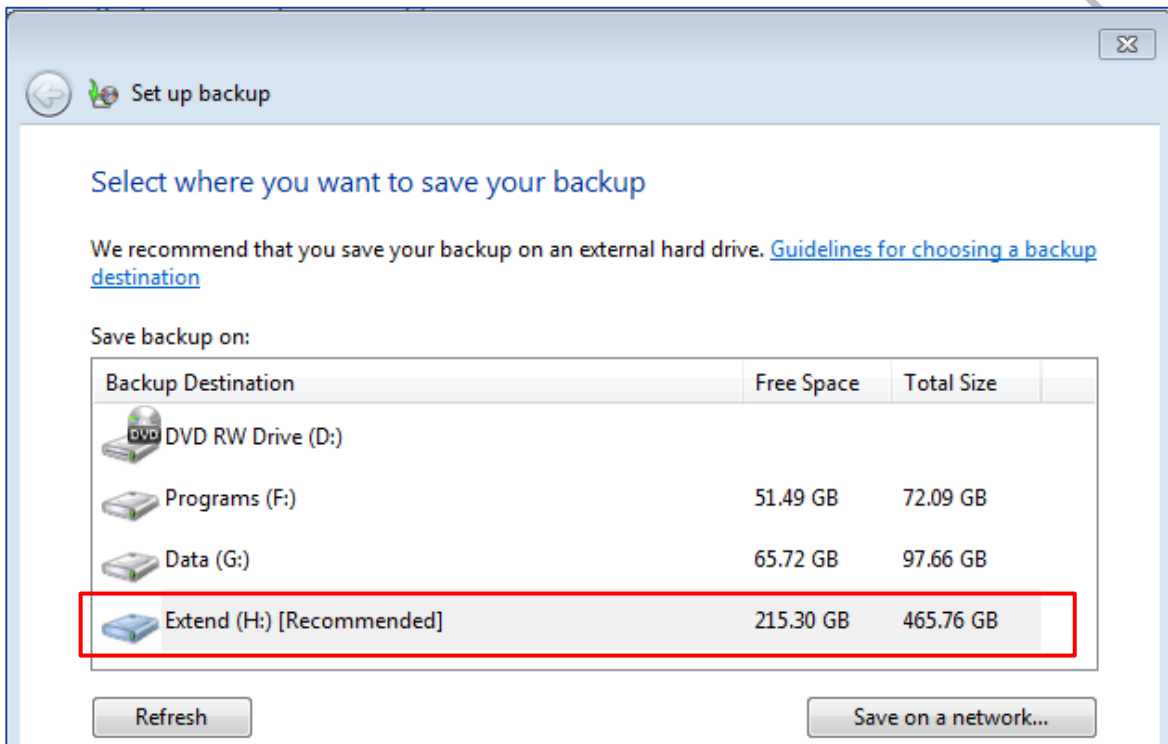
Cách thực hiện: click chuột phải vào chương trình muốn pin, chọn *Pin to this list*. Muốn gỡ bỏ một chương trình thì chọn *Unpin from this list*.





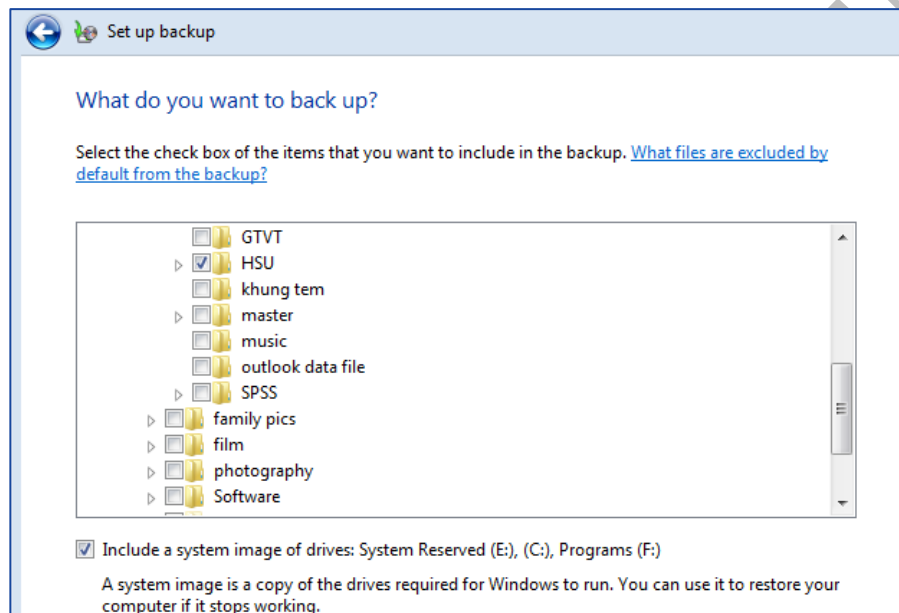
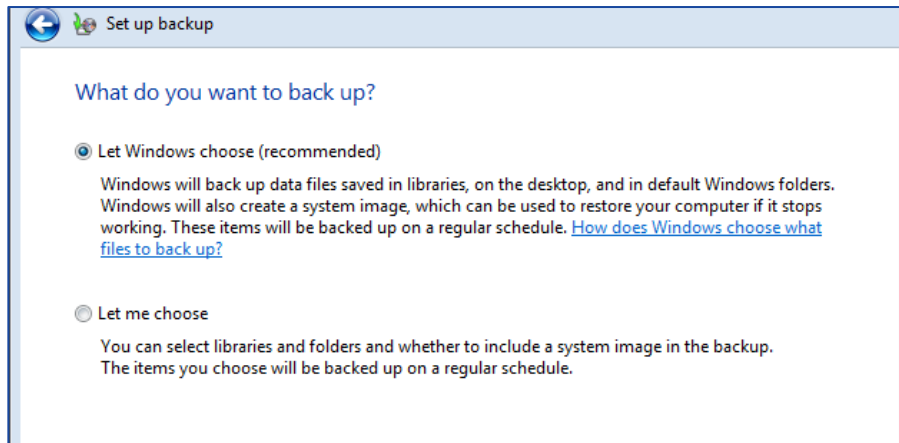


Tiếp theo, chọn ổ đĩa để lưu file backup của bạn. Một kinh nghiệm khôn ngoan là bạn nên lưu file này vào một ổ đĩa gắn ngoài. Vì nếu hệ thống bị lỗi, rất nhiều khả năng ngay cả file backup của bạn cũng bị mất.

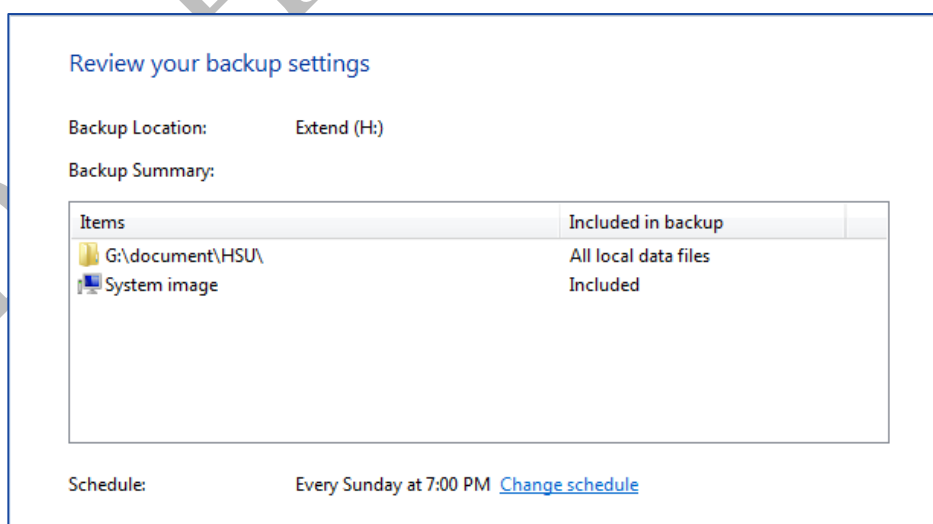


Tiếp theo, Windows 7 đưa ra cho bạn hai lựa chọn

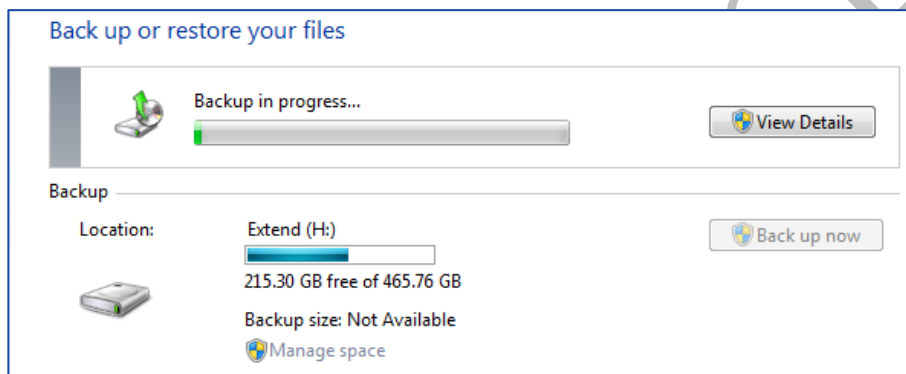
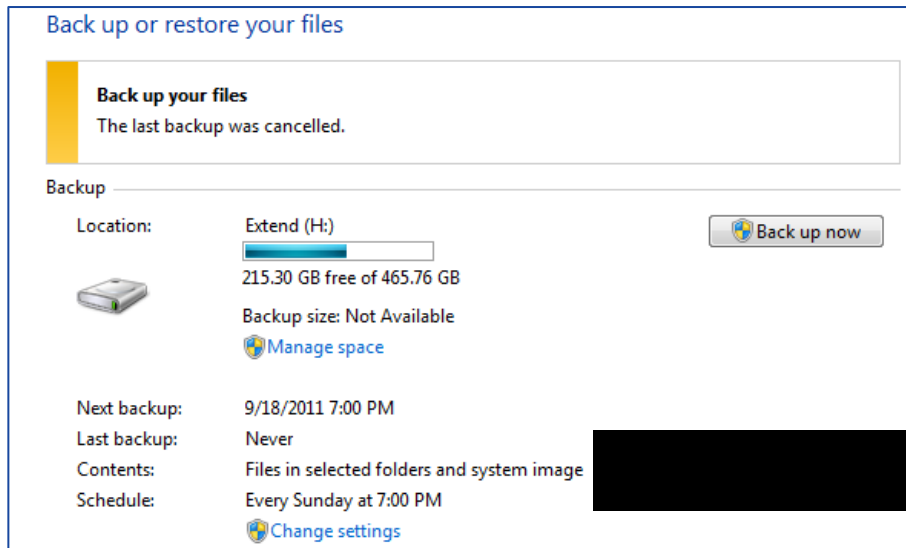
- **Let Windows choose:** Để Windows tự quyết định sẽ sao lưu cái gì. (Với lựa chọn này, Windows sẽ chỉ sao lưu các thư mục hệ thống).
- **Let me choose:** Tự bạn chọn file hoặc folder của mình để sao lưu. Sử dụng lựa chọn này khi bạn muốn sao lưu những tập tin cá nhân của mình.



Đánh dấu check vào mục ***Include a system image of drives***, ngoài thư mục cá nhân của bạn, Windows sẽ tự động sao lưu toàn bộ đĩa cứng.

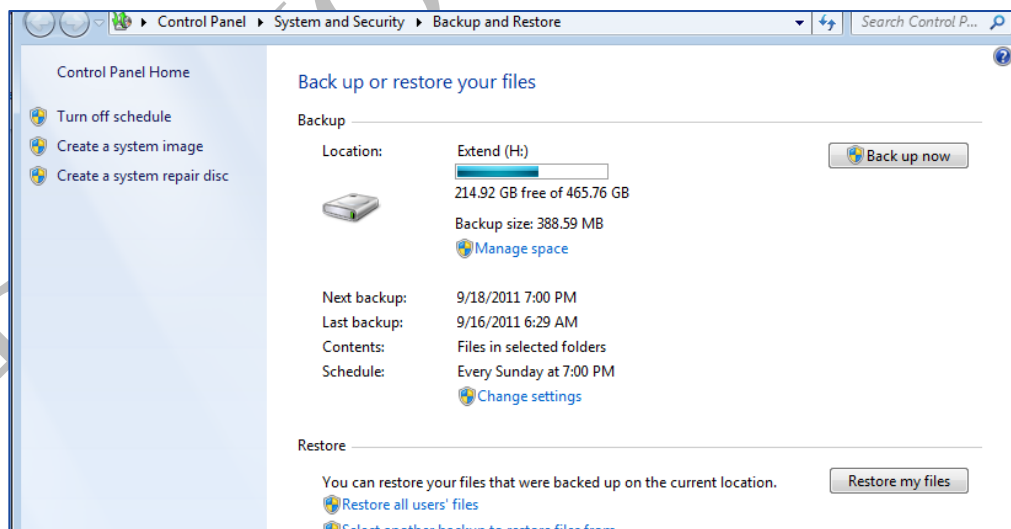


Click vào *Change schedule* để lập lịch cho Windows 7 tự động sao lưu.



b. Restore

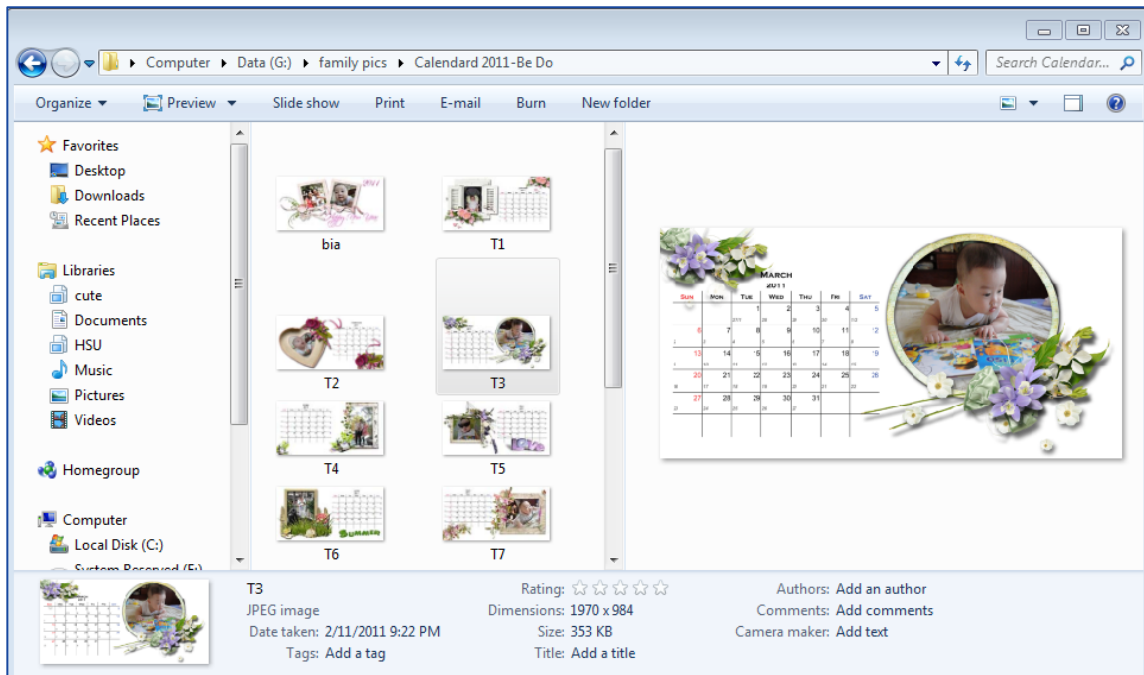
Trở lại mục **Control Panel > System and Security > Back up your computer**



Chọn **Restore my file** để phục hồi dữ liệu. Trong trường hợp có nhiều file backup khác nhau thì bạn chọn **Select another backup to restore files from**.

Nếu muốn tắt chức năng cập nhật tự động, chọn **Turn off schedules**

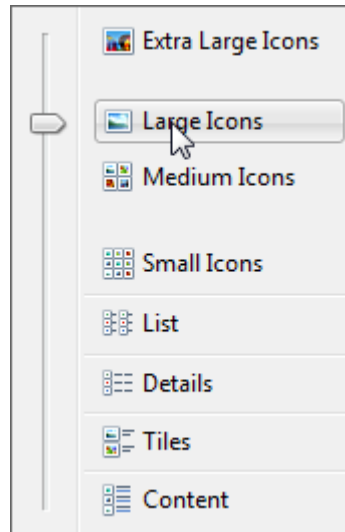
máy tính nối mạng trong nhà bạn. Một số tiện ích đặc trưng mà chương trình này đem lại là sắp xếp các tập tin và thư mục theo nhiều tiêu chí khác nhau, xem trước nội dung tập tin, lưu giữ địa chỉ trên thanh address để tiết kiệm thời gian di chuyển giữa các thư mục, chỉ báo tình trạng của dung lượng ổ đĩa, ghi trực tiếp tập tin và thư mục ra đĩa CD/DVD, chia sẻ dữ liệu với các máy tính trong mạng cục bộ, tìm kiếm tập tin và thư mục



a. Các chế độ xem:

Windows 7 hỗ trợ thao tác chuyển đổi chế độ xem nhanh chóng. Bạn có thể click chọn vào 8 chế độ xem được liệt kê trong menu bên hoặc dùng chuột kéo thanh cuộn để thay đổi kích thước hiển thị của tập tin.

Xem trước nội dung file với *preview pane*



b. Các lệnh cơ bản về thư mục và tập tin:

Hệ thống thư mục trong Windows 7 được tổ chức theo dạng cây như hình bên

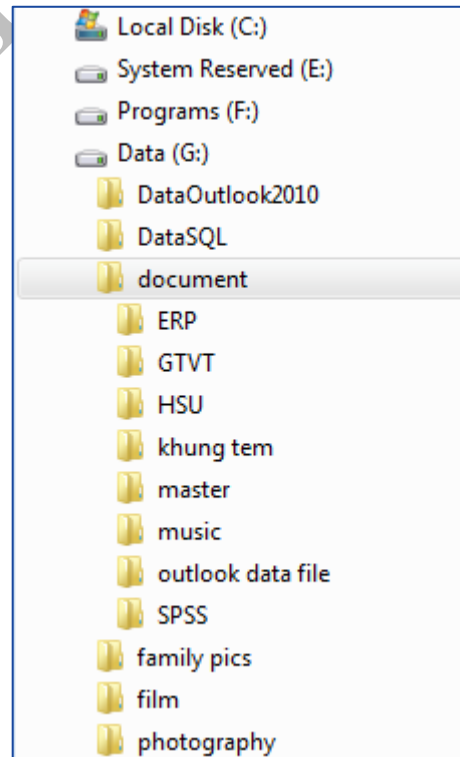
Cách tạo thư mục: Chọn ổ đĩa và thư mục chứa thư mục mới. Click vào *New folder* dưới thanh *address bar*. Nhập tên thư mục mới.



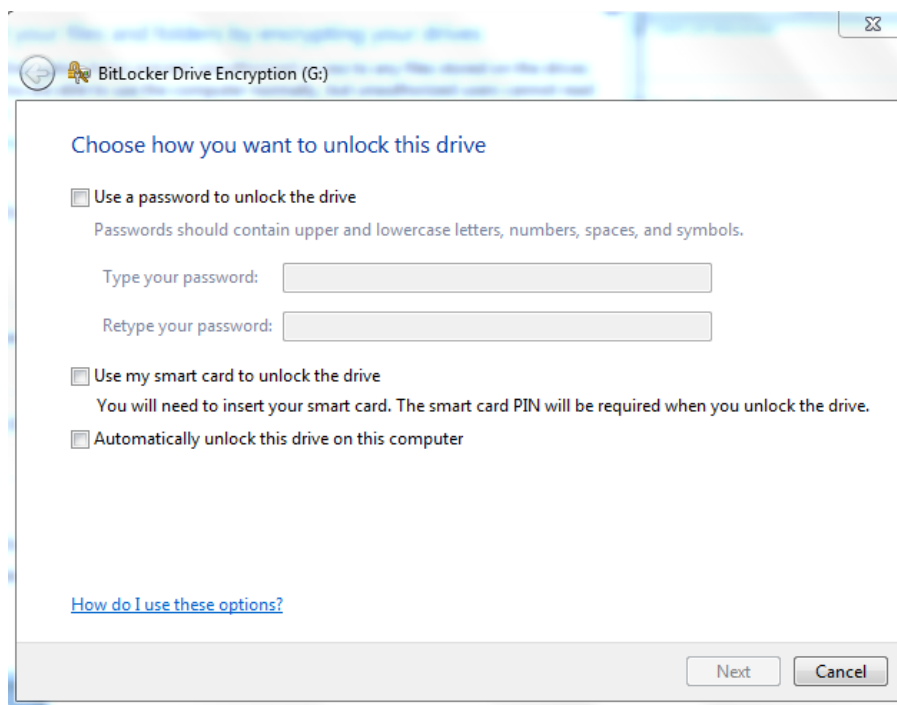
Đưa một thư mục vào library: Chọn thư mục đó và click vào *Include in library*.

Bạn có thể tự tìm hiểu thêm về cách thực hiện qua menu ngữ cảnh (click chuột phải lên tập tin/thư mục để mở menu ngữ cảnh)

Các thao tác khác trên thư mục/tập tin (đổi tên/sao chép/di chuyển/xóa) đều được thực hiện qua menu ngữ cảnh. Đặc biệt đối với thao tác xóa, bạn có thể chọn thư mục/tập tin cần xóa và bấm phím *Delete* trên bàn phím. Đối với thao tác copy thì dùng tổ hợp phím Ctrl+C và sao chép sang thư mục mới bằng phím Ctrl+V



Cách tạo shortcut: Shortcut (lối tắt) cho phép người dùng mở nhanh một thư mục/tập tin dù thư mục/tập tin đó được lưu trữ ở bất kỳ vị trí nào. Khi đổi tên hoặc xóa



- ***Use a password to unlock the drive:*** dùng mật khẩu để bảo vệ dữ liệu.
- ***Use my smart card to unlock the drive:*** dùng thẻ nhớ để bảo vệ dữ liệu. Đối với tùy chọn này, các thông tin về ổ đĩa sẽ được lưu vào một thẻ nhớ. Mỗi khi cần mở ổ đĩa để xem dữ liệu cần phải dùng chính thẻ nhớ này.
- ***Automatically unlock this drive on this computer:*** tự động unlock ổ đĩa này mỗi khi bạn đăng nhập vào máy tính.

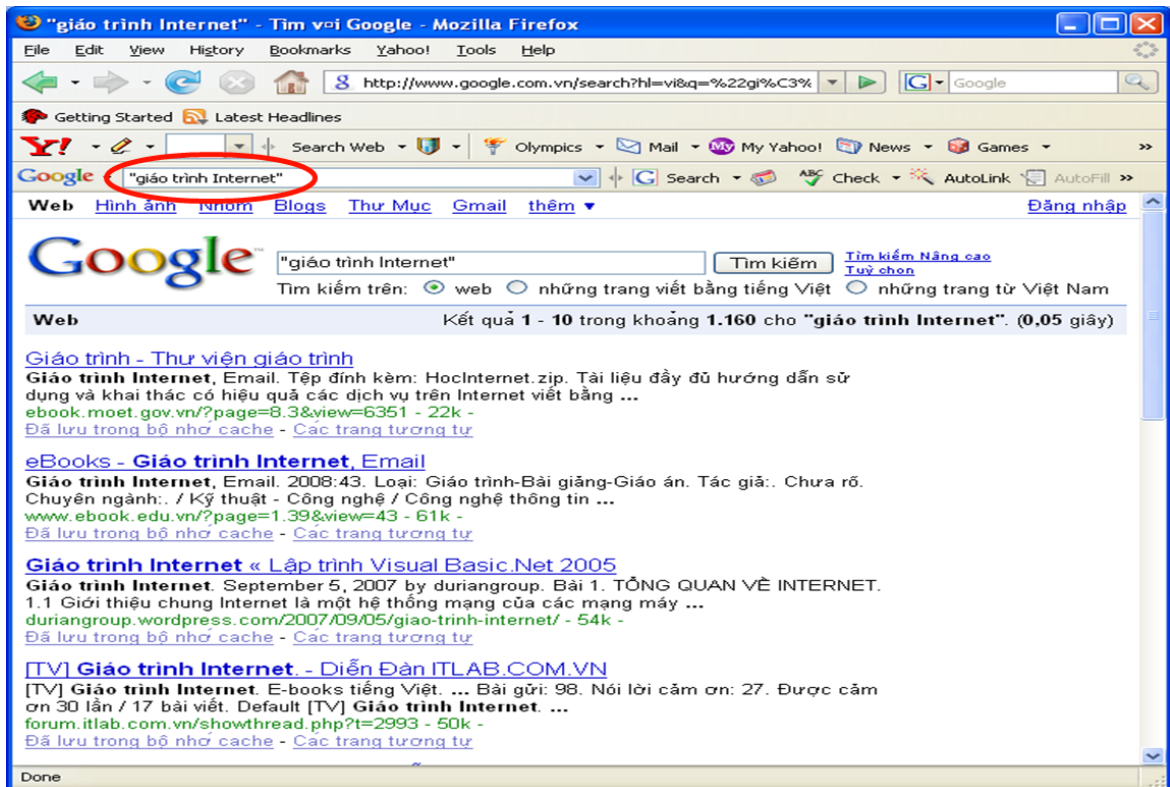
Tiếp theo, Windows sẽ yêu cầu bạn tạo một *recovery key*. Recovery key được lưu dưới dạng một file, được sử dụng khi người dùng quên mật khẩu hoặc mất thẻ nhớ không truy cập được vào ổ đĩa.

Windows không cho phép người dùng lưu recovery key trên ổ đĩa được mã hóa. Trong trường hợp ổ đĩa đó nằm trên đĩa cứng của máy, người dùng cũng không được lưu recovery key lên bất kỳ ổ đĩa nào của đĩa cứng đó.

Quá trình mã hóa nhanh hay chậm phụ thuộc vào kích thước dữ liệu trên ổ đĩa được mã hóa.

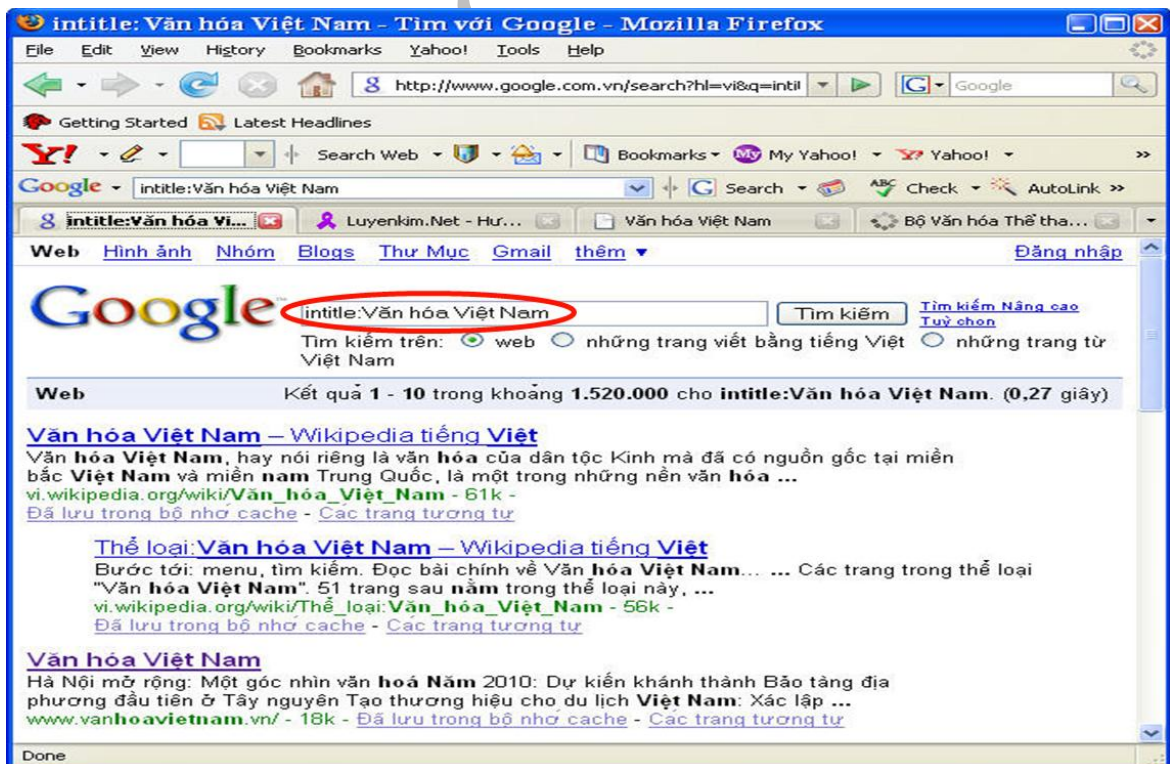
⊞ + Spacebar	Làm trong suốt tất cả các cửa sổ để bạn có thể nhìn thấy màn hình desktop
⊞ + Pause/Break	Mở hộp thoại System Properties
⊞ + B	Di chuyển về thanh taskbar (Thanh dưới cùng bên phải)
⊞ + Tab	Mở Aero 3D [Ấn nút Tab để thay đổi các cửa sổ]
⊞ + D	Hiện/ Ẩn desktop
⊞ + E	Bật Window Explorer
⊞ + F	Tìm kiếm
⊞ + G	Đưa tất cả gadgets lên đầu
⊞ + L	Khóa máy
⊞ + M	Thu nhỏ tất cả các cửa sổ
⊞ + P	Mở menu cho máy chiếu
⊞ + T	Hiện thumbnail của các ứng dụng đang chạy dưới thanh Taskbar mà không cần di chuột qua
⊞ + =	Phóng to màn hình
⊞ + Shift + ↑	Chỉnh tối đa kích thước cửa sổ theo chiều dọc
⊞ + Shift + ↓	Khôi phục kích thước cửa sổ theo chiều dọc
⊞ + Shift + ←	Chuyển sang màn hình trái
⊞ + Shift + →	Chuyển sang màn hình phải
Shift + Del	Xóa vĩnh viễn thư mục/tập tin
Ctrl+Shift+N	Tạo thư mục mới
F2	Đổi tên thư mục
F3	Tìm kiếm

b. Tìm chính xác nhóm từ với dấu “ ”

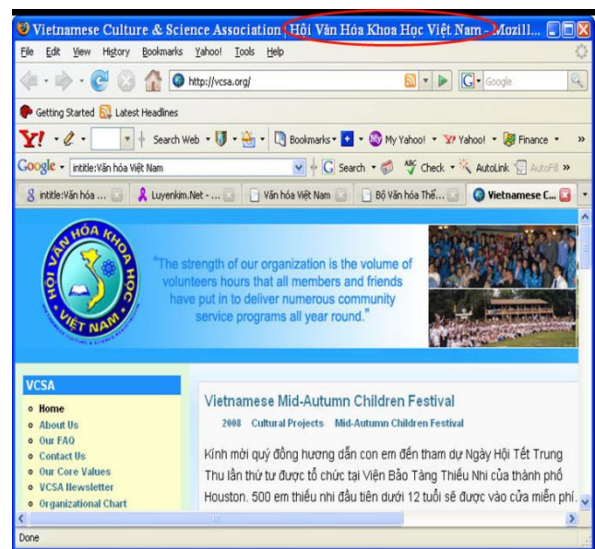
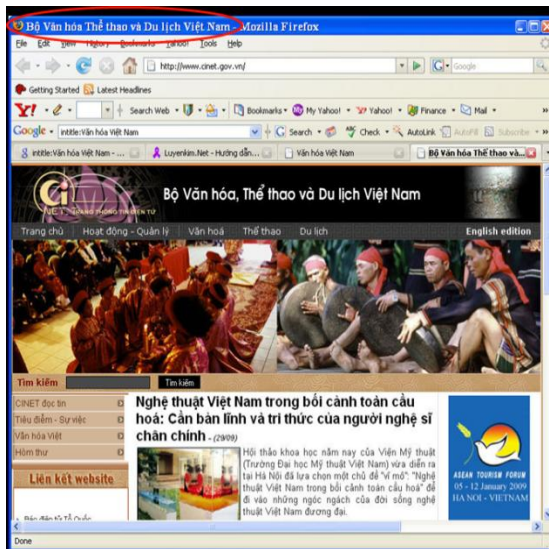


c. Tìm nhóm ký tự chứa trong tiêu đề của trang web

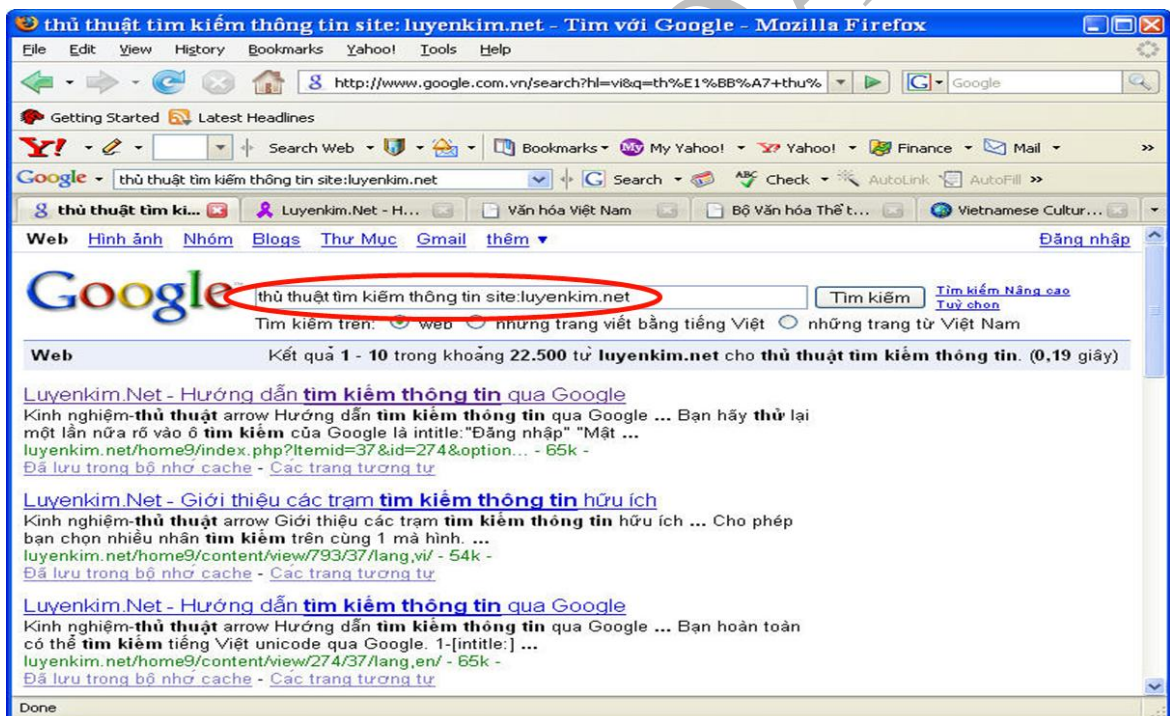
intitle:nhóm từ cần tìm



Kết quả tìm kiếm bằng từ khóa **intitle:Văn hóa Việt Nam**

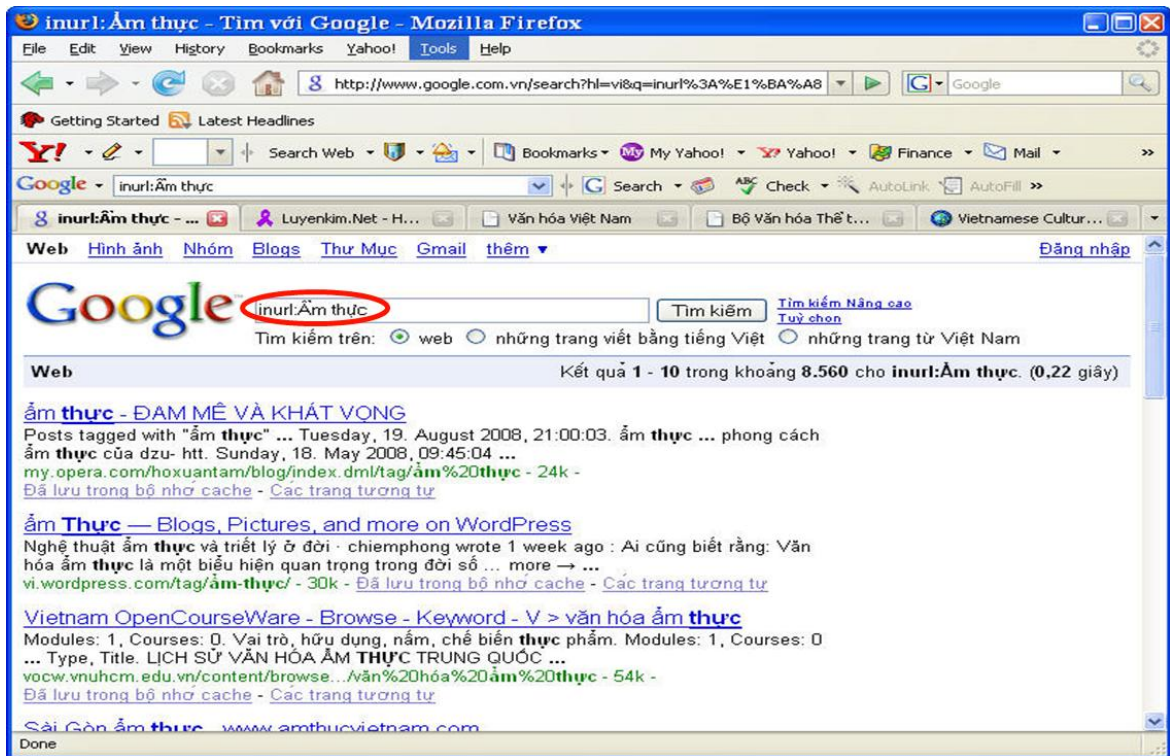


d. Tìm kiếm ký tự chứa trong phạm vi tên miền nhóm từ cần tìm site:*tên miền*



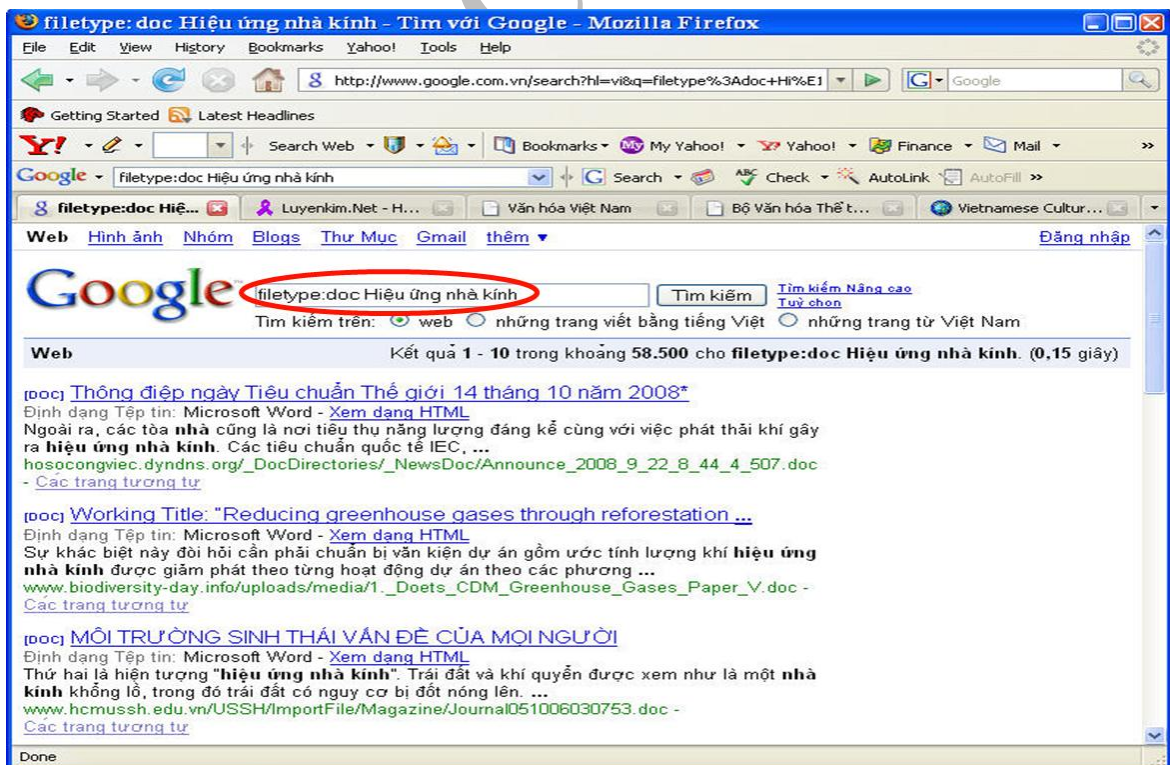
e. Tìm kiếm ký tự chứa trong URL













*inurl:*nhóm từ cần tìm



f. Tìm kiếm theo phần mở rộng của file








*filetype:*ext nhóm từ

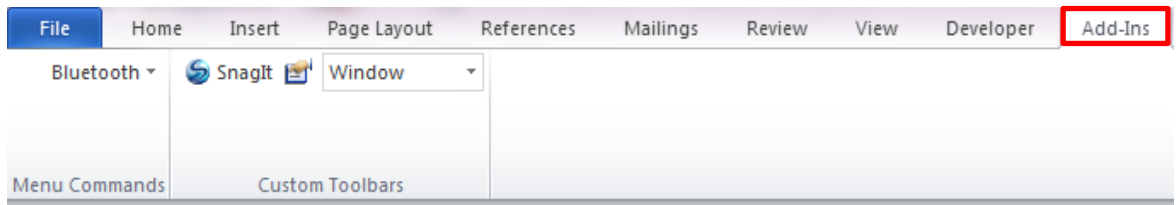


	Bold (Ctrl + B): in đậm phần văn bản được chọn.
	Italic (Ctrl + I): in nghiêng phần văn bản được chọn.
	Underline (Ctrl + U): gạch dưới phần văn bản được chọn.
	Strikethrough : tạo đường ngang dòng văn bản được chọn.
	Subscript (Ctrl + =): tạo chữ nhỏ bên dưới chữ.
	Superscript (Ctrl + Shift + =): tạo chữ nhỏ bên trên chữ.
	Grow Font (Ctrl +]): tăng kích thước font chữ.
	Shrink Font (Ctrl + [): giảm kích thước font chữ.
	Change Case : thay đổi tất cả văn bản được chọn thành chữ in hoa, chữ thường hoặc chữ hoa đầu từ.
	Text Effect : áp dụng hiệu ứng lên văn bản được chọn.
	Text Highlight Color : đánh dấu văn bản bằng cách tô vệt sáng.
	Font Color : chọn màu chữ cho văn bản được chọn.
	Clear Formatting : xóa các định dạng lên văn bản được chọn.

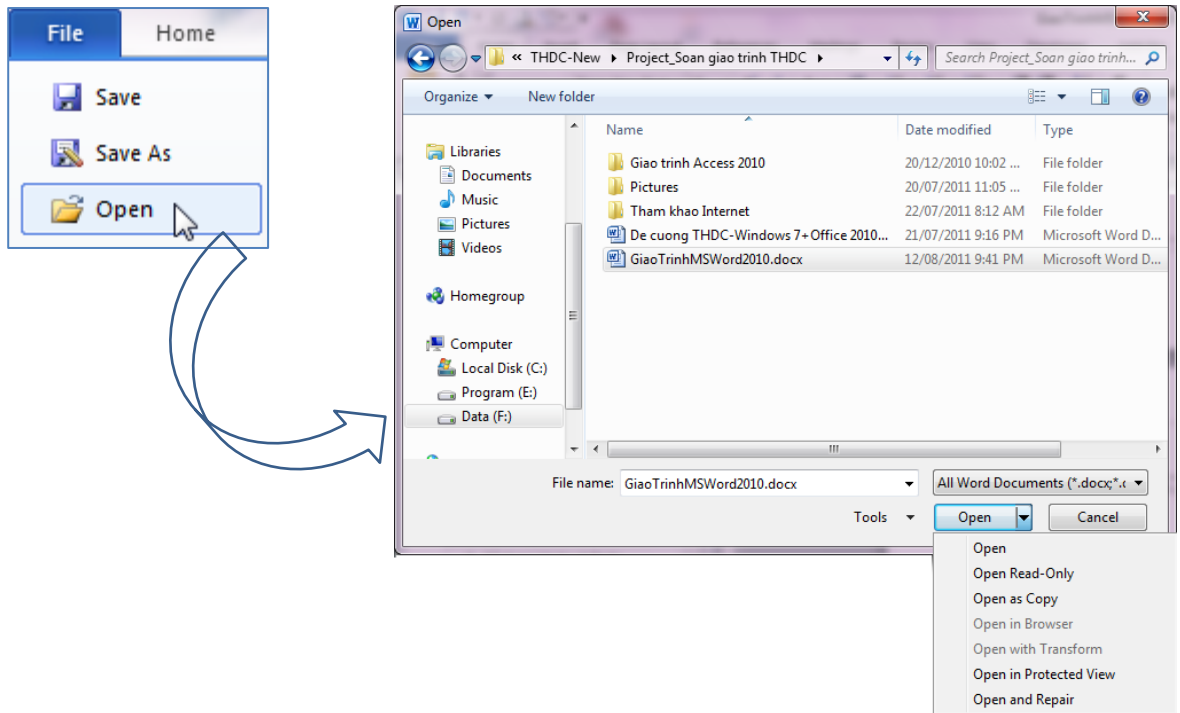
c. Nhóm lệnh Paragraph – định dạng đoạn văn bản

Dùng để định dạng nhanh các định dạng thông thường cho các đoạn văn bản như: canh lề (Alignment), đánh dấu đầu đoạn (Bullets), đánh số tự động (Numbering), tăng/giảm khoảng cách thụt dòng (Decrease/Increase Indent), bật/tắt Border Toolbar (Borders), ...

	Bullets : tự động đánh dấu đầu mục các đoạn văn bản đã chọn
	Numbering : tự động đánh số đầu mục các đoạn văn bản đã chọn
	Multilevel List : phân cấp các đoạn văn bản
	Decrease Indent : giảm mức thụt đầu dòng của đoạn bản
	Increase Indent : tăng mức thụt đầu dòng của đoạn văn bản
	Sort : sắp xếp dữ liệu theo mẫu tự ABC hoặc theo số
	Show/Hide : bật/tắt hiển thị của các ký tự định dạng văn bản như



ĐẠI HỌC HOA SEN

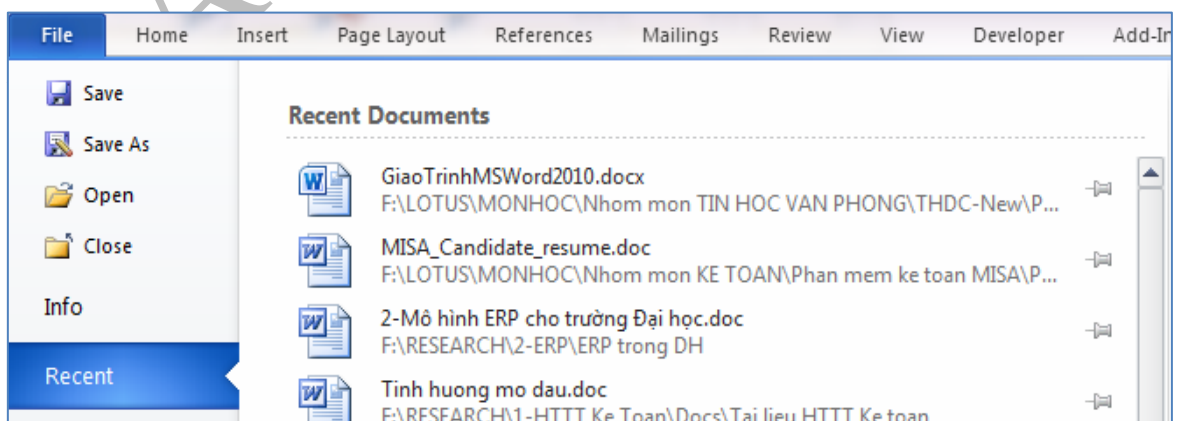


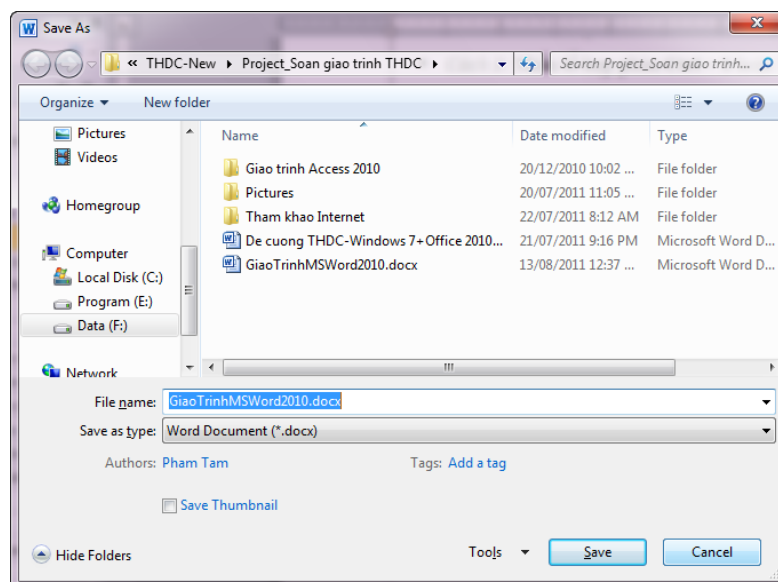
Tại nút **Open** có các tùy chọn:

- **Open**: mở bình thường
- **Open Read-Only**: không được lưu thay đổi.
- **Open as Copy**: tạo bản sao của file và mở ra.
- **Open in Browser**: mở file bằng trình duyệt web mặc định.
- **Open in Protect View**: mở ở chế độ bảo vệ
- **Open and Repair**: mở trong trường hợp file bị lỗi

Cách 2:

- Click chọn tab **File** > chọn **Recent** (mở các file được soạn thảo gần đây nhất)
- Chọn tên file trong danh sách **Recent**



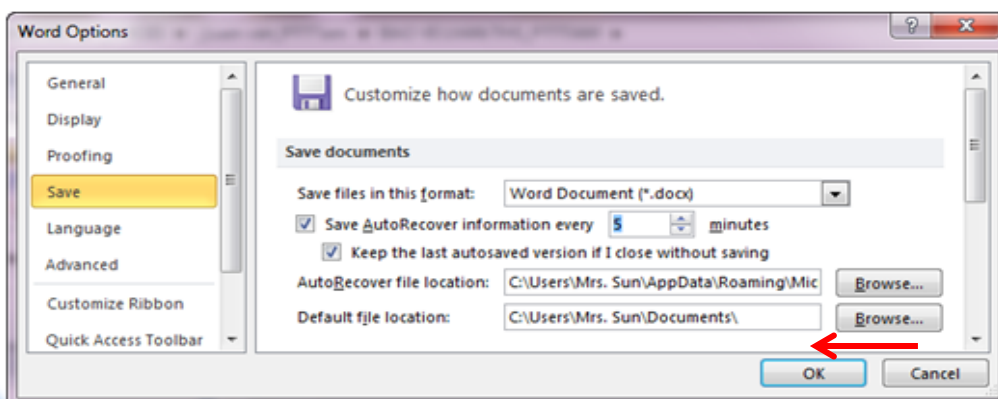










Lưu ý: Quy tắc đặt tên file trong Word cũng giống như quy tắc đặt tên file trong Windows. Tên file dài tối đa 255 ký tự bao gồm cả khoảng trắng. Tuy nhiên, trong tên file không được dùng các ký hiệu như: \ ? : * “ > <.

Ghi chú: Nhằm bảo toàn dữ liệu, bạn nên bật tính năng Auto Recover, Word sẽ tự động thực hiện lệnh lưu theo thời gian quy định (mặc định là 10 phút lưu một lần).

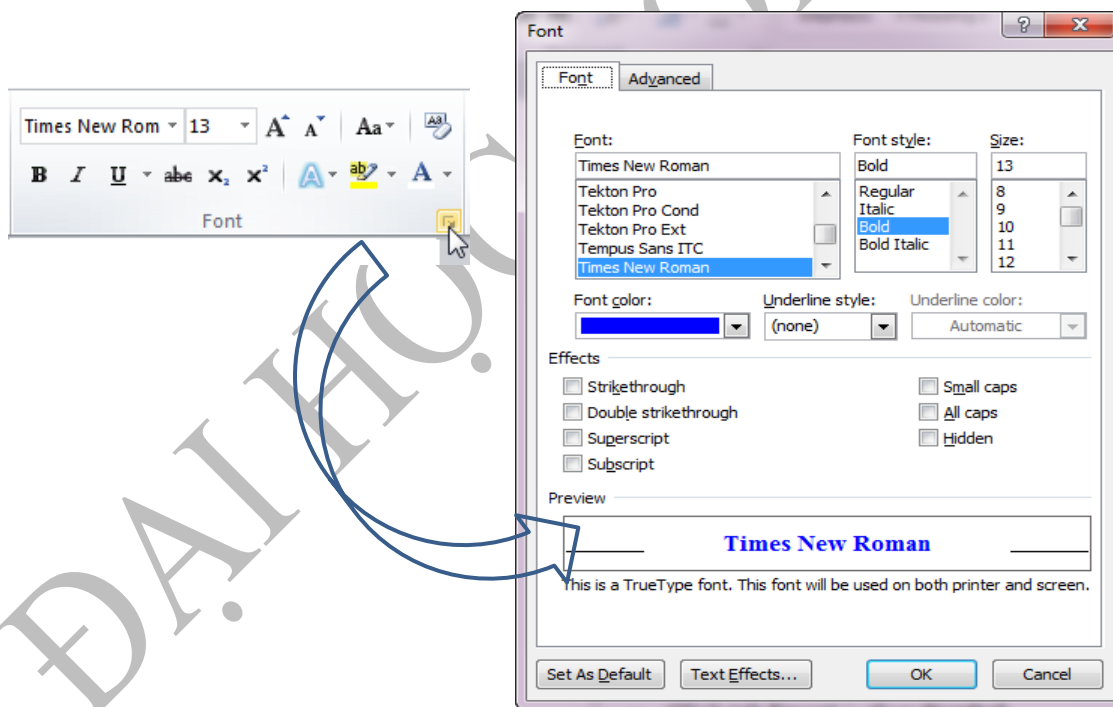
Để sử dụng tính năng này, bạn tiến hành thực hiện như sau:

- Click chọn tab **File** > chọn **Options**. Hộp thoại **Word Options** xuất hiện.
- Click chọn **Save**, đánh dấu vào ô **Save AutoRecover information every** và nhập vào số phút bạn muốn ấn định thời gian tự động lưu, thường giới hạn trong khoảng từ 1 đến 120 phút.



	Tạo đường ngang dòng văn bản được chọn.
 (Ctrl + =)	Tạo chỉ số dưới (chữ nhỏ bên dưới chữ)
 (Ctrl + Shift + =)	Tạo chỉ số trên (chữ nhỏ bên trên chữ)
 (Ctrl + Shift + >)	Tăng kích thước font chữ.
 (Ctrl + Shift + <)	Giảm kích thước font chữ.
	Đánh dấu văn bản bằng cách tô vệt sáng.
	Chọn màu chữ cho văn bản được chọn.
	Xóa các định dạng lên văn bản được chọn.

Cách 2: Click vào mũi tên bên phải nhóm lệnh **Font** để mở ra hộp thoại **Font** (phím tắt là **Ctrl + D**)

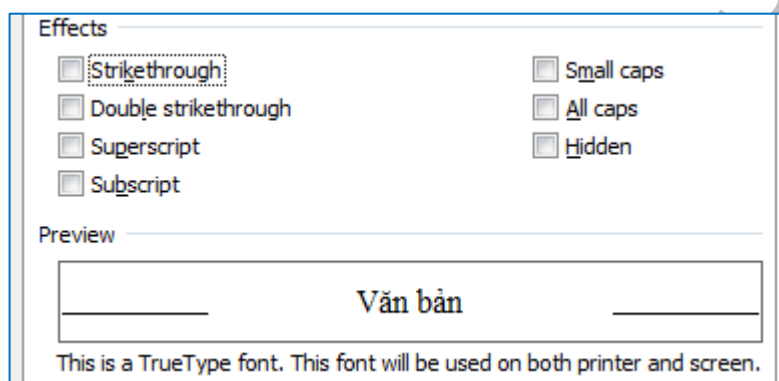


Trong hộp thoại **Font**, chọn tab **Font** để định dạng ký tự. Các thông số trong tab này bao gồm:

Mục chọn	Ý nghĩa
Font	Chọn Font chữ
Font Style	Chọn kiểu chữ, bao gồm

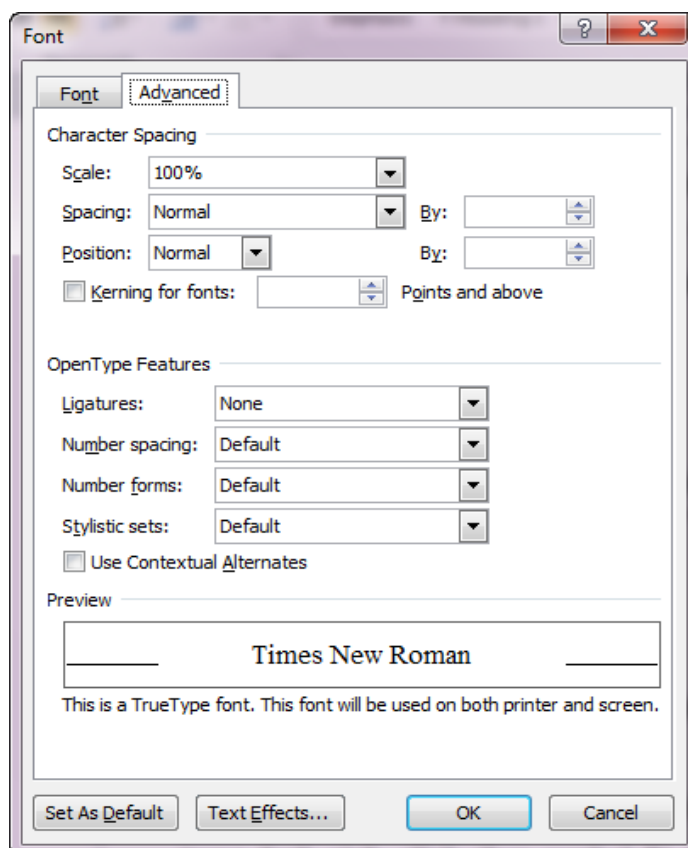
	<ul style="list-style-type: none"> ▪ Regular: bình thường ▪ Italic: <i>ngiên</i> ▪ Bold: đậm ▪ Bold Italic: <i>đậm nghiêng</i>
Size	Xác định kích thước chữ cho văn bản được chọn
Font color	Chọn màu chữ cho văn bản được chọn
Underline style	Chọn kiểu đường gạch chân
Underline color	Chọn màu cho đường gạch chân

Khi định dạng văn bản, ngoài việc thay đổi font chữ, kiểu chữ, kích thước, màu sắc cho chữ,... bạn có thể sử dụng một hiệu ứng (Effect) cho chữ để làm thêm sự chú ý hay để biểu diễn những ký tự đặc biệt



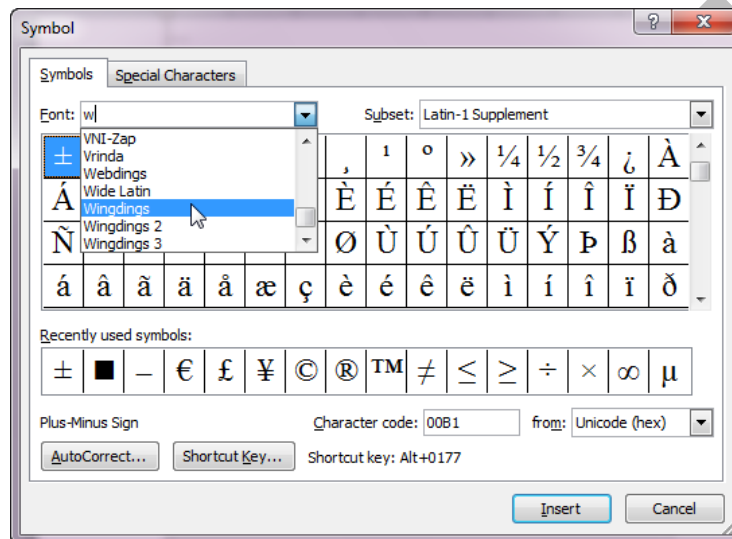
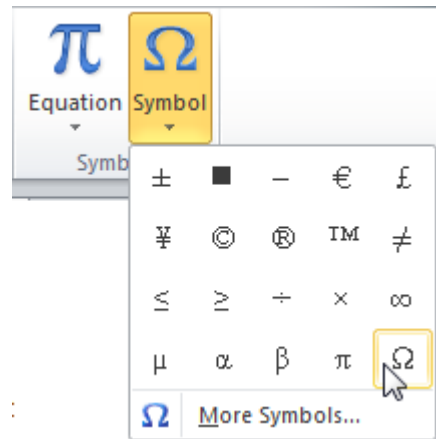
Hiệu ứng	Ý nghĩa
Strikethrough	Tạo một đường gạch ngang văn bản
Double strikethrough	Tạo hai đường gạch ngang văn bản
Super ^{script}	Chỉ số trên (chữ nhỏ bên trên chữ)
Sub _{script}	Chỉ số dưới (chữ nhỏ bên dưới chữ)
SMALL CAPS	Mọi ký tự đều được in hoa, nhưng ký tự đầu trong mỗi từ có kích thước bình thường, trong khi các ký tự còn lại có kích thước nhỏ hơn
ALL CAPS	Mọi ký tự đều được in hoa với chiều cao bằng nhau
Hidden	Ẩn dấu tạm thời phần văn bản được chọn, không hiển thị trên màn hình. Để hiển thị lại phần văn bản đã được ẩn dấu, click bỏ chọn Hidden

Với tab **Advanced** trong hộp thoại **Font**, bạn có thể thay đổi khoảng cách giữa các ký tự trong văn bản hoặc có thể làm cho các ký tự “nhích lên nhích xuống” theo chiều dọc so với *đường dẫn văn bản (Text Base Line)* thông thường.

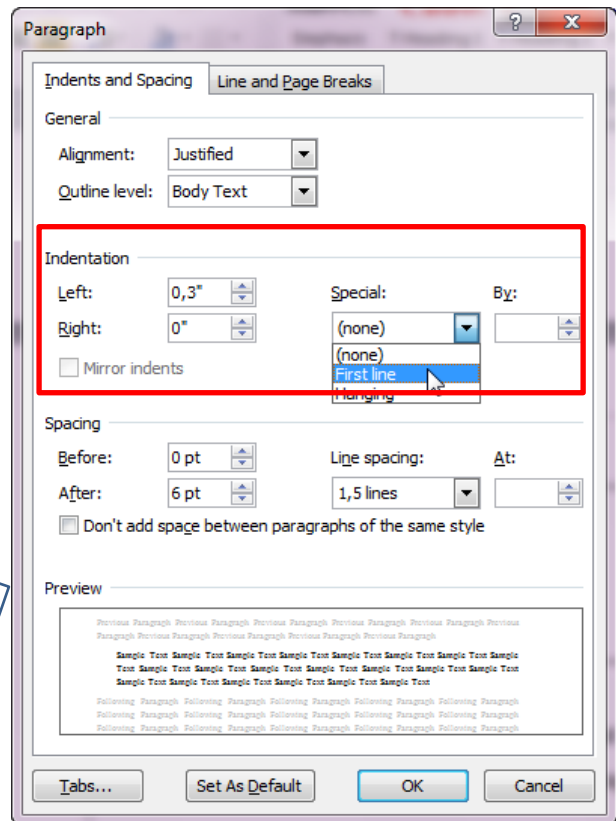
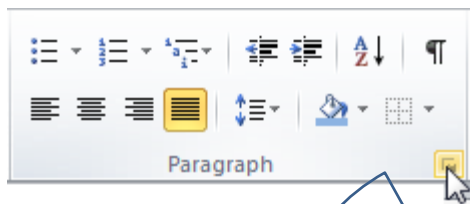


Mục chọn	Ý nghĩa
Scale	Cơ giãn tỉ lệ của chữ theo phần trăm
Spacing	Khoảng cách ngang giữa các ký tự <ul style="list-style-type: none"> ▪ Expanded: tăng khoảng cách ngang giữa các ký tự ▪ Condensed: giảm khoảng cách ngang giữa các ký tự
Position	Vị trí dọc so với Text Base Line <ul style="list-style-type: none"> ▪ Normal: đặt văn bản nằm đúng theo Text Base Line ▪ Raise: đặt văn bản nằm phía trên Text Base Line ▪ Lower: đặt văn bản nằm phía dưới Text Base Line

- Click tab **Insert** > chọn **Symbol**
- Một danh sách các symbol thường xuyên được sử dụng sẽ xuất hiện.
- Nếu symbol mà bạn muốn chèn xuất hiện trong danh sách, click chọn symbol đó. Nếu không, click chọn **More Symbols**.
- Click chọn nhóm **Symbol** như: Symbol, Wingdings,... trong hộp **Font** và chọn symbol muốn chèn vào tài liệu
- Click **Insert** để chèn vào tài liệu và làm tiếp như vậy cho các ký hiệu muốn chèn khác.
- Sau khi đã chèn xong thì nhấn nút Cancel.

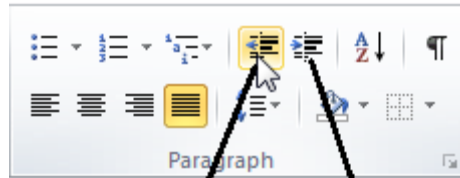


- * Nút lệnh **AutoCorrect** dùng để đưa các symbol này thành một mục từ của AutoCorrect.
- * Nút lệnh **Shortcut Key** dùng để gán một tổ hợp phím tắt cho một symbol thường sử dụng.
- * Chọn tab **Special Character** để chèn các ký tự đặc biệt như Copyright ©, Register ®,...



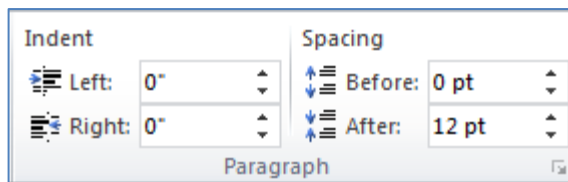
Mục chọn	Ý nghĩa
Left	Thụt lề trái của đoạn văn sang phải một khoảng theo kích thước xác định.
Right	Thụt lề phải của đoạn văn sang trái một khoảng theo kích thước xác định.
Special	Thụt lề đặc biệt
None	Đoạn văn không thụt lề
First Line	Thụt lề dòng đầu tiên của đoạn văn một khoảng cách xác định được quy định trong ô By.
Hanging	Thụt lề từ dòng thứ hai của đoạn văn một khoảng cách xác định được quy định trong ô By.

Cách 2: Click lên các nút lệnh **Decrease Indent** (giảm khoảng thụt lề) và **Increase Indent** (tăng khoảng thụt lề) trong nhóm lệnh **Paragraph** của tab **Home**.



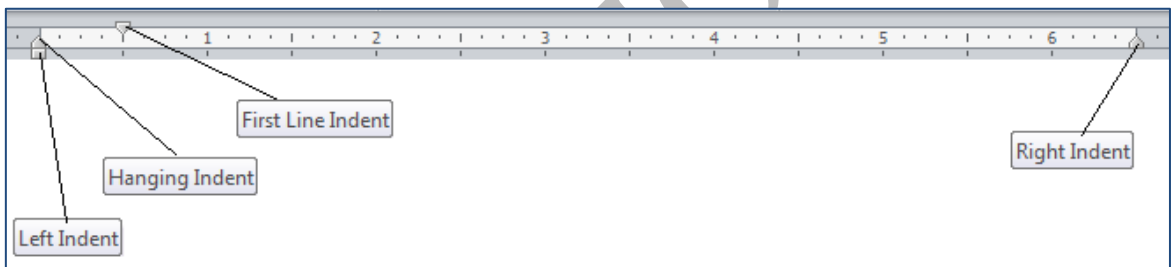
Decrease Indent (Ctrl + Shift + M) Increase Indent (Ctrl + M)





Bạn cũng có thể chỉ định khoảng cách thụt lề trái (**Left**) và thụt lề phải (**Right**) của đoạn văn khung **Spacing** của nhóm lệnh **Paragraph** trên tab **Page Layout**



Cách 3: Sử dụng thước (Ruler)

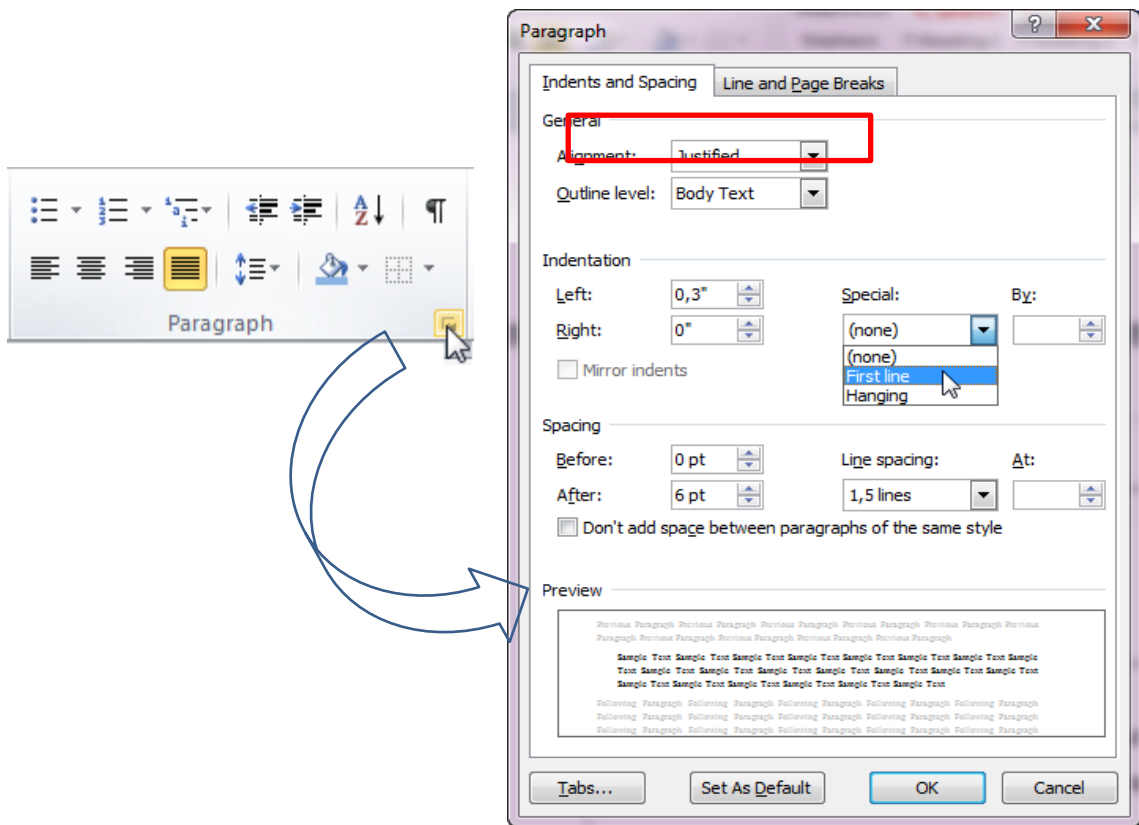
Trên thước ngang có các nút dùng để ấn định khoảng cách thụt dòng vào so với lề trang. Công dụng của các nút này như sau:



Ký hiệu	Ý nghĩa
 (Tab)	First Line Indent : định vị trí bắt đầu của dòng đầu đoạn văn
 (Ctrl + T)	Hanging Indent : định vị trí bắt đầu các dòng còn lại (từ dòng thứ hai) của đoạn văn
 (Ctrl + M)	Left Indent : định vị trí bắt đầu cho cả đoạn văn
	Right Indent : định vị trí kết thúc cho cả đoạn văn

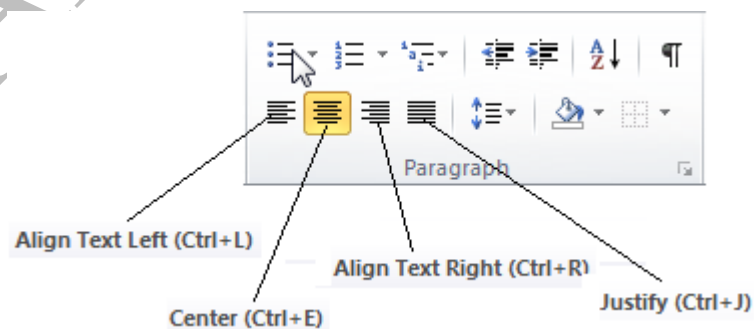
Ghi chú: Nút **Hanging Indent** và **Left Indent** luôn đi cặp với nhau nên khi kéo một nút thì cả hai nút cùng di chuyển theo. Nút **First Line Indent** di chuyển độc lập với hai nút **Hanging Indent** và **Left Indent**.

Minh họa các cách thụt lề:



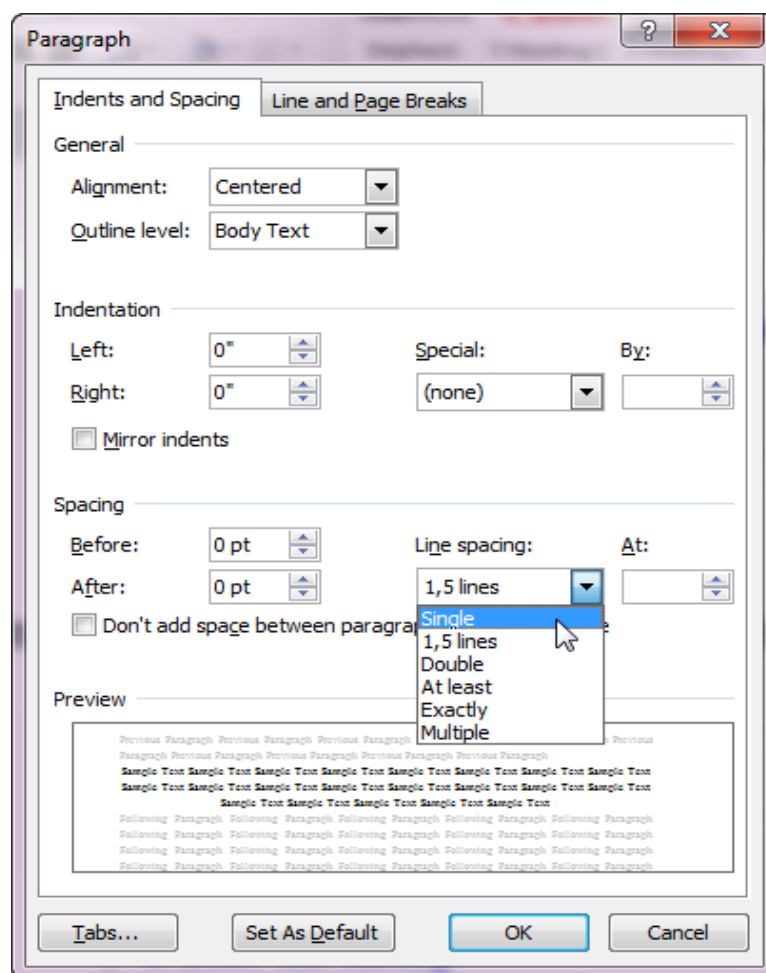
Mục chọn	Ý nghĩa
Left (Ctrl + L)	Canh đoạn văn thẳng theo lề trái
Center (Ctrl + E)	Canh đoạn văn ở giữa
Right (Ctrl + R)	Canh đoạn văn thẳng theo lề phải
Justify (Ctrl + J)	Canh đoạn văn thẳng theo cả lề trái và lề phải

Cách 2: Click lên các nút lệnh dùng để canh lề trong nhóm lệnh **Paragraph** của tab **Home**.



Minh họa các cách canh lề:

Align Left	Chủ yếu môn này liên quan đến kỹ năng, do đó sinh viên sẽ tự học dưới sự hướng dẫn là chính. Môn học này được tiến
-------------------	--



Word đo khoảng cách giữa các đoạn theo đơn vị point (pt).

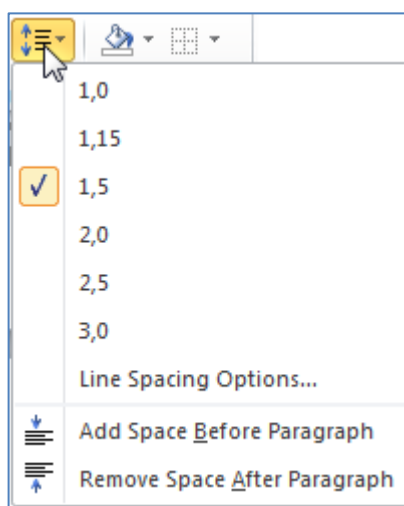
Khoảng cách mặc định giữa các dòng được tính theo số dòng, nhưng Word cho phép bạn thiết lập các khoảng cách này theo đơn vị Inch và cm.

Trong khung **Spacing**, bạn có thể tạo ra các khoảng cách thông qua các mục chọn trong bảng bên dưới

Mục chọn	Ý nghĩa
Before	Khoảng cách giữa đoạn văn được chọn so với đoạn văn trước
After	Khoảng cách giữa đoạn văn được chọn so với đoạn văn sau
Line Spacing	Khoảng cách giữa các dòng trong đoạn văn.
Single (Ctrl + 1)	Khoảng cách là 1 dòng đơn
1.5 lines (Ctrl + 5)	Khoảng cách là 1.5 dòng
Double (Ctrl + 2)	Khoảng cách là 2 dòng
At least	Khoảng cách tối thiểu giữa 2 dòng mà Word có thể chấp

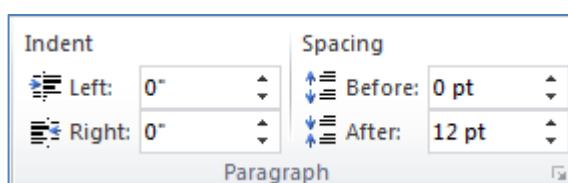
	nhận để làm vừa vặn văn font chữ. Khoảng cách này được hiển thị trong khung At
Exactly	Khoảng cách chính xác giữa 2 dòng. Khoảng cách này được hiển thị trong khung At
Multiple	Khoảng cách bằng với số dòng được thiết lập thông qua hộp At

Cách 2: Xác định khoảng cách dòng nhanh bằng lệnh **Line and Paragraph Spacing** trong nhóm lệnh **Paragraph** của tab **Home**

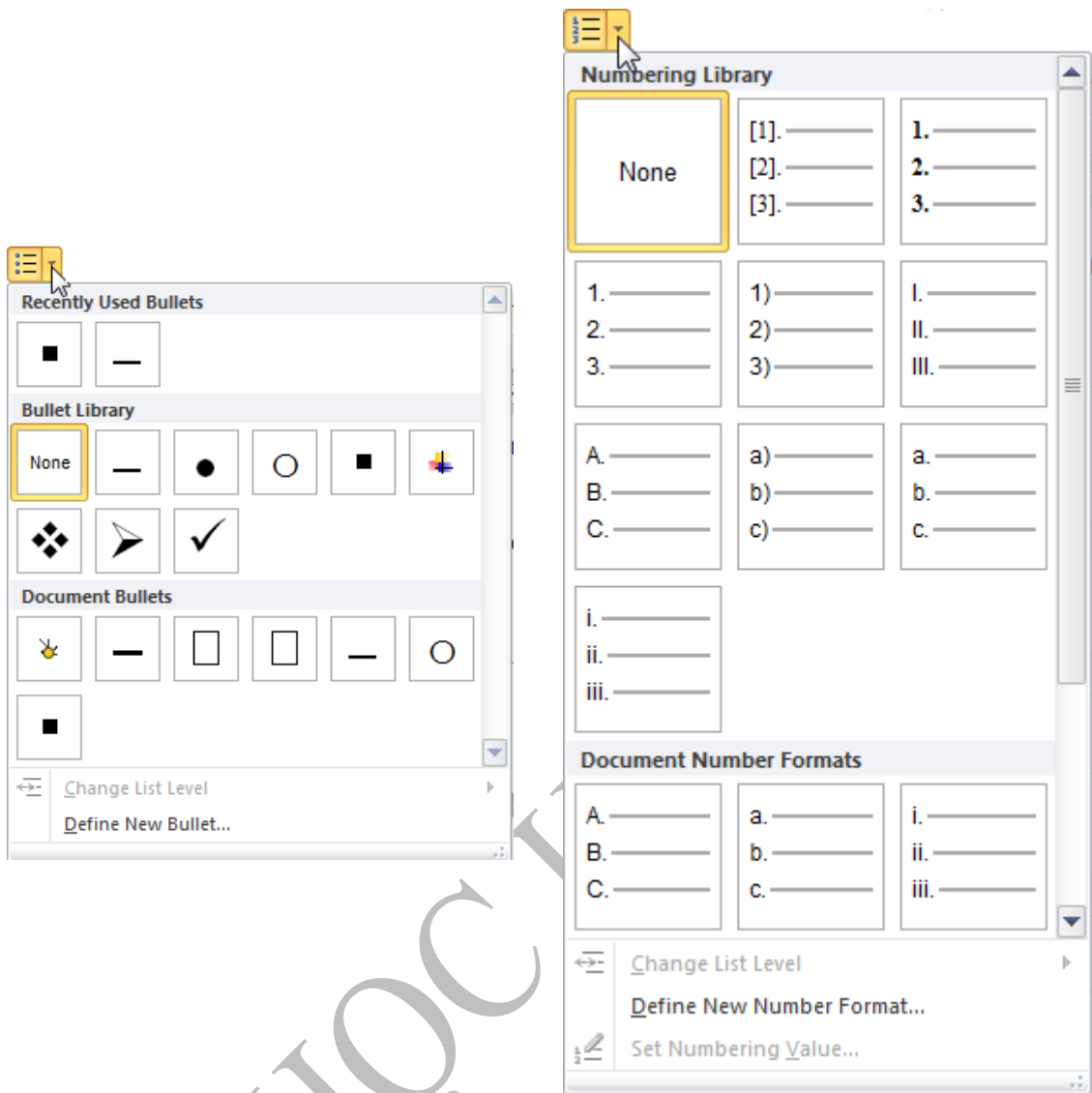


- Click chọn **Line Spacing Options** để mở hộp thoại **Paragraph** định dạng khoảng cách đoạn và dòng tùy ý.
- Click chọn **Add/Remove Space Before Paragraph** để thêm/loại bỏ khoảng cách trước đoạn văn.
- Click chọn **Add/Remove Space After Paragraph** để thêm/loại bỏ khoảng cách sau đoạn văn.

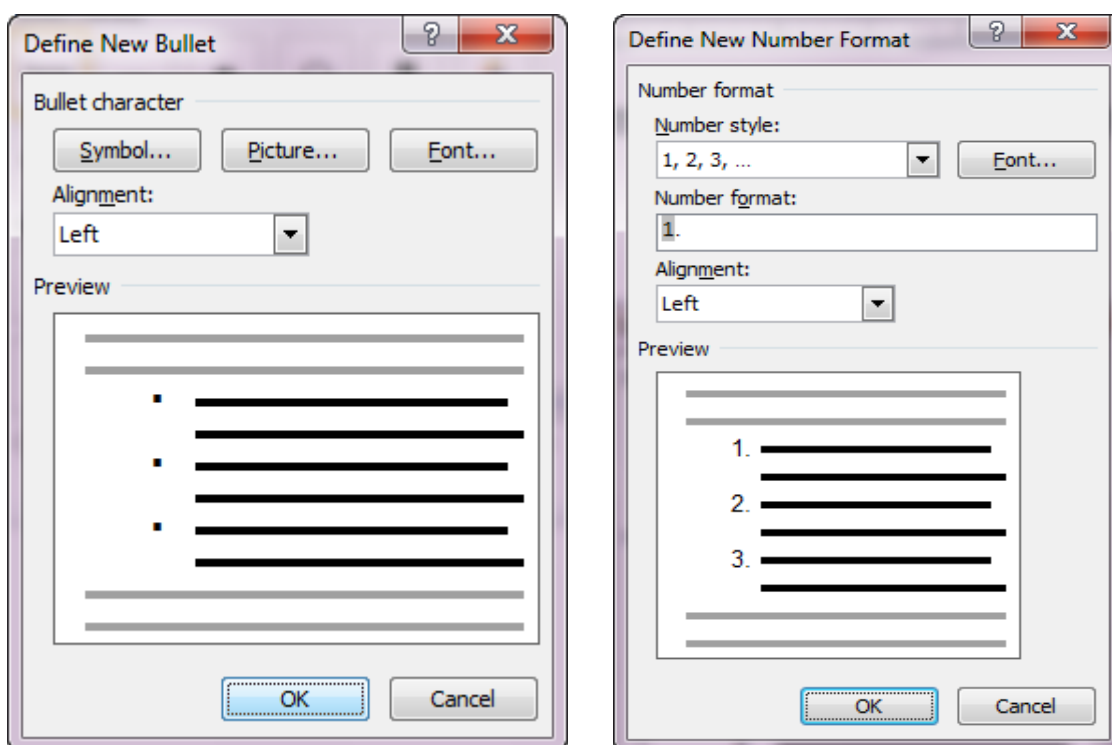
Cách 3: Xác định khoảng cách trước hay sau đoạn văn trực tiếp trong khung **Spacing** của nhóm lệnh **Paragraph** trên tab **Page Layout**



Minh họa cho cách xác định khoảng cách đoạn và khoảng cách dòng



Nếu bạn muốn thay đổi hoặc tạo ra một dạng **Bullet** hoặc **Numbering** mới, click vào **Define New Bullet** hoặc **Define New Number Format**, hộp thoại **Define New Bullet** hoặc **Define New Number Format** sẽ xuất hiện.

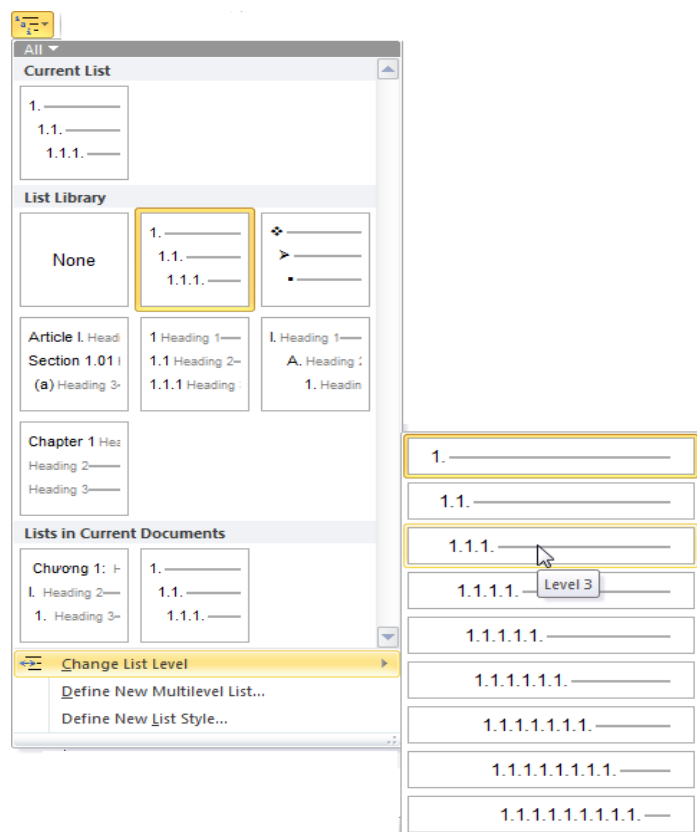


Trong hộp thoại **Define New Bullet**:

- Để thay đổi dấu được lấy làm bullets, bạn click vào nút **Symbol** bên dưới **Bullet character** để mở hộp thoại **Symbol**. Trong hộp thoại **Symbol**, click chọn ký tự mà bạn muốn dùng làm bullet. Click **OK**. Ký tự mới này sẽ xuất hiện trong khung **Preview** để xem trước. Click **OK** để thêm dấu đầu mỗi mục phần văn bản được chọn.
- Nếu bạn muốn dùng hình ảnh làm các đầu mục, click chọn **Picture** để mở hộp thoại **Picture Bullet**, chọn hình có sẵn trong danh sách hoặc click **Import** để thêm một hình mới. Click **OK** để chọn hình làm dấu đầu mục cho phần văn bản được chọn.
- Để thay đổi màu sắc, kích thước, font chữ cho các bullet, click chọn **Font** để mở hộp thoại **Font** và định dạng tùy ý. Nếu không định dạng thì các bullet ở chế độ mặc định là **Auto**.
- Chọn kiểu canh lề (Left, Centered, Right) cho các bullet trong mục **Alignment**.

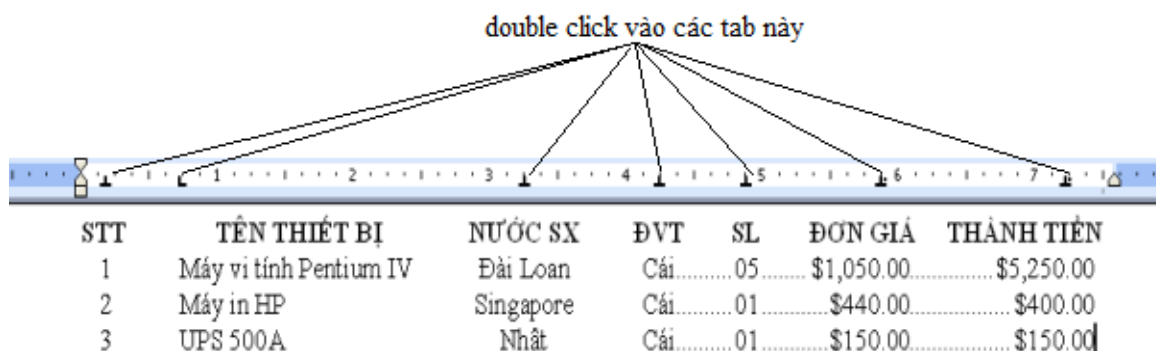
Trong hộp thoại **Define New Number Format**:

- Chọn kiểu của số (kiểu la mã: I, II, ...; kiểu số: 1, 2, ...; kiểu chữ: a, b, ...; ...) trong mục **Number style** và định dạng lại font chữ, màu sắc, kích thước nếu muốn bằng cách click chọn nút **Font**. Nếu không định dạng thì các số ở chế độ mặc định là **Auto**. Kiểu số cùng với định dạng sẽ xuất hiện trong ô **Number format**.
- Chọn kiểu canh lề (Left, Centered, Right) cho các số trong mục **Alignment**.



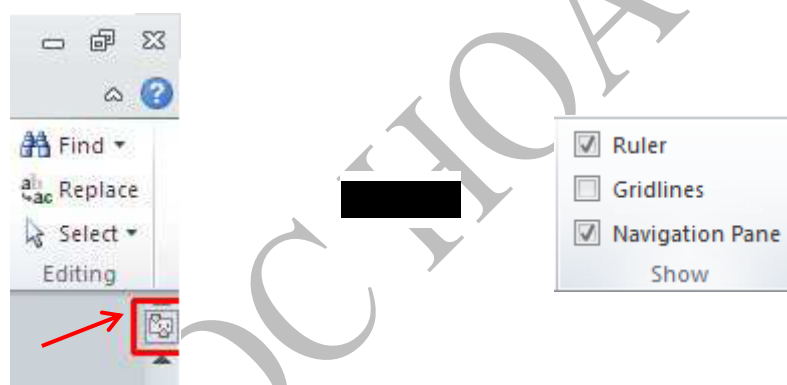
Để tạo mới danh sách phân cấp chọn **Define New Multilevel List**. Hộp thoại **Define new Multilevel List** xuất hiện. Muốn định dạng cho cấp nào, bạn phải chọn cấp trước trong khung **Level**, sau đó chọn kiểu số cho cấp đó trong khung **Number style for this level**. Tiếp tục như vậy, bạn sẽ định dạng được tất cả các cấp của **Multilevel list**

Để mở hộp thoại **Tabs**, bạn double click vào các ký hiệu của **Tab** trên thanh thước hoặc click nút **Tabs** trong hộp thoại **Paragraph**.








a. Cài đặt Tab

Để cho việc cài đặt Tab được thuận tiện, bạn nên cho hiển thị thước ngang ở trên đỉnh tài liệu bằng cách nhấn nút **View Ruler** ở trên đầu thanh cuộn dọc hoặc click chọn mục **Ruler** trong nhóm lệnh **Show** của tab **View**.



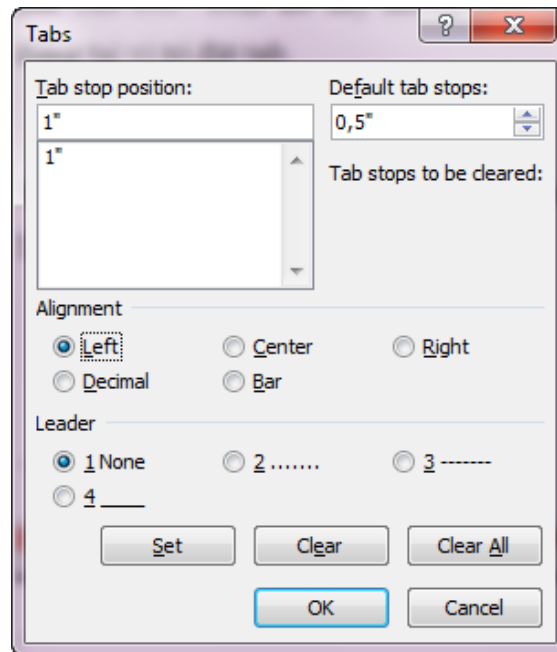
Bạn có thể nhanh chóng chọn tab bằng cách lần lượt nhấn vào ô tab selector – là ô giao nhau giữa thước dọc và thước ngang (ở góc bên trái trang tài liệu) cho khi chọn được kiểu tab mong muốn, sau đó click chuột vào thanh thước ngang nơi bạn muốn đặt tab.

Tùy từng trường hợp cụ thể mà bạn có thể sử dụng một trong các loại tab sau:

- **Tab trái**  : Đặt vị trí bắt đầu của đoạn text mà từ đó sẽ chạy sang phải khi bạn nhập liệu.
- **Tab giữa**  : Đặt vị trí chính giữa đoạn text. Đoạn text sẽ nằm giữa vị trí đặt tab khi bạn nhập liệu.
- **Tab phải**  : Nằm ở bên phải cuối đoạn text. Khi bạn nhập liệu, đoạn text sẽ di chuyển sang trái kể từ vị trí đặt tab.
- **Tab thập phân**  : Khi đặt tab này, những dấu chấm phân cách phần thập phân sẽ nằm trên cùng một vị trí.
- **Bar Tab**  : Loại tab này không định vị trí cho text. Nó sẽ chèn một thanh thẳng đứng tại vị trí đặt tab.

b. Tùy chỉnh Tab

Sau khi cài đặt các tab, bạn có thể tùy chỉnh các tab này bằng cách sử dụng hộp thoại **Tabs**.

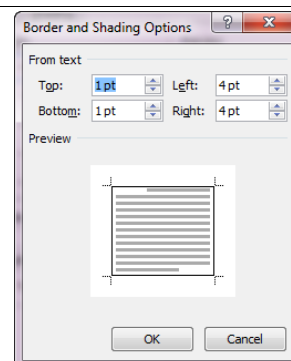


Mục chọn	Ý nghĩa
Default tab stops	Khoảng cách dừng của tab mặc định, thông thường là 0.5”
Tab stop position	Vị trí các điểm dừng tab.
Alignment	Chọn một loại tab là Left – Center, Right, Decial, Bar (Bar tab là tab chỉ có tác dụng tạo một đường kẻ dọc ở một vị trí đã định
Leader	chọn cách thể hiện tab
1 None	tab tạo ra khoảng trắng
2-3-4	tab tạo ra là các dấu chấm, dấu gạch hoặc gạch liền nét
Các nút lệnh	<ul style="list-style-type: none"> ▪ Set: đặt điểm dừng tab, vị trí đã xác định được đưa vào danh sách. ▪ Clear: xóa tab tại vị trí đã chọn. ▪ Clear All: xóa tất cả các tab đã đặt.

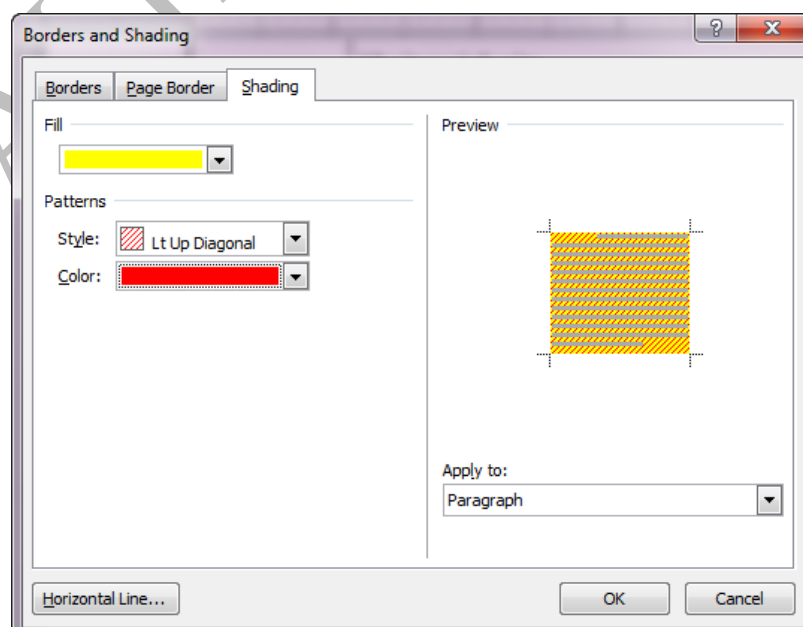
c. Xóa Tab

- Để xóa một điểm dừng tab, bạn click giữ mouse lên ký hiệu Tab trên thước rồi kéo ký hiệu Tab ra khỏi thước, sau đó thả mouse.
- Hoặc bạn có thể sử dụng nút lệnh **Clear** hoặc **Clear All** trong hộp thoại **Tabs**.

Mục chọn	Ý nghĩa
Setting <ul style="list-style-type: none"> ▪ None ▪ Box ▪ Shadow ▪ 3-D ▪ Custom 	Chọn các hình thức đóng khung có sẵn <ul style="list-style-type: none"> ▪ Không đóng khung ▪ Đóng khung đoạn ▪ Đóng khung đoạn có bóng ▪ Đóng khung tạo hiệu ứng 3-D ▪ Đóng khung theo ý muốn
Style	Kiểu đường kẻ khung
Color	Màu của đường kẻ
Width	Độ dày của đường kẻ
Preview	Thẻ hiện đoạn văn bản muốn kẻ khung. Bạn bấm vào các cạnh muốn kẻ khung hay các nút bên trái và bên dưới khung này.
Apply to <ul style="list-style-type: none"> ▪ Paragraph ▪ Text 	<ul style="list-style-type: none"> ▪ Đóng khung cho đoạn ▪ Đóng khung cho khối văn bản
Options	Mở hộp thoại Border and Shading Options : ấn định khoảng cách từ văn bản đến các đường kẻ trên, dưới, trái, phải. Bạn nhập hay bấm mũi tên lên xuống trong các hộp Top, Bottom, Left, Right để ấn định giá trị cho các khoảng cách này.




Để tô nền cho văn bản, sử dụng tab **Shading**.



Một số tùy chọn trong tab này như sau:

Mục chọn	Ý nghĩa
Fill	Chọn màu nền cho đoạn văn
Pattern <ul style="list-style-type: none"> ▪ Style ▪ Color 	<ul style="list-style-type: none"> ▪ Chọn một kiểu hoa văn cho nền. ▪ Chọn màu cho nền.
Apply to <ul style="list-style-type: none"> ▪ Paragraph ▪ Text 	<ul style="list-style-type: none"> ▪ Tô nền cho đoạn ▪ Tô nền cho khối văn bản

Chú ý: Bạn có thể sử dụng nút lệnh **Text Highlight Color**  trong nhóm lệnh **Font** của tab **Home** để tô màu nền cho khối văn bản.

Ví dụ về đóng khung và tô nền cho văn bản như sau:

LUSCIOUS JAM CO. PTY. LTD
239 Campbell Street, Chippendale, NSW. 2008, Tel: 213489

Công nghệ RamBus tăng khối lượng xuất liệu
RamBus hứa hẹn những bo mạch nhanh hơn, rẻ hơn.

Các chi tiết xin liên hệ với:

Trường Cao Đẳng Bán Công HOA SEN
Khoa Đào tạo Ngắn hạn – Phòng C01
Số 8 Nguyễn Văn Tráng, Q. 1, TP. HCM
☎: 8301 877 – 8324 602 – 8323 224
Fax: 8301 878 - ✉: lotus1@hcm.vnn.vn

HOA SEN

HOA SEN

HOA SEN

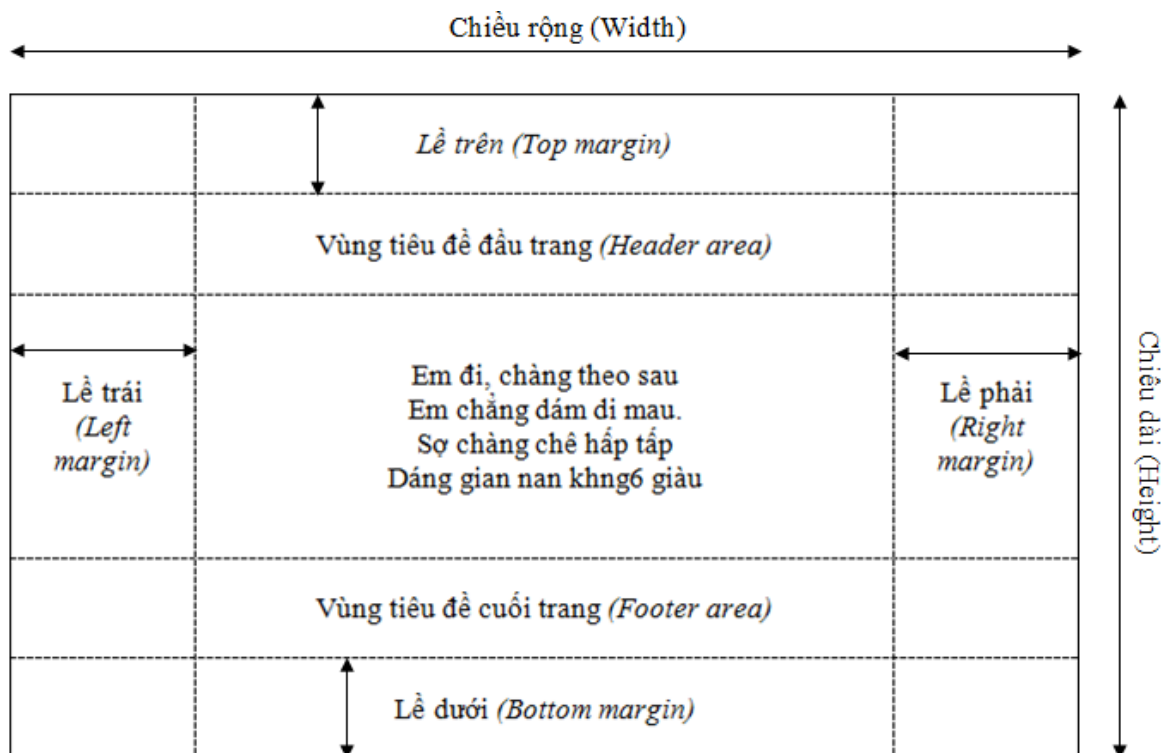
HOA SEN

HOA SEN

HOA SEN

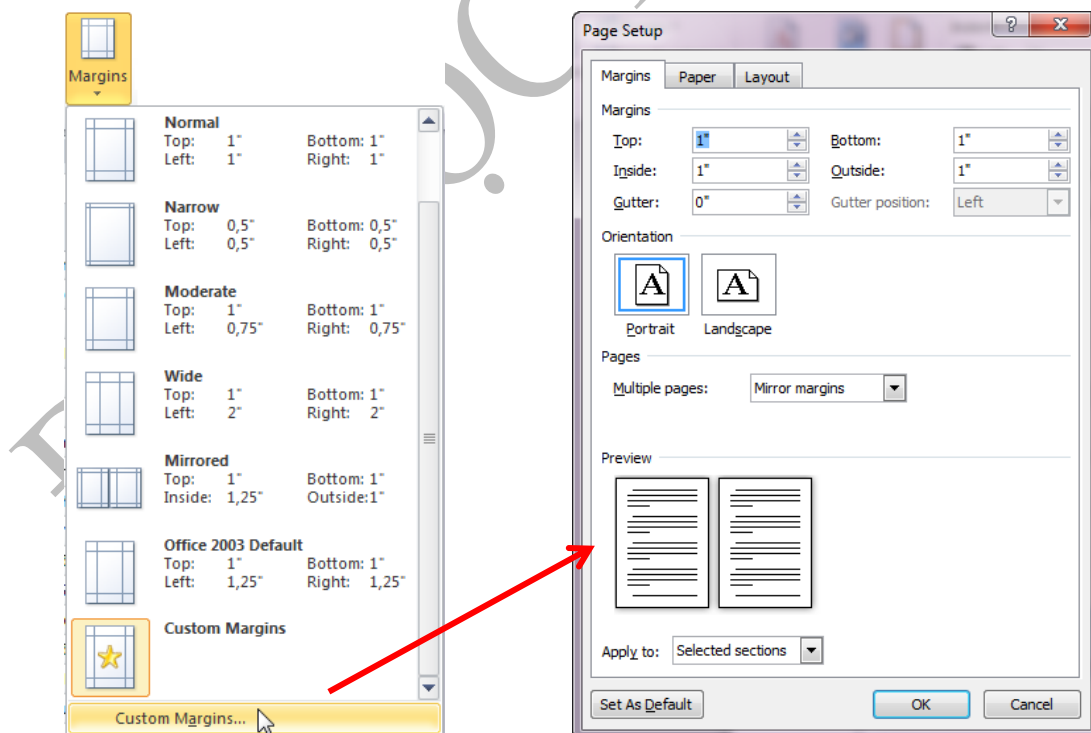
HOA SEN

HOA SEN

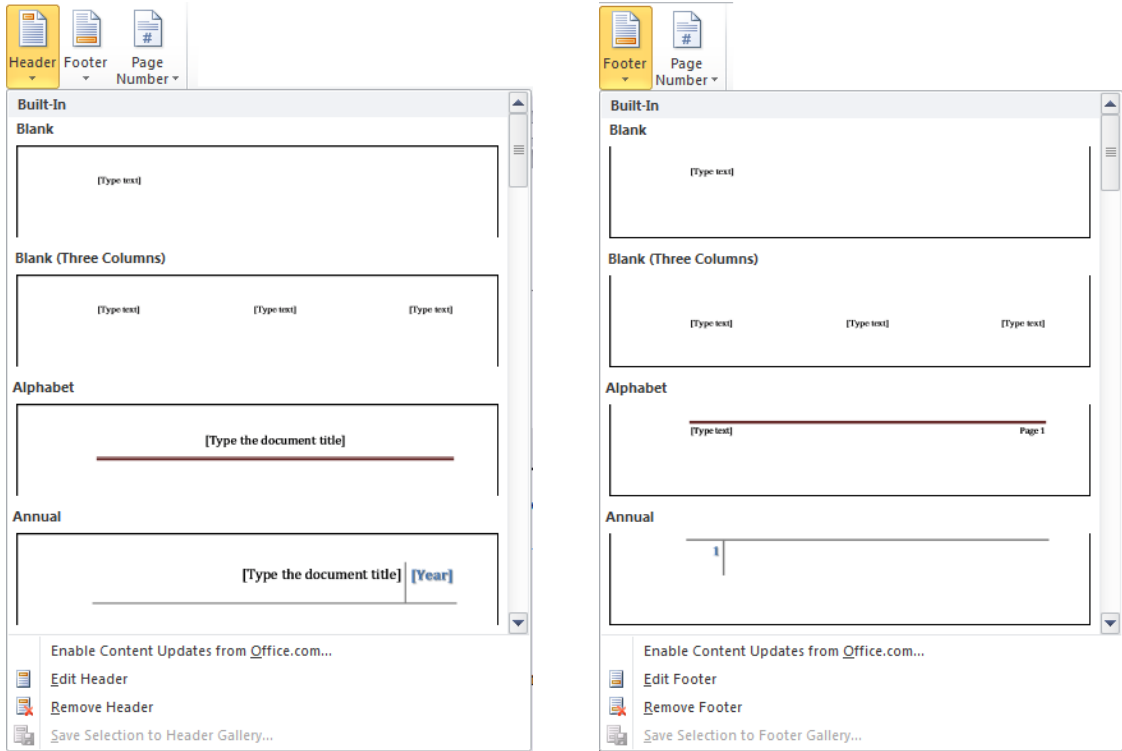


Để ấn định lề trang giấy:

- Trên tab **Page Layout** > nhóm lệnh **Page Setup** > chọn nút lệnh **Margins**. Một danh sách các mẫu canh lề có sẵn...Bạn chọn một mẫu hoặc click chọn **Custom Margins** để canh lề theo ý mình. Hộp thoại Page Setup xuất hiện

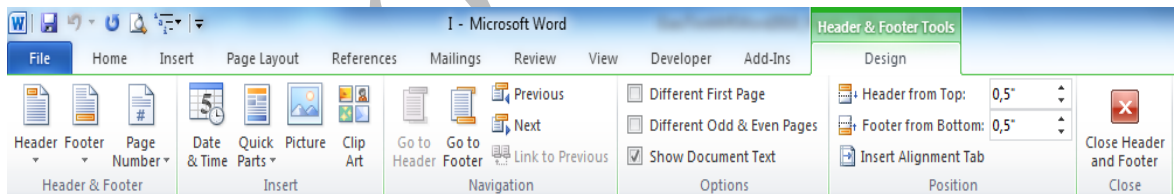


- Chọn tab **Margins**
 - Trong các ô **Top, Bottom, Left, Right, Gutter** bạn hãy nhập các số đo cho phù hợp





Một danh sách hiển thị cho phép bạn chọn các kiểu của header các kiểu của footer. Bạn chọn một kiểu tùy ý trong danh sách.

Bạn có thể click vào lệnh **Edit Header** hoặc **Edit Footer** trong danh sách hiển thị bên trên (cũng có thể double click vào cùng đầu trang Header và Footer). Thanh công cụ **Header & Footer Tools** hiển thị với tab **Design**, bao gồm những nhóm lệnh riêng cho việc soạn thảo Header and Footer.



Ngoài các lệnh trên tab này, bạn có thể phối hợp sử dụng các lệnh trên các tab khác để định dạng ký tự, định dạng đoạn văn bản, chèn hình ảnh... trong quá trình tạo Header và Footer

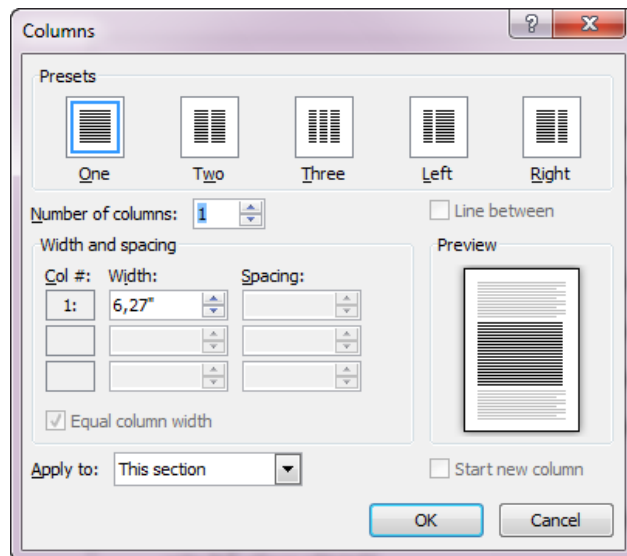
NÚT LỆNH	CHỨC NĂNG
 Header	Thay đổi tiêu đề header của tài liệu. Nội dung trong header sẽ xuất hiện ở đầu của mỗi trang in.
 Footer	Thay đổi tiêu đề footer của tài liệu. Nội dung trong footer sẽ xuất hiện ở đầu của mỗi trang in.

Vi dụ: muốn in các trang 2, từ trang 4 đến trang 12 và trang 15, bạn nhập vào hộp **Pages** như sau: 2, 4-12, 15.

- Hộp chọn **Print One Sided:**
 - Print One Sided: in một mặt của trang giấy
 - Manually Print on Both Sides: in hai mặt của trang giấy
- Hộp chọn **Collated:** chọn để in hết từng bản in một, rồi mới quay lại in bản kế tiếp hoặc in một trang của tài liệu thành nhiều bản, sau đó tiếp tục như vậy cho các trang kế.
- Hộp chọn **Portrait Orientation:** chọn hướng in của giấy là hướng đứng (*Portrait Orientation*) hay hướng ngang (*Landscape Orientation*).
- Hộp chọn **Leters và Normal margins:** chọn khổ giấy và lề trang cho giấy in
- Hộp chọn **1 Page Per Sheet:** chọn số lượng trang in trên một tờ giấy.

- **Right**: chia văn bản thành 03 cột không bằng nhau với bề rộng cột phải bằng nửa bề rộng cột trái.

Nếu bạn muốn chia văn bản thành nhiều cột khác với các loại trên hoặc tùy chỉnh các sự lựa chọn, bạn click chọn **More Coloumn**. Hộp thoại **Columns** xuất hiện.




Ý nghĩa của các mục lựa chọn trong hộp thoại này như sau:

Mục chọn	Ý nghĩa
Presets <ul style="list-style-type: none"> One Two Three Left Right 	Chọn cách chia cột theo định dạng của Word <ul style="list-style-type: none"> Hoàn trả lại văn bản về tình trạng không chia cột. Chia văn bản thành 02 cột bằng nhau. Chia văn bản thành 03 cột bằng nhau. Chia văn bản thành 02 cột không bằng nhau với bề rộng cột trái bằng nửa bề rộng cột phải. Chia văn bản thành 03 cột không bằng nhau với bề rộng cột phải bằng nửa bề rộng cột trái.
Number of columns	Ấn định số cột bằng cách nhập vào số cột hoặc bấm nút mũi tên để tăng/giảm số cột
Line between	Chọn để tạo đường kẻ giữa các cột
Width and Spacing <ul style="list-style-type: none"> Width Spacing 	Ấn định bề rộng các cột và khoảng cách giữa các cột <ul style="list-style-type: none"> Điều chỉnh bề rộng cột Điều chỉnh khoảng cách giữa các cột
Equal columns width	Chọn để các cột có cùng bề rộng
Apply to	Chọn cách áp dụng chia cột vào văn bản, tùy theo văn bản được chọn hay vị trí con trỏ đang đứng mà bạn có các lựa chọn sau: <ul style="list-style-type: none"> Whole document: áp dụng chia cột cho toàn bộ tài liệu.


	<ul style="list-style-type: none"> ▪ This point forward: áp dụng chia cột từ vị trí chia cột trở đi. <i>Lưu ý:</i> trước đó bạn phải đặt một dấu phân chia section trước cho con ▪ Selected text: chỉ áp dụng chia cột cho phần văn bản được chọn ▪ Selected sections: chỉ áp dụng chia cột cho các phân đoạn được chọn. ▪ This section: chỉ áp dụng chi cột cho phân đoạn hiện hành (đăng chứa con trở).
Preview	Hiện thị cách chia cột đã chọn.

Ngoài cách thay đổi khoảng cách giữa các cột và bề rộng cột bằng hộp thoại Columns, bạn có thể dùng các nút Indent marker trên thước để thay đổi chúng. Các bước thực hiện như sau:

Để thay đổi khoảng cách giữa các cột:

- Đưa con trỏ vào phần văn bản nhiều cột muốn thay đổi khoảng cách cột.
- Đưa con trỏ mouse vào biên của Marker trên thước , mouse trở thành mũi tên hai chiều.
- Bấm và kéo để thay đổi khoảng cách giữa các cột.

Để thay đổi bề rộng cột:

- Đưa con trỏ vào phần văn bản muốn thay đổi bề rộng.
- Đưa con trỏ mouse vào giữa Marker trên thước , mouse trở thành mũi tên hai chiều.
- Bấm vào kéo để thay đổi bề rộng cột.

Chú ý: Nếu giữ phím Alt khi kéo mouse sẽ hiện số đo trên thước.

Ví dụ về văn bản dạng cột

Two Column

There is a complexity in the structures of Chinese Christianity which is little understood in the West. To start with, Catholicism and Protestantism are designated by the state as two separate religions.

The Haidian Christian Church in Beijing was completely re-built to cope with rising numbers

Throughout the 20th Century, Christianity was associated with Western imperialism. After the Communist victory in 1948, the missionaries were expelled, but Christianity was permitted

in state-sanctioned churches, so long as they gave their primary allegiance to the Communist Party.

Mao, on the other hand, described religion as "poison", and the Cultural Revolution of the 1960s and 70s attempted to eradicate it. Driven underground, Christianity not only survived, but with its own Chinese martyrs, it grew in strength.

Since the 1980s, when religious belief was again permitted, the official Churches have gradually created more space for themselves.

Left Column

There is a complexity in the structures of Chinese Christianity which is little understood in the West. To start with, Catholicism and Protestantism are designated by the state as two separate religions.

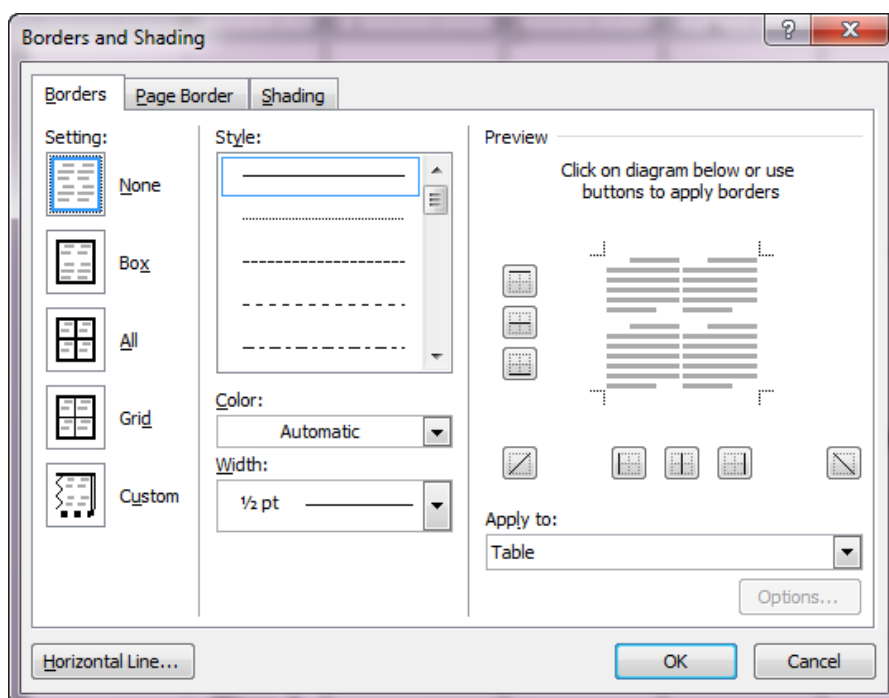
The Haidian Christian Church in Beijing was completely re-built to cope with rising numbers

Throughout the 20th Century, Christianity was associated with Western imperialism. After the Communist victory in 1948, the missionaries were expelled, but Christianity was permitted in state-sanctioned churches, so long as they gave their primary allegiance to the Communist Party.

Mao, on the other hand, described religion as "poison", and the Cultural Revolution of the 1960s and 70s attempted to eradicate it. Driven underground, Christianity not only survived, but with its own Chinese martyrs, it grew in strength.

Since the 1980s, when religious belief was again permitted, the official Churches have gradually created more space for themselves.

Right Column



Để đóng khung cho bảng, chọn tab **Border**. Một số tùy chọn trong tab này như sau:

Mục chọn	Ý nghĩa
Setting <ul style="list-style-type: none"> ▪ None ▪ Box ▪ All ▪ Grid ▪ Custom 	Chọn các hình thức đóng khung có sẵn <ul style="list-style-type: none"> ▪ Không đóng khung ▪ Đóng khung 4 hướng ngoài của bảng ▪ Đóng khung cả bảng theo dòng và theo cột ▪ Đóng khung cả bảng theo dòng và theo cột ▪ Đóng khung theo ý muốn
Style	Kiểu đường kẻ khung
Color	Màu của đường kẻ
Width	Độ dày của đường kẻ
Preview	Thể hiện bảng muốn kẻ khung. Bạn bấm vào các cạnh muốn kẻ khung hay các nút bên trái và bên dưới khung này.
Apply to <ul style="list-style-type: none"> ▪ Paragraph ▪ Cell ▪ Table 	<ul style="list-style-type: none"> ▪ Đóng khung cho đoạn văn ▪ Đóng khung cho ô ▪ Đóng khung cho bảng

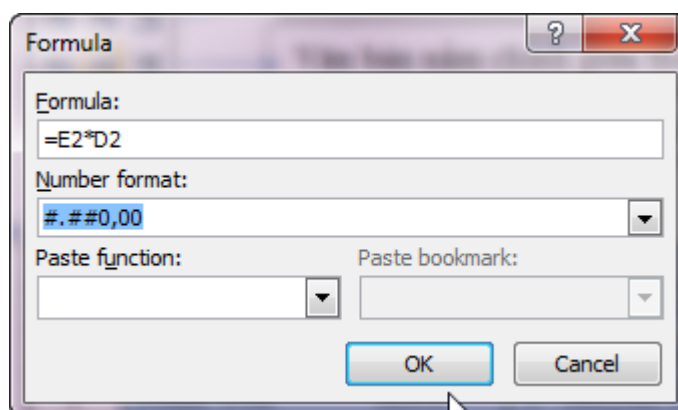
b. Tô nền

Muốn nhấn mạnh một số ô hoặc toàn bộ bảng để lưu ý người đọc, bạn có thể tô nền một vài hoặc tất cả các ô của bảng. Bạn tiến hành thực hiện như sau:

Cách 1: dùng hộp thoại **Border and Shading**

- Chọn các ô hoặc bảng muốn tô nền

- Trên tab **Layout**, trong nhóm lệnh **Data**, chọn nút lệnh **Formula**. Hộp thoại **Formula** xuất hiện.



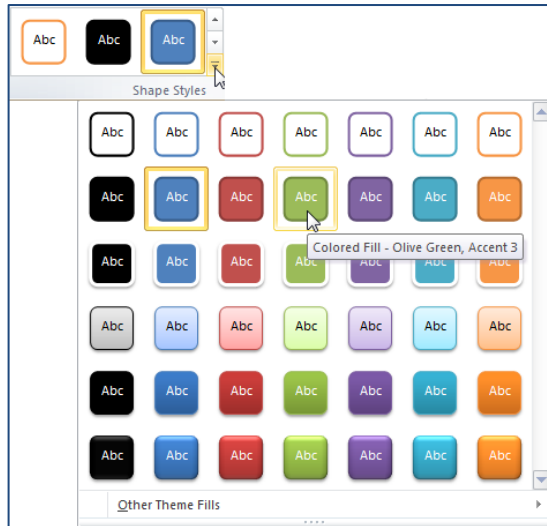
- Trong hộp **Formula**, nhập công thức muốn tính toán sau dấu =
- Trong hộp **Number format**: chọn dạng thể hiện kết quả của công thức.
- * Nếu muốn sử dụng hàm trong công thức, bạn có thể nhập trực tiếp hay từ hộp **Paste function**
- Click nút OK

Cách ghi địa chỉ ô

- **Địa chỉ ô**: các ô được ghi địa chỉ như sau: A1, B2, C3,...trong đó, ký tự đứng trước chỉ cột, đứng sau chỉ dòng.

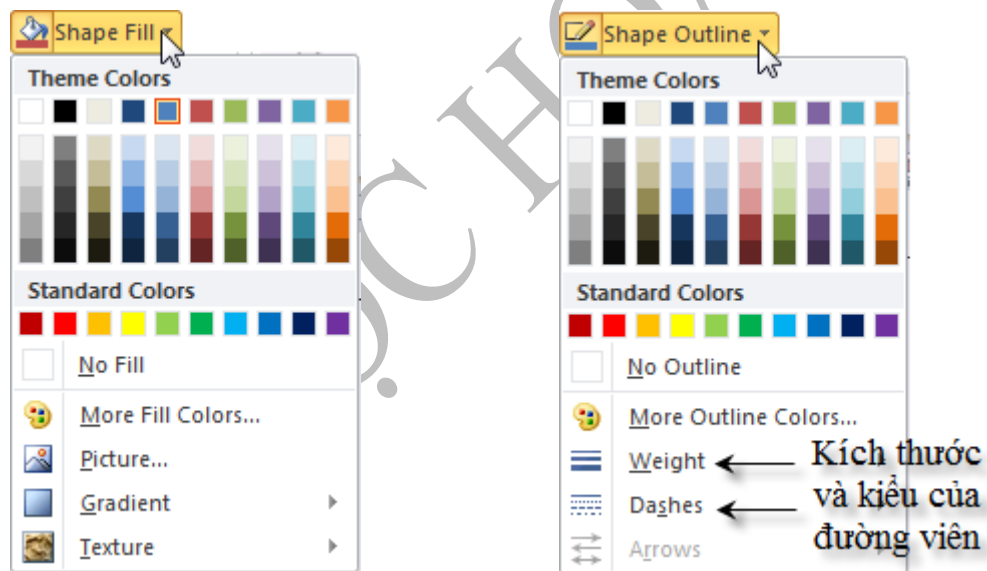
	A	B	C
1	A1	B1	C1
2	A2	B2	C2
3	A3	B3	C3

- **Vùng**: gồm nhiều ô liên tiếp nhau được ghi như sau: ghi địa chỉ ô đầu tiên, dấu hai chấm (:), địa chỉ ô cuối cùng (Ví dụ: A1:C3)
- Ngoài cách ghi địa chỉ vùng như trên, Word còn cho phép sử dụng cách ghi khác với ý nghĩa như sau:
 - **ABOVE**: gồm các ô kể từ ô nằm ngay bên trên cùng cột với ô đang nhập công thức và lên trên cho đến khi gặp ô chứa giá trị không phải là kiểu số.
 - **BELOW**: gồm các ô kể từ ô nằm ngay bên dưới cùng cột với ô đang nhập công thức và xuống dưới cho đến khi gặp ô chứa giá trị không phải là kiểu số.
 - **LEFT**: gồm các ô kể từ ô nằm ngay bên trái cùng dòng với ô đang nhập công thức và qua trái cho đến khi gặp ô chứa giá trị không phải là kiểu số.
 - **RIGHT**: gồm các ô kể từ ô nằm ngay bên phải cùng dòng với ô đang nhập công thức và qua phải cho đến khi gặp ô chứa giá trị không phải là kiểu số.



b. Hiệu chỉnh đường viền và tô nền cho hình

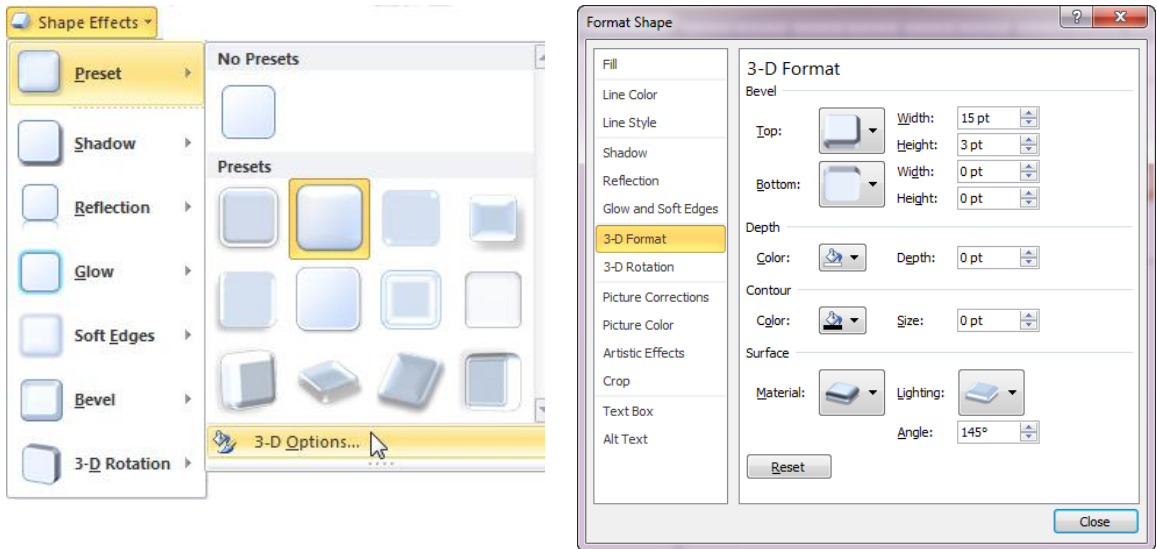
Sử dụng nút lệnh **Shape Fill** và **Shape Outline** trong nhóm lệnh **Shape Styles**, bạn có thể thay đổi đường viền và nền cho hình như: thay đổi màu sắc (Color), kiểu (Dash) và kích thước (Weight) của đường viền hoặc tô nền cho hình.



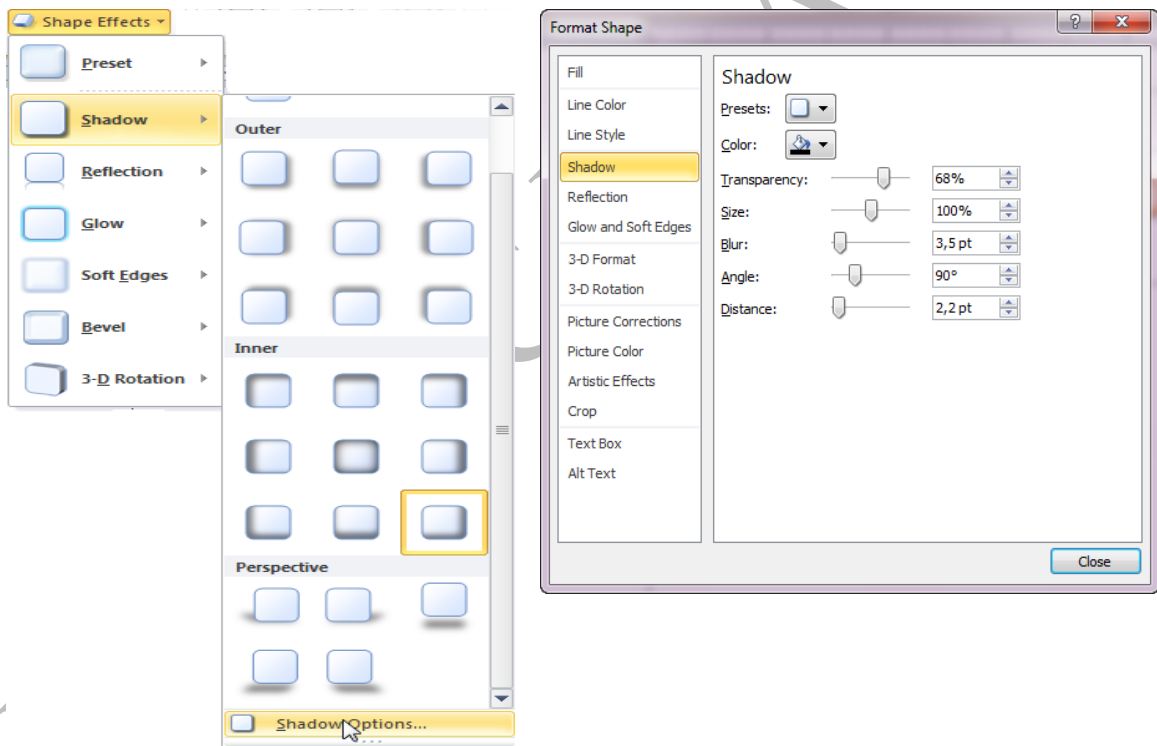
c. Tạo hiệu ứng cho hình (Shape Effects)

Để tạo hiệu ứng cho hình như: bóng đổ (Shadow), phản chiếu (Reflection), phát sáng (Glow)..., trong nhóm lệnh **Shape Styles**, click chọn **Shape Effects**, một danh sách xổ xuống cung cấp cho bạn các tùy chọn sau:

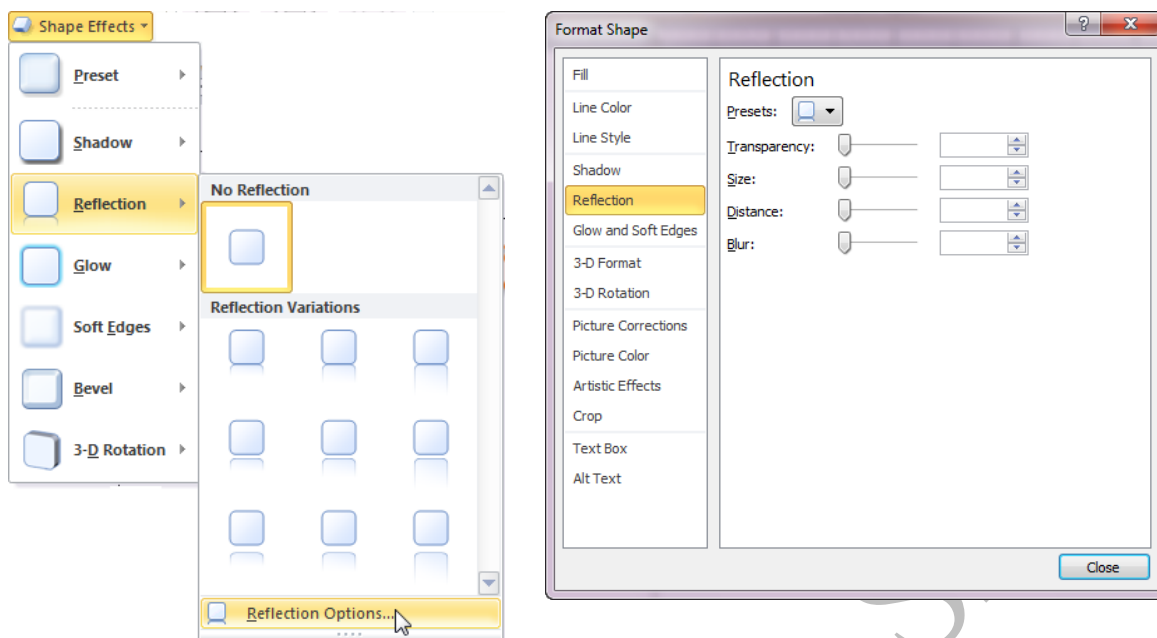
Preset: Khi click chọn **3-D Options**, hộp thoại **Format Shape** cho phép bạn thiết lập tùy ý.



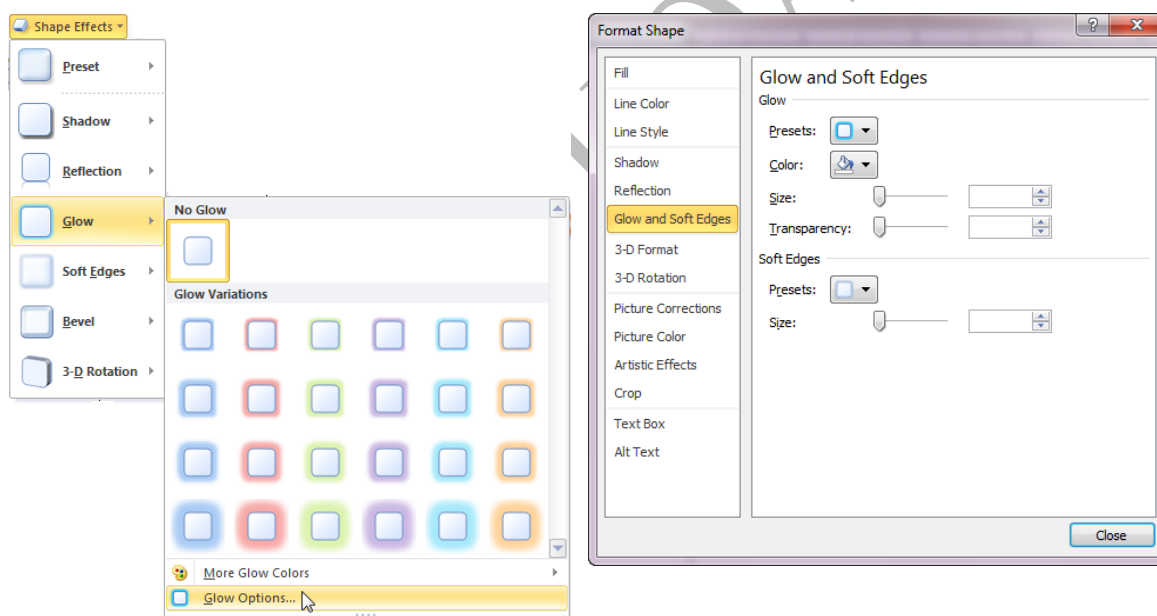
Shadow: tạo bóng đổ bên trong (Inter), bên ngoài (Outer), phối cảnh (Perspective) cho đối tượng. Click chọn **Shadow Options** cho phép bạn thiết lập bóng đổ tùy ý.



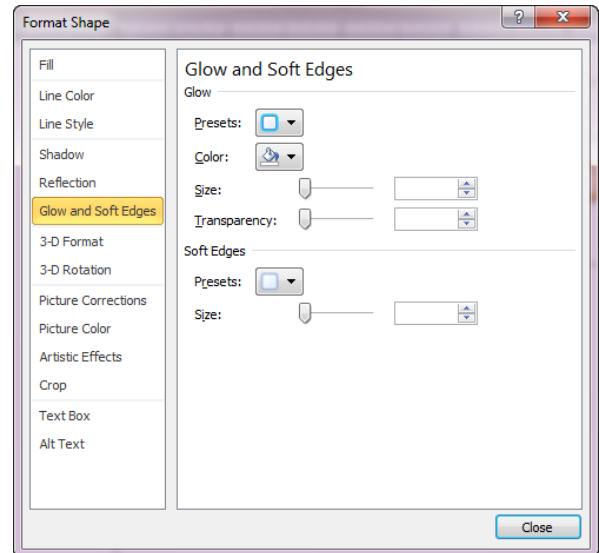
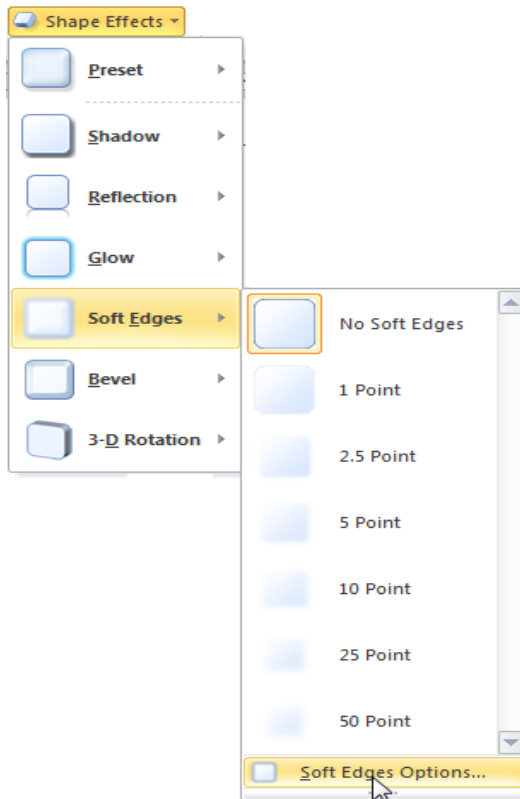
Reflection: độ phản chiếu. Click chọn **Reflection Options** để tùy chỉnh.



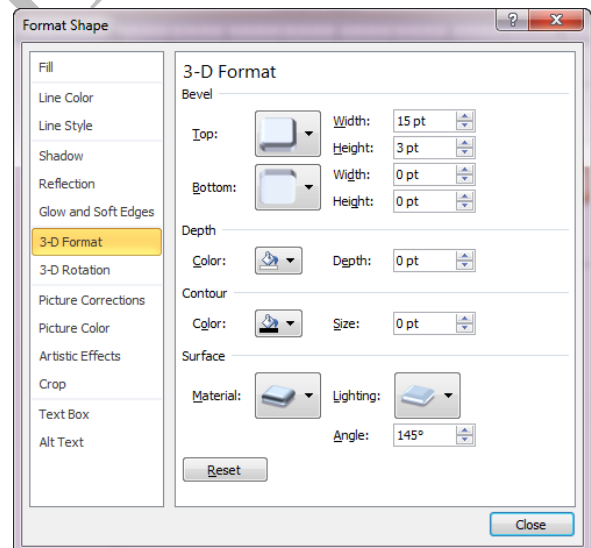
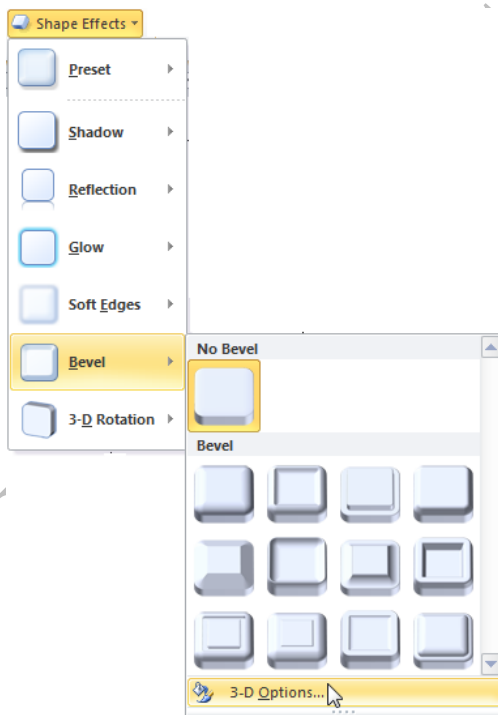
Glow: độ phát sáng. Click chọn **Glow Options** để tùy chỉnh.



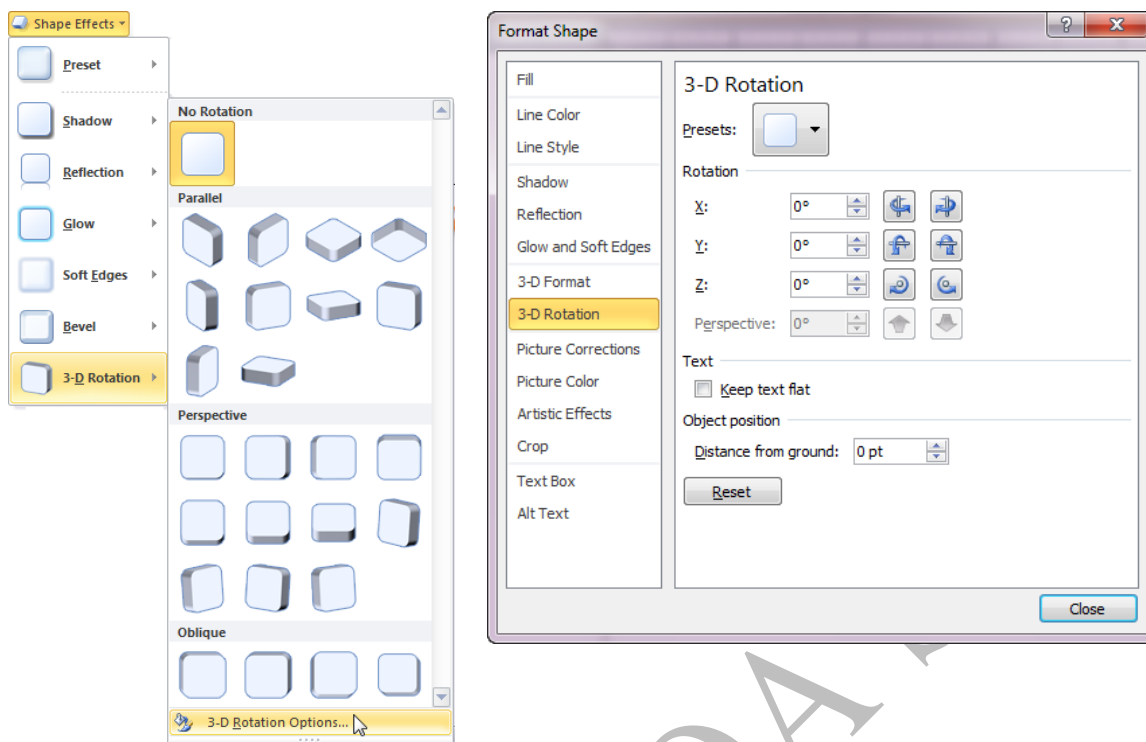
Soft Edges: Click chọn **Soft Edges Option** để tùy chỉnh.



Bevel: tạo gờ nổi bề mặt và biên. Click chọn **3-D Options** để tùy chỉnh.



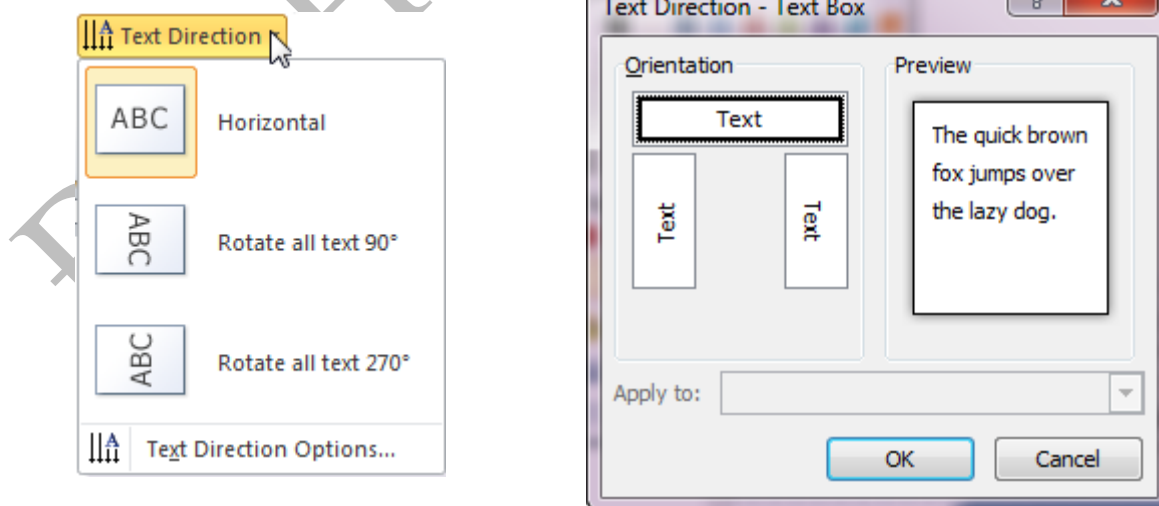
3-D Rotation: tạo góc xoay trong không gian 3 chiều. Click chọn **3-D Rotations Options** để tùy chỉnh.



d. Thay đổi hướng trình bày văn bản bên trong hình

Mặc định khi bạn nhập văn bản vào hình, văn bản sẽ được trình bày theo hướng nằm ngang. Tuy nhiên, bạn có thể thay đổi hướng trình bày văn bản bên trong hình bằng cách:

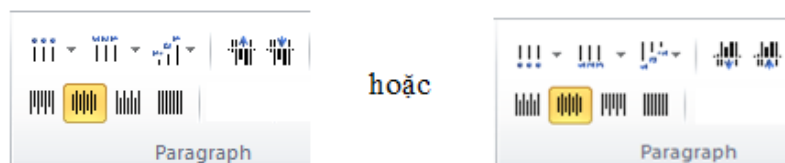
- Chọn hình muốn thay đổi hướng trình bày của văn bản bên trong.
- Trong nhóm lệnh **Text**, chọn **Text Direction** và chọn một hướng trình bày tùy ý



Bạn có thể chọn **Text Direction Options** để mở hộp thoại **Text Direction – Text Box**.

- Trong khung **Orientation**, chọn một hướng trình bày cho văn bản bằng cách bấm chọn một trong 3 hộp chứa hướng văn bản. Kết quả chọn hướng sẽ hiện ra trong khung **Preview**.
- Bấm nút OK

Chú ý: Khi văn bản trong hình ở hướng đứng, các nút canh lề văn bản và các nút Bullets and Numbering, nút Increase Indent, Decrease Indent (trong nhóm lệnh **Paragraph** của tab **Home**) sẽ thay đổi như sau:



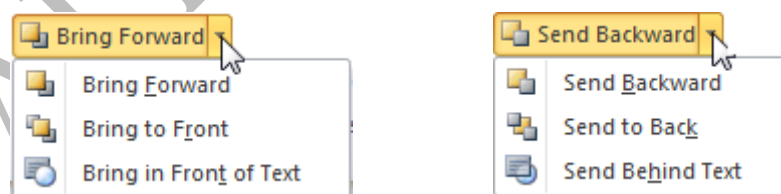
e. Thứ tự của các đối tượng (Order)

Khi làm việc với các đối tượng, bạn cần nắm vững các nguyên tắc sau:

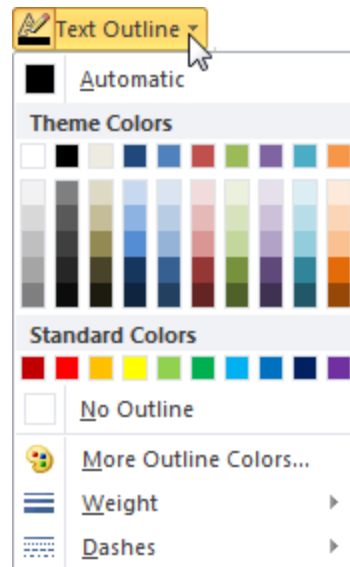
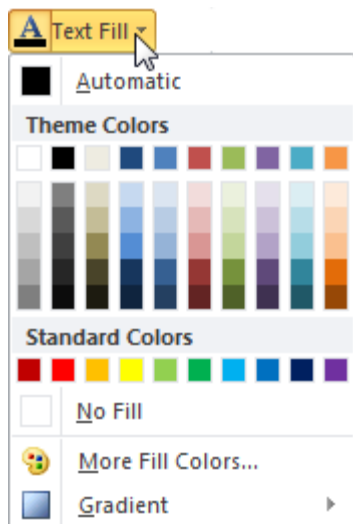
- Các lớp đối tượng có thể nằm xếp chồng lên nhau, đối tượng nằm ở lớp dưới sẽ bị đối tượng ở lớp trên che khuất.
- Đối tượng được tạo (vẽ) sau mặc nhiên sẽ được tạo trước.
- Các đối tượng có thể nằm trên văn bản, cũng có thể nằm dưới văn bản.

Để thay đổi thứ tự (Order) của các đối tượng, bạn thực hiện như sau:

- Chọn thứ tự muốn thay đổi thứ tự
- Trên tab **Format** > nhóm lệnh **Arrange** > chọn nút lệnh **Bring Forward** nếu muốn mang đối tượng lên trên hoặc **Send Backward** nếu muốn đưa đối tượng xuống dưới các đối tượng khác. Một số sự lựa chọn khác khi chọn 2 nút lệnh này như sau:



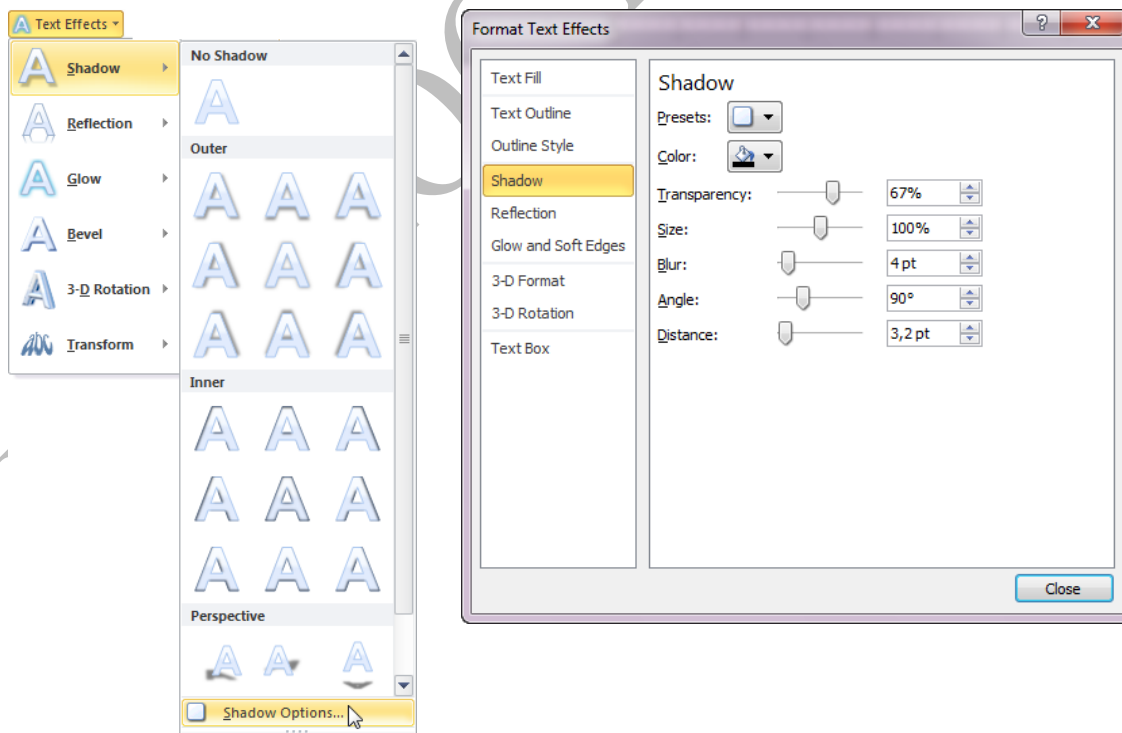
- **Bring Forward:** đem đối tượng lên trên một lớp
- **Bring to Front:** đem đối tượng lên lớp trên cùng (trên tất cả các đối tượng khác).
- **Bring in Front of Text:** đem đối tượng lên trên văn bản
- **Send Backward:** đem đối tượng xuống dưới một lớp
- **Send to Back:** đem đối tượng xuống lớp dưới cùng (dưới tất cả các đối tượng khác).
- **Bring Behind Text:** đem đối tượng xuống dưới văn bản



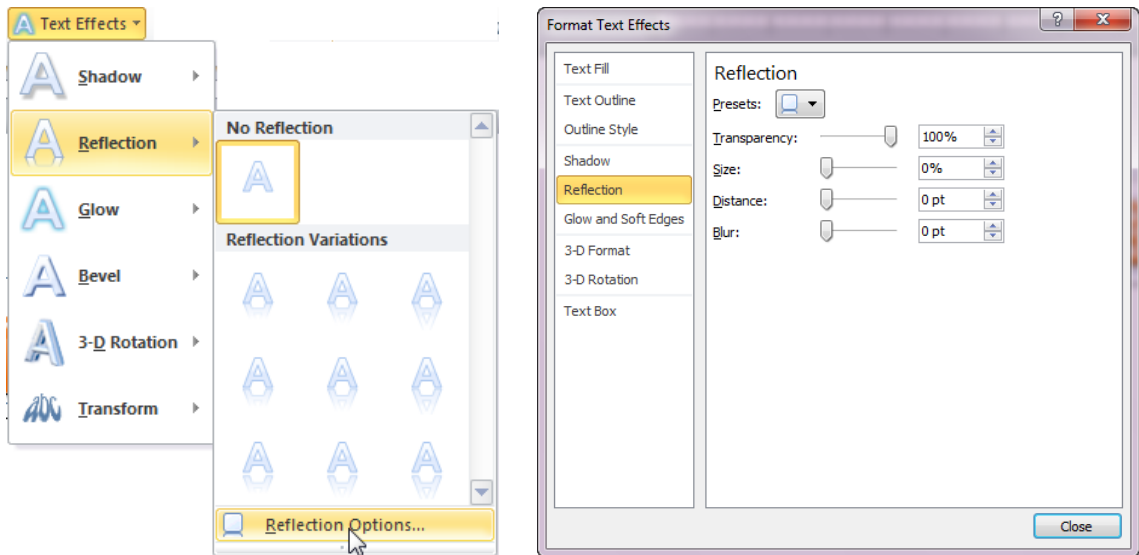
c. Tạo hiệu ứng cho WordArt (Text Effects)

Để tạo hiệu ứng cho WordArt như: bóng đổ (Shadow), phản chiếu (Reflection), phát sáng (Glow)..., trong nhóm lệnh **WordArt Styles**, click chọn **Text Effects**, một danh sách xổ xuống cung cấp cho bạn các tùy chọn sau:

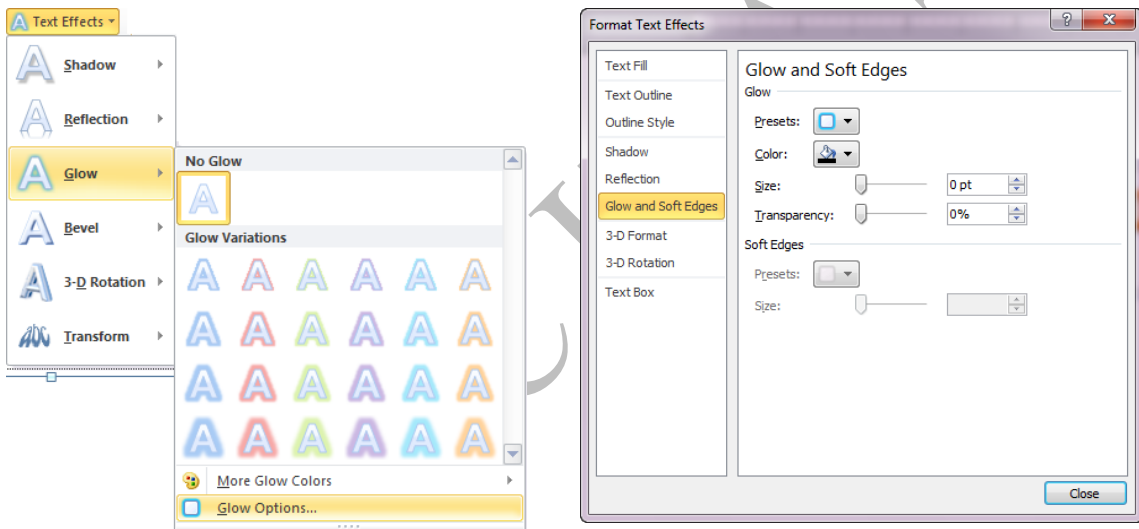
Shadow: tạo bóng đổ bên trong (Inter), bên ngoài (Outer), phối cảnh (Perspective) cho đối tượng. Khi click chọn **Shadow Options**, hộp thoại **Format Text Effects** cho phép bạn thiết lập bóng đổ tùy ý.



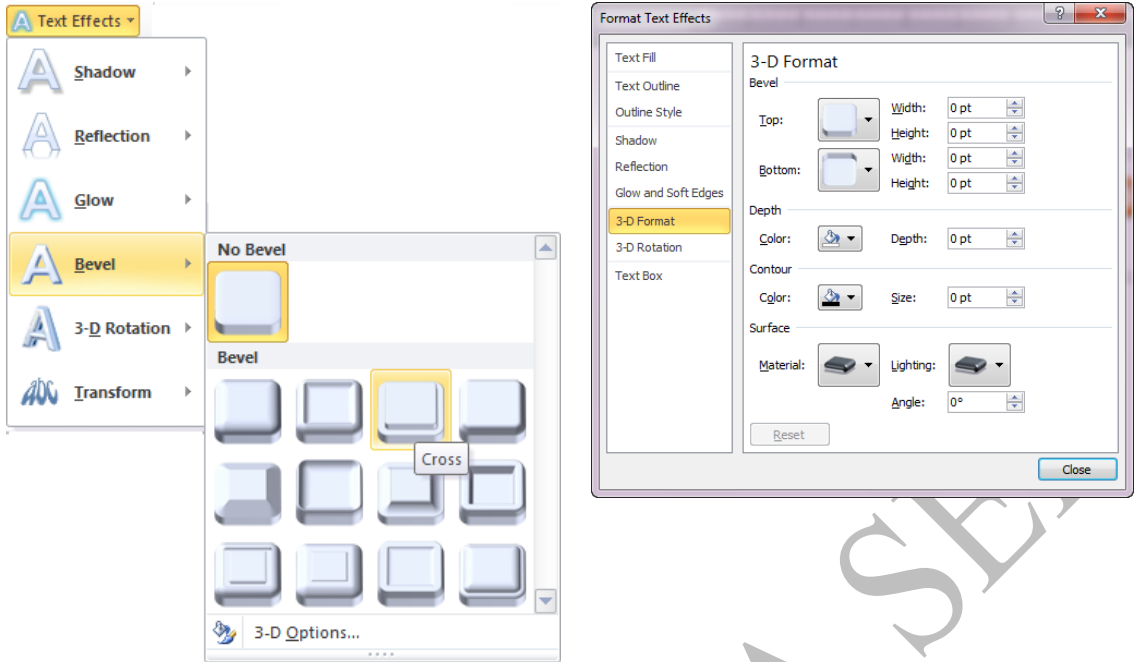
Reflection: độ phản chiếu. Click chọn **Reflection Options** để tùy chỉnh.



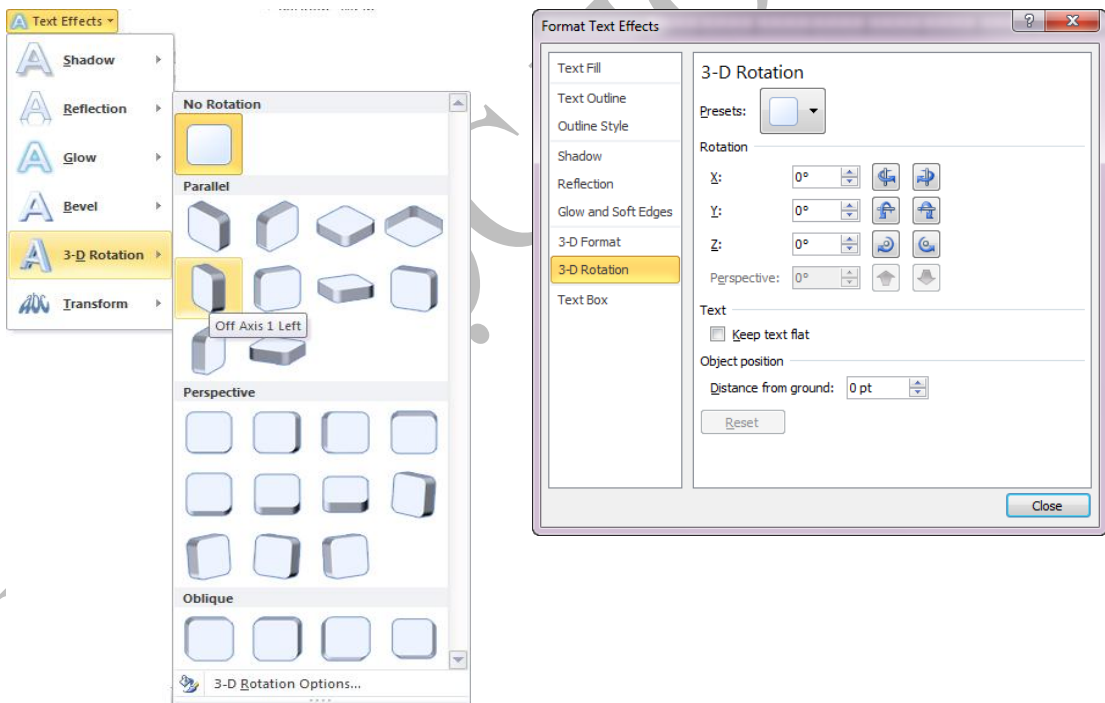
Glow: độ phát sáng. Click chọn **Reflection Options** để tùy chỉnh.



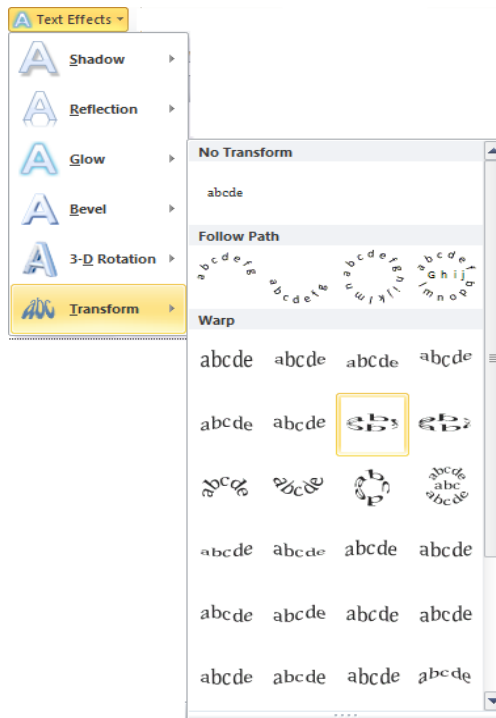
Bevel: tạo gờ nổi bề mặt và biên. Click chọn **3-D Options** để tùy chỉnh.



3-D Rotation: tạo góc xoay trong không gian 3 chiều. Click chọn **3-D Rotations Options** để tùy chỉnh.

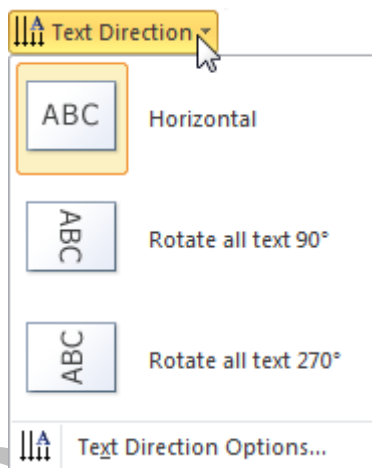


Transform: tạo vùng bao phủ (Wrap), đường dẫn (Path) cho văn bản.



d. Thay đổi hướng trình bày WordArt

Để thay đổi hướng trình bày WordArt, bạn sử dụng nút lệnh **Text Direction** trong nhóm lệnh **WordArt Styles**.



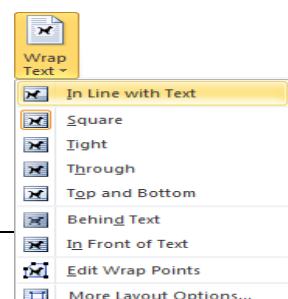
Ví dụ:



e. Thay đổi cách bao phủ văn bản xung quanh WordArt (Wrap Text)

Để thay đổi cách bao phủ văn bản xung quanh WordArt, trong nhóm **Arrange**, click chọn nút lệnh **Wrap Text**.

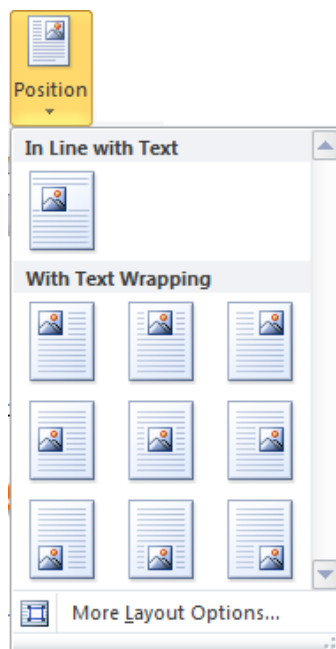
- *In Line with Text*: WordArt di chuyển dọc theo văn bản.
- *Square*: văn bản xung quanh Word theo dạng hình vuông
- *Tight*: văn bản bao sát WordArt
- *Through*: văn bản xuyên qua WordArt.
- *Top and Bottom*: văn bản ở bên trên và bên dưới WordArt.
- *Behind Text*: WordArt ở phía sau văn bản.



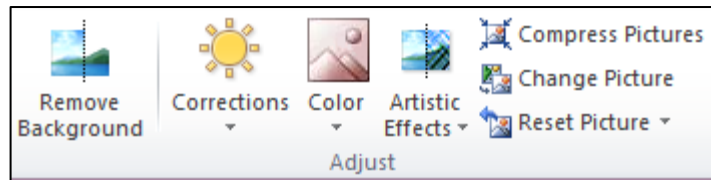
– *In Front of Text*: WordArt ở phía trước văn bản.

f. Thay đổi vị trí của WordArt so với văn bản (Position)

Để thay đổi vị trí của WordArt so với văn bản, trong nhóm **Arrange**, click chọn nút lệnh **Position**. Văn bản tự động thiết lập bao phủ xung quanh đối tượng. Dựa theo vị trí của đối tượng trong các tùy chọn minh họa mà bạn chọn cách bố trí cho phù hợp.



* Ngoài các thao tác trên, WordArt cũng có một số thao tác hiệu chỉnh giống như khi hiệu chỉnh hình vẽ như: thay đổi hướng trình bày WordArt (Text Direction), thay đổi thứ tự của các đối tượng (Order), gom hoặc rã nhóm các đối tượng (Group hoặc Ungroup), quay và lật đối tượng (Rotate or Flip) [Xem tại Chương 6 – mục II.3.d, e, f, g, h]



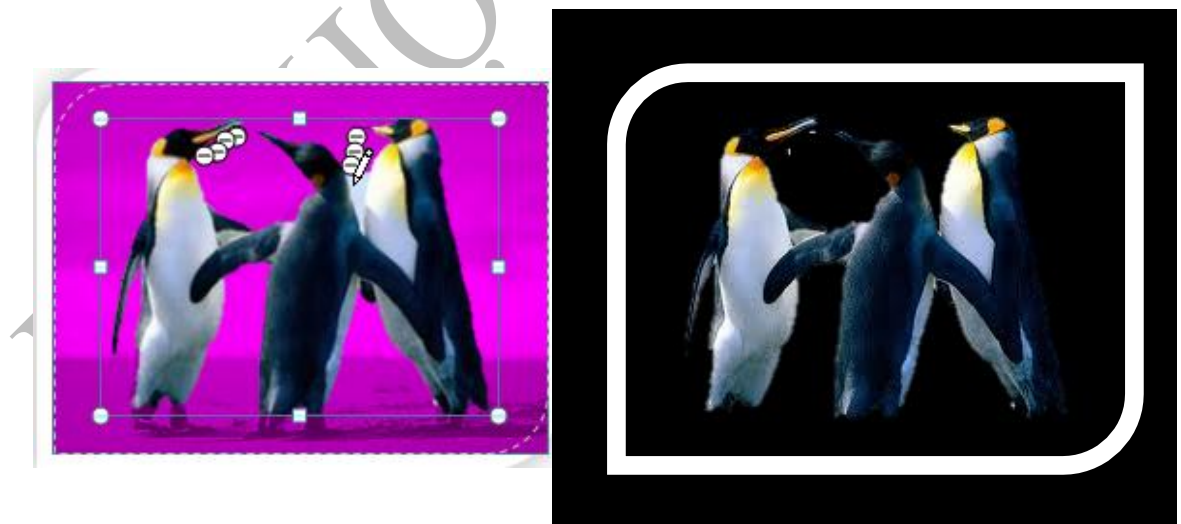
Để tách hình nền cho ảnh, bạn chọn ảnh và click chọn nút **Remove Background**. Khi đó, thanh công cụ **Picture Tools** sẽ có sự thay đổi và nền background của ảnh chuyển sang màu hồng – chính là lớp mặt nạ (Mask)



Bạn click chọn **Mark Areas to keep** hoặc **Mark Areas to Remove** trên tab **Background Removal**. Khi đó con trỏ sẽ biến đổi thành cây bút chì.

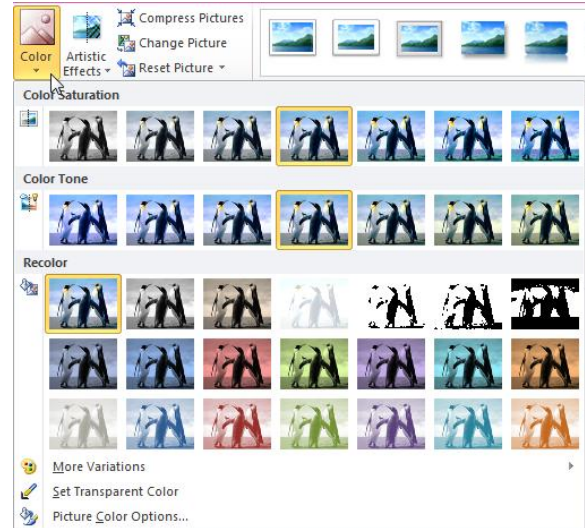
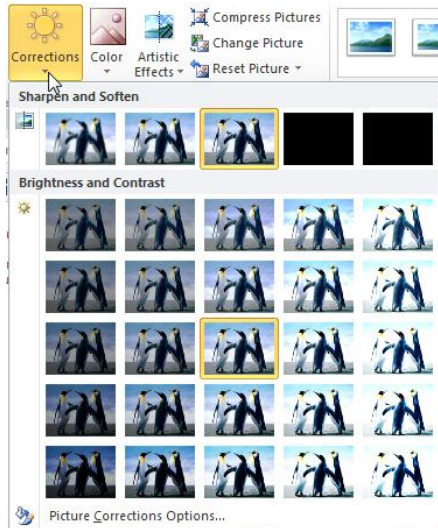
Click con trỏ vào nền của ảnh, tại những vùng muốn giữ lại hoặc muốn loại trừ màu hồng xâm phạm.

Xong click nút **Keep Changes**. Hình nền đã được loại bỏ.



Để thay đổi độ sắc nét (**shapen**) của ảnh hay điều chỉnh ánh sáng (**brightness**), độ tương phản (**contrast**) của ảnh, bạn chọn lệnh **Corrections**.

Muốn thay đổi màu sắc của ảnh, bạn chọn lệnh **Color**



Hay muốn nén hình ảnh để giảm bớt dung lượng cho file tài liệu, bạn sử dụng nút lệnh **Compress Pictures**.

b. Thay đổi style cho hình ảnh

Click vào mũi tên bên phải của nhóm lệnh **Picture Styles**, bạn sẽ thấy một danh sách các kiểu hiển thị. Bạn click chọn một kiểu, hình ảnh sẽ được hiển thị theo tùy chọn của bạn.



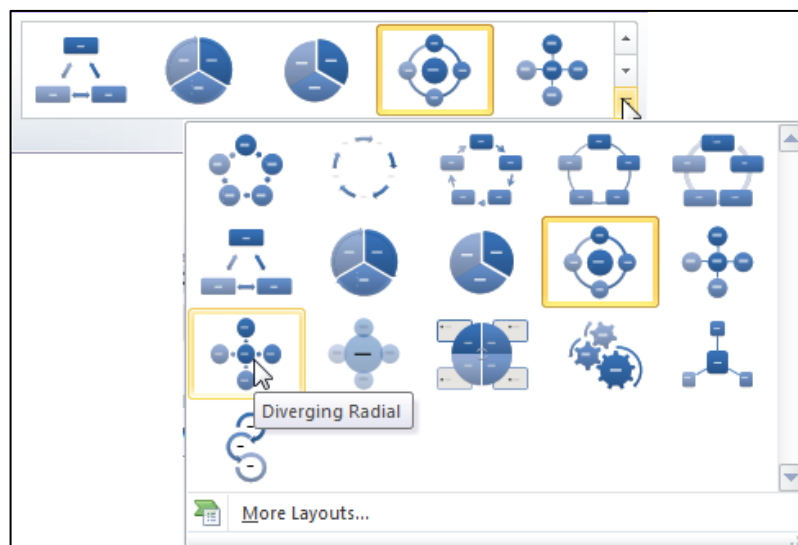
c. Thêm đường viền cho hình ảnh

Để thêm/thay đổi màu của đường viền cùng với kiểu đường viền và độ dày đường viền, bạn sử dụng lệnh **Picture Border**.

b. Thay đổi dạng sơ đồ

Với mỗi một loại sơ đồ sẽ có nhiều hình dạng khác nhau. Để thay đổi dạng khác của cùng một loại sơ đồ:

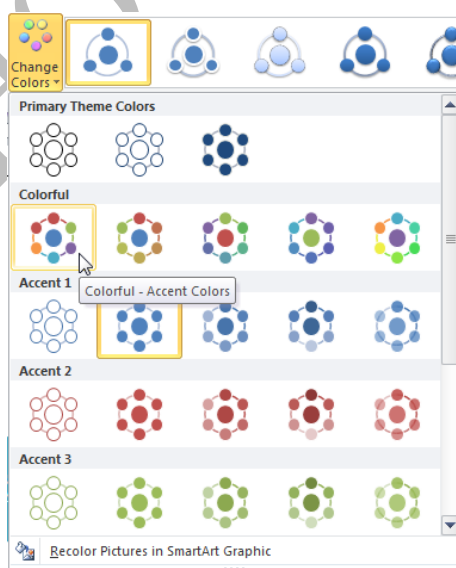
- Trên tab **Design**, click chọn mũi tên bên phải trong nhóm lệnh **Layouts**, một danh sách các dạng của loại biểu đồ hiện tại được hiển thị.



- Bạn chọn một dạng sơ đồ muốn thay đổi (Ví dụ với sơ đồ trên sẽ được chuyển từ dạng **Radial Cycle** sang **Diverging Radial**).

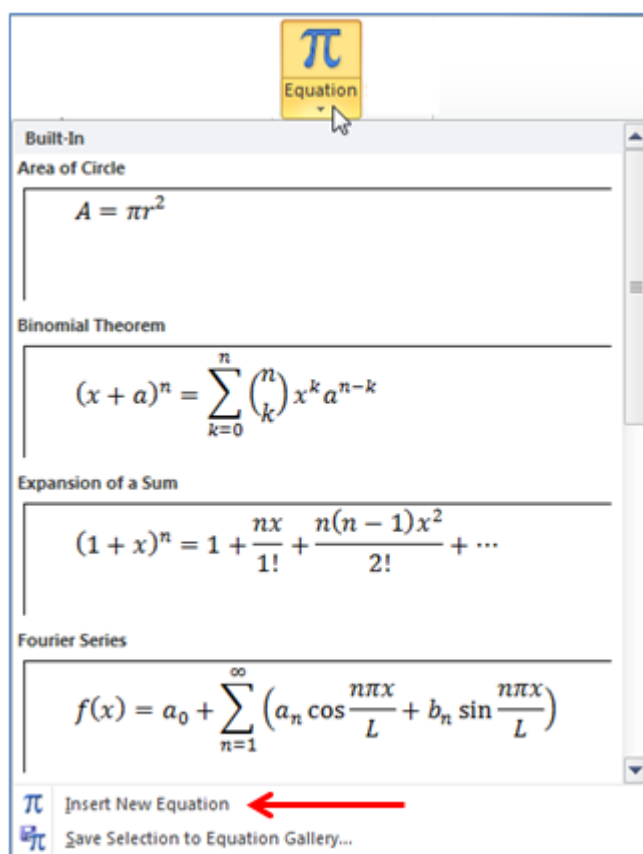
c. Thay đổi màu sắc cho sơ đồ

- Trên tab **Design** > chọn nhóm lệnh **SmartArt Styles** > chọn nút lệnh **Change Color** để thay đổi màu sắc cho sơ đồ



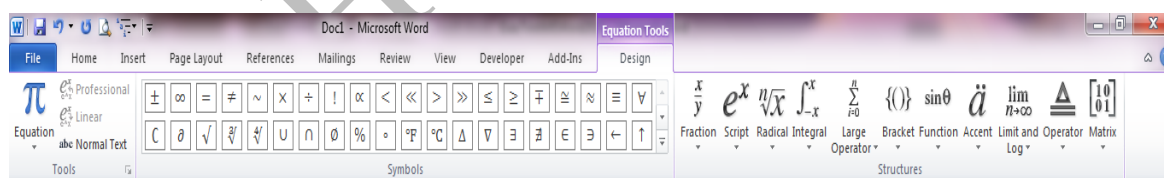
d. Thay đổi style của sơ đồ

Bạn click vào mũi tên bên phải của nhóm lệnh **SmartArt Styles** để thay đổi style cho sơ đồ



Nếu bạn không muốn sử dụng các công thức có sẵn, bạn có thể tự tạo công thức của riêng mình bằng cách click vào lệnh **Insert New Equation**

Khi đó, tại vị trí diêm chèn con trỏ sẽ xuất hiện khung đồng thời thanh công cụ **Equation Tools** sẽ xuất hiện cùng với tab **Design** để sẵn sàng cho bạn nhập công thức



Trên tab **Design** được chia thành 3 nhóm lệnh sau:

- **Nhóm lệnh Tools**: cho phép biểu diễn công thức đã chèn dưới các dạng khác nhau.
- **Nhóm lệnh Symbols**: cho phép bạn nhập các phép tính, các ký hiệu toán học (phép toán so sánh, ký hiệu Hy Lạp, ký hiệu lượng giác,...)
- **Nhóm lệnh Structure**: cho phép nhập các dạng công thức khác nhau (tích phân, lượng giác, phân số, căn thức,...)

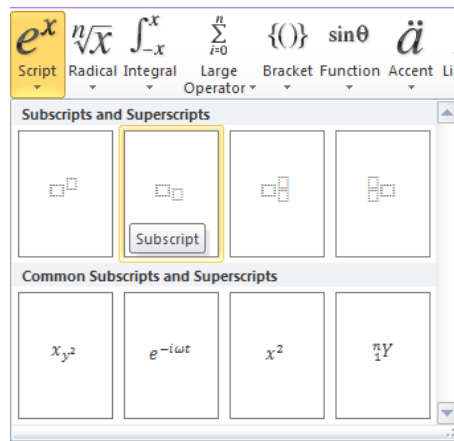
Tùy vào từng biểu thức toán học của mình mà bạn chọn các phép tính, ký hiệu hay các công thức trên từng nhóm lệnh cụ thể.

Ví dụ: bạn muốn nhập công thức:

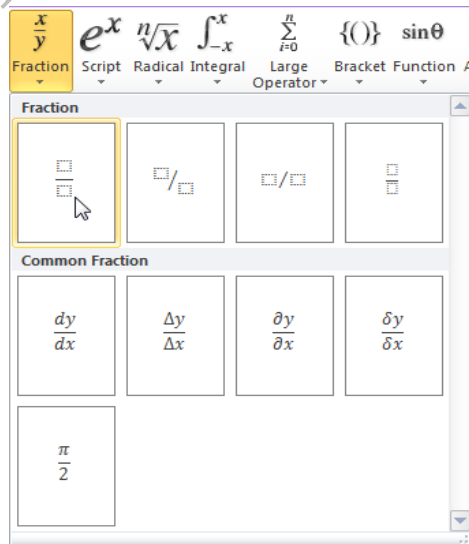
$$x_{1,2} = \frac{-b \pm \sqrt{\Delta}}{2a}$$

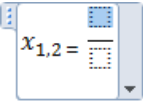
Bạn thực hiện như sau:

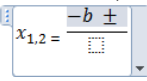
- Đặt con trỏ vào vị trí muốn chèn công thức
- Click chọn lệnh **Equation** trong nhóm lệnh **Symbols** của tab **Insert** > chọn **Insert New Equation**.
- Khi đó, tại vị trí điểm chèn con trỏ xuất hiện khung
- Bạn chọn lệnh **Script** trong nhóm lệnh **Structures** trên tab **Design** của thanh công cụ **Equation Tools** > chọn **Subscript**



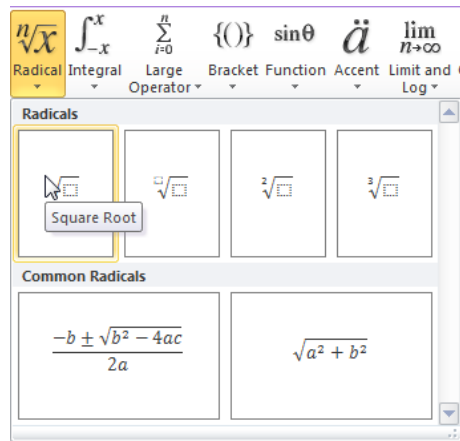
- Tại điểm chèn con trỏ lúc này xuất hiện khung , bạn di chuyển hoặc click con trỏ lên ô vuông lớn và gõ phím **x** rồi đến ô vuông nhỏ và gõ phím số **1,2**
- Tiếp tục, bạn đánh dấu =
- Chọn tiếp bạn chọn lệnh **Fraction** trong nhóm lệnh **Structures** > chọn **Stacked Fraction**.

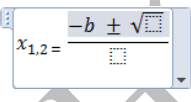


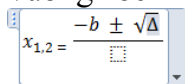
- Tại vị trí con trỏ là khung , bạn tiếp tục di chuyển con trỏ đến tử số (ô vuông bên trên) và đánh nội dung là $-b$ và chọn ký hiệu \pm từ nhóm lệnh

Symbols 

- Tiếp tục chọn lệnh **Radical** trong nhóm lệnh **Structures** > chọn **Square Root**.



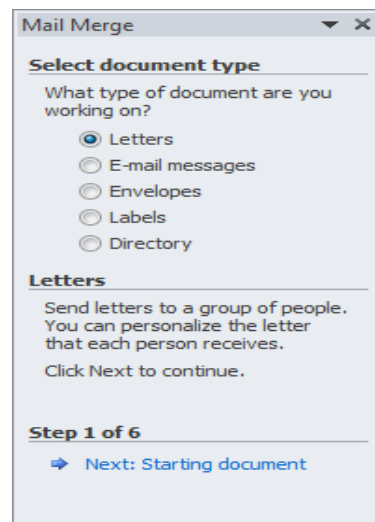
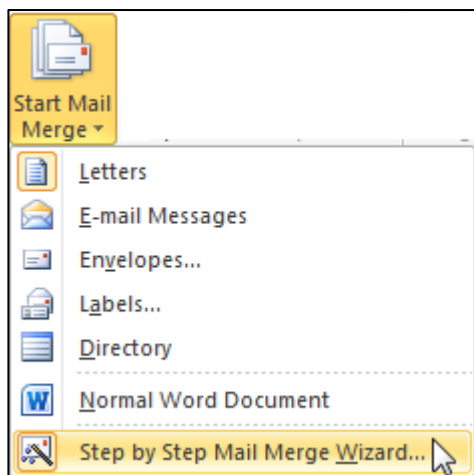
- Khung ngay vị trí con trỏ là , bạn di chuyển con trỏ vào bên ô vuông bên trong dấu $\sqrt{\quad}$ và chọn ký hiệu Δ từ nhóm lệnh **Symbols**



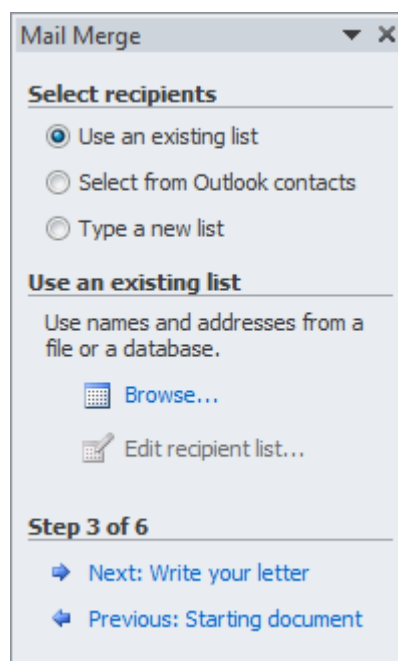
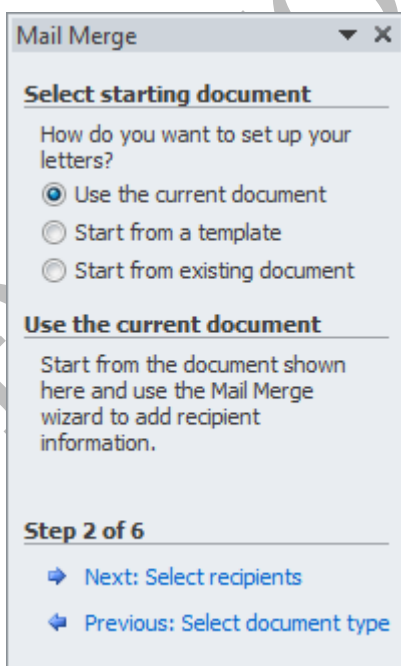
- Di chuyển cho con trỏ đến mẫu số (ô vuông bên dưới) và đánh nội dung là $2a$
- Nhập công thức xong, bạn click con trỏ vào vị trí bất kỳ trong vùng soạn thảo tài liệu, bạn sẽ có được công thức như yêu cầu.

$$x_{1,2} = \frac{-b \pm \sqrt{\Delta}}{2a}$$

Để hiệu chỉnh công thức toán học, bạn click con trỏ lên công thức và di chuyển đến thành phần muốn hiệu chỉnh và sử dụng các ký hiệu hay nút lệnh trên tab **Design** của thanh công cụ **Equatation Tools** để hiệu chỉnh lại giống như lúc bạn nhập công thức..

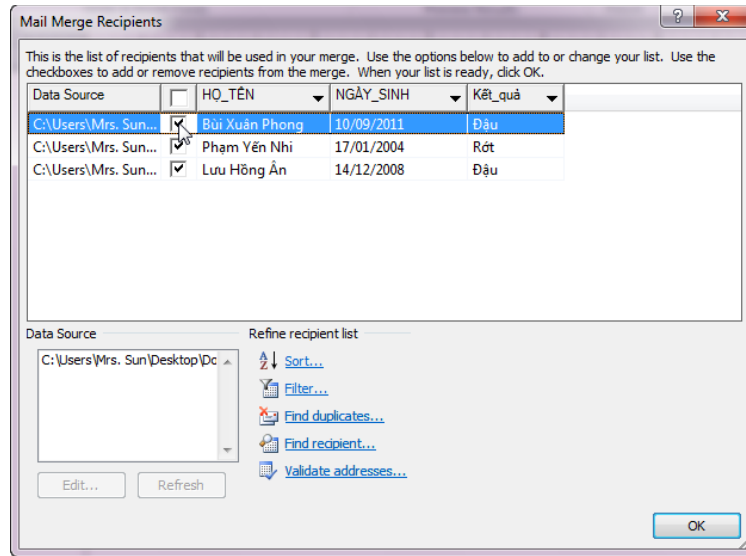


- **Bước 1 (Step 1 of 6)**: bạn chọn loại tài liệu muốn trộn và in thư (Ví dụ chọn Letters là loại văn bản thông thường). Click Next để sang bước 2.
 - **Bước 2 (Step 2 of 6)**: Chọn tài liệu mẫu mà bạn muốn trộn và in thư. Có ba lựa chọn sau:
 - ***Use the current document***: trộn và in thư bằng tài liệu hiện hành. Nếu như bạn đã mở sẵn tài liệu mẫu thì sử dụng lựa chọn này, nếu không thì sử dụng tùy chọn thứ ba.
 - ***Start from a template***: trộn và in thư với một template do Word cung cấp.
 - ***Start from the existing document***: trộn và in thư với tài liệu mẫu đã được lưu trước đó
- Bạn chọn ***Use the current document*** và click Next để sang bước 3.

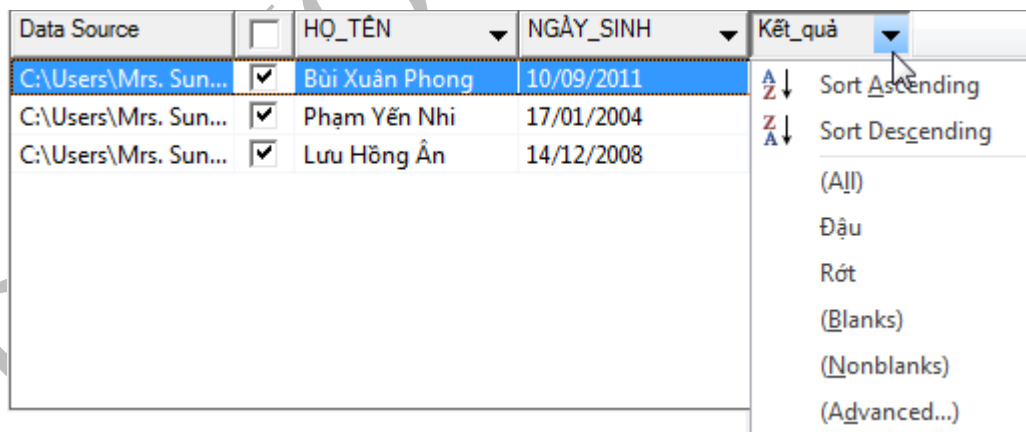


- **Bước 3 (Step 3 of 6)**: Chọn danh sách người nhận (nguồn dữ liệu)

- **Use an existing list:** Sử dụng tập tin danh sách người nhận đã được soạn sẵn (tập tin nguồn dữ liệu)
 - **Select from Outlook contacts:** Nếu là người thường xuyên sử dụng Outlook, bạn có thể chọn mục này để lấy danh sách từ sổ địa chỉ do Outlook quản lý.
 - **Type a new list:** Tạo một danh sách mới.
- Chọn **Use an existing list** và click chọn Browse để mở tập tin dữ liệu nguồn. Bạn chọn đường dẫn đến vị trí lưu trữ tập tin dữ liệu nguồn và click Open.
- Hộp thoại Mail Merge Recipients xuất hiện.

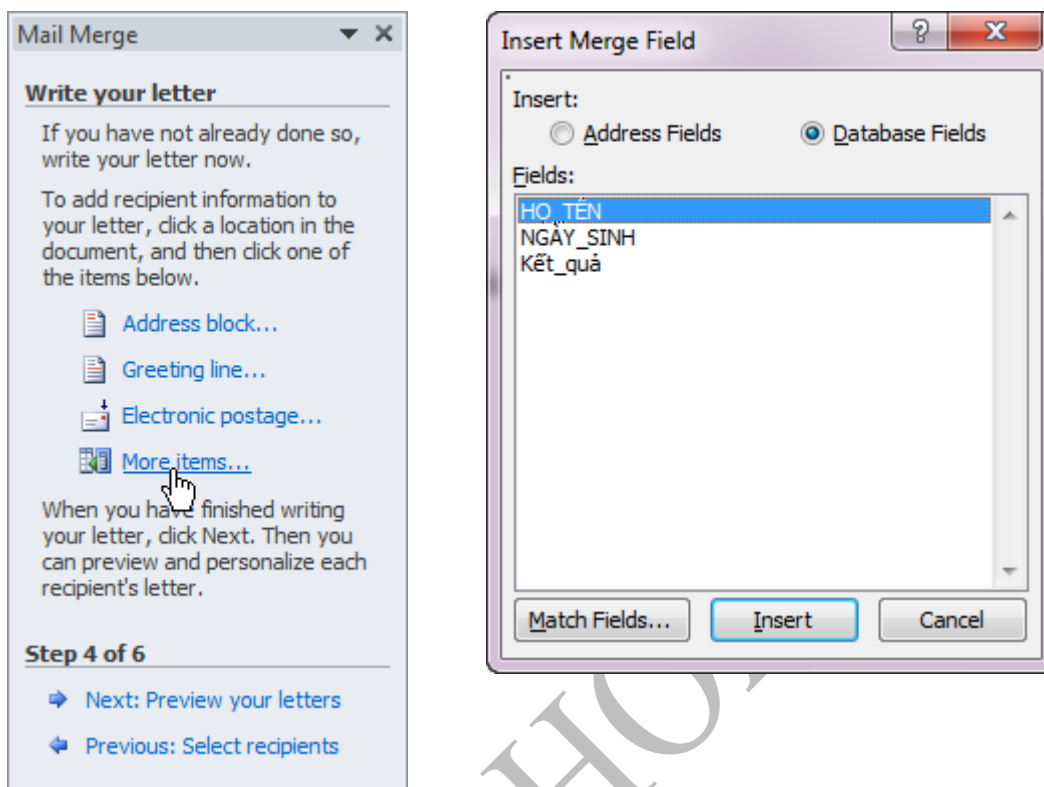


- Bạn có thể đánh dấu vào check box trước tên của những người mình muốn gửi hoặc cũng có thể click chuột vào mũi tên phía trên cột để lọc ra danh sách người nhận theo một tiêu chí nào đó.

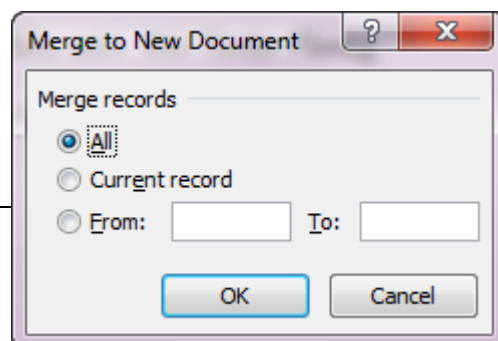


- Bạn có thể tìm hiểu thêm về các chức năng tương tự như *Sort*, *Filter*, *Find duplicates*, *Find recipient*, *Validate addresses*...
- Tùy chọn xong, bạn click **OK** để trở lại khung ở bước 3.
- Bạn có thể chọn lại danh sách (tập tin dữ liệu nguồn) khác bằng cách click chọn lên **Select a different list...**
- Bạn chọn **Next** để sang bước 4
- **Bước 4 (Step 4 of 6):** Chọn Merge Field

- Đặt con trỏ mouse vào vị trí muốn chèn **Merge Field** và click chọn **More Items** để chọn **Merge Field** để trộn vào tài liệu mẫu



- Chọn mục dữ liệu tương ứng với vị trí bạn muốn chèn thông tin trên tài liệu mẫu, sau đó click nút **Insert** và click nút **Close**.
- Lặp lại các thao tác ở bước 4 cho đến khi hết các Merge Field cần chèn vào tài liệu mẫu.
- Click **Next** để sang bước 5
- **Bước 5 (Step 5 of 6):** Xem kết quả trước khi in
- Dùng các nút và để xem qua tài liệu sau khi trộn.
- Ngoài ra, tại bước này, bạn vẫn có thể điều chỉnh lại danh sách người nhận bằng tùy chọn **Find a recipient...** hoặc xóa trực tiếp một người bằng nút lệnh **Exclude this recipient**
- Click **Next** để sang bước 6 – hoàn tất việc trộn.
- **Bước 6 (Step 6 of 6):** In ấn tài liệu
- Bạn có thể sử dụng tùy chọn **Print...** để in trực tiếp hoặc **Edit individual letters...** để xuất kết quả ra một tập tin với một danh sách nhiều người nhận, bạn được quyền lựa chọn in cho người nào với hộp thoại bên **Merge to New Document**.
- **All:** In tất cả
- **Current record:** chỉ in văn bản đang hiển thị trên màn hình.



- **From... to...** chỉ in văn bản từ người thứ mấy đến người thứ mấy.

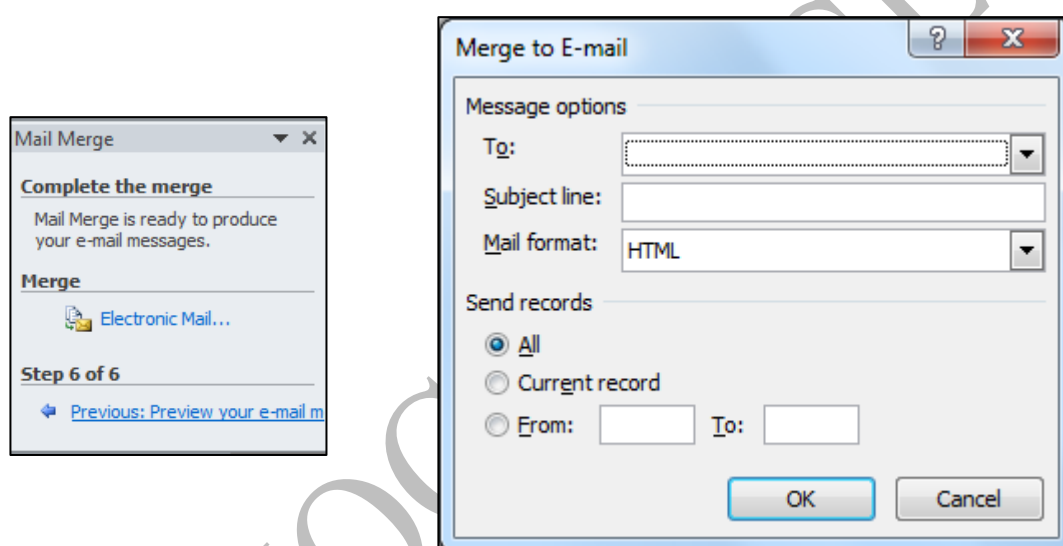
GIỚI THIỆU MỘT SỐ LOẠI TÀI LIỆU KHÁC TRONG KHI TRỘN VÀ IN THƯ.

❖ Email message

Máy của bạn phải kết nối Internet và kết quả trộn thư thay vì được in ra giấy sẽ được gửi đến địa chỉ email có trong danh sách người nhận.

Trong trường hợp trộn và gửi email, từ bước 1 đến bước 5 hoàn toàn tương tự với những gì đã trình bày trong phần trước.

Trong bước 6, click chọn vào **Electronic Mail** để mở hộp thoại **Merge to E-mail** với các tùy chọn sau đây:



- **To:** chọn cột chứa địa chỉ email trong danh sách người nhận
- **Subject line:** tiêu đề email
- **Mail format:** định dạng của mail.
- **Send records:** chỉ định văn bản nào được gửi đi

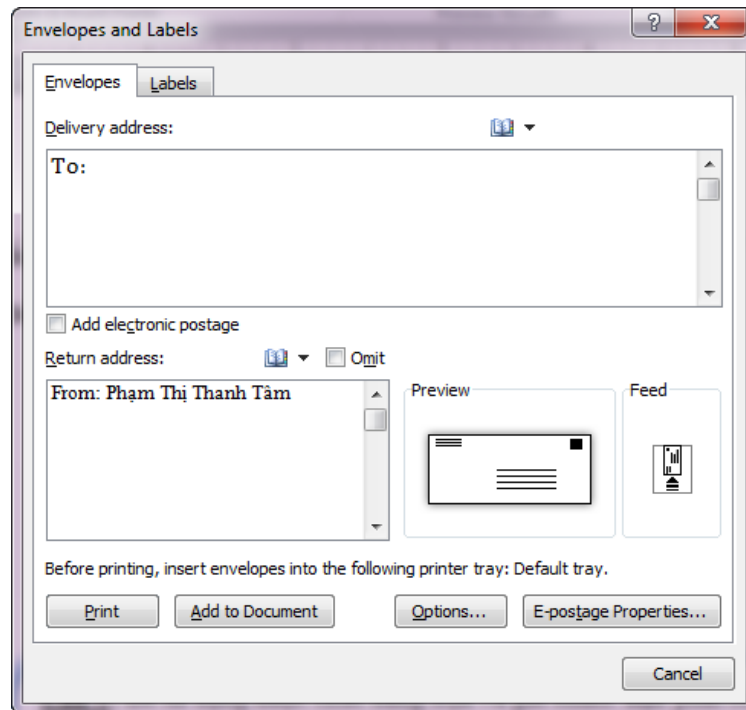
❖ Envelopes:

Bạn có thể hiểu một cách đơn giản, đây là bì thư mà phần ghi địa chỉ gửi đến (**To**) chính là chỗ để điền thông tin có trong danh sách người nhận của bạn.

Như vậy đối với loại văn bản này, trước tiên, bạn cũng phải tạo ra một bì thư mẫu, sau đó tiến hành trộn với danh sách người nhận đã được soạn sẵn.

Cách tạo bì thư mẫu

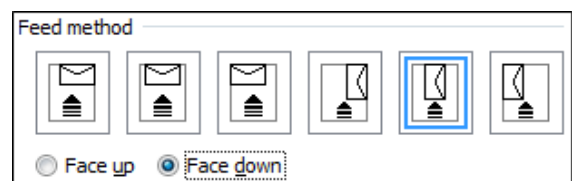
Trên tab Maillings, trong nhóm lệnh Create > chọn Envelopes . Hộp thoại Envelopes và Labels xuất hiện



- **Delivery Address:** ô để điền địa chỉ người nhận. Vì bạn đang muốn trộn địa chỉ từ danh sách người nhận nên phần này để trống.
- **Return Address:** ô để điền địa chỉ người gửi (đánh dấu vào ô Omit nếu như bạn muốn trên bì thư không hiển thị địa chỉ này).
- **Address book** : trong trường hợp bạn có sử dụng Outlook, thì có thể lấy địa chỉ người nhận từ Outlook bằng cách click vào biểu tượng này.
- **Feed:** lưu ý rằng, kích thước bì thư luôn luôn nhỏ hơn khổ giấy mà bạn đưa vào máy in. Do đó, bạn click lên khung Feed để quy định bì thư của bạn sẽ được in tại vị trí nào của tờ giấy.



Chọn vị trí cho mặt trên



Chọn vị trí cho mặt dưới

- **Nút Options:** mở hộp thoại **Envelope Options**. Trong hộp thoại này, bạn có thể quy định kích thước bì thư, font chữ, vị trí đặt địa chỉ người gửi và người nhận.
- **Add to document:** chuyển trang giấy hiện tại thành dạng bì thư.

Sau khi tạo bì thư mẫu, bạn tiến hành từ bước 1 đến bước 6 như đã nói ở phần trước để tiến hành trộn thông tin người nhận lên bì thư của mình và in ra.

Bài tập 1. Định dạng văn bản

NU TÂM XUÂN

Trèo lên cây bưởi hái hoa
Bước xuống vườn cà hái nụ tầm xuân
Nụ tầm xuân nở ra xanh biếc
Em có chồng anh tiếc lắm thay.

Ba đồng một mớ trâu cày
Sao anh chẳng hỏi những ngày còn không
Bây giờ em đã có chồng
Như chim vào lồng như cá cắn câu.

Cá cắn câu biết đâu mà gỡ
Chim vào lồng biết thuở nào ra
Trèo lên cây bưởi hái hoa
Bước xuống vườn cà hái nụ tầm xuân

Bài tập 2. Định dạng văn bản theo mẫu

THI HÔNG

Tấp tễnh người đi, tở cũng đi
Cũng lều cũng chõng cũng vô thi
Tiền chân còn mất ba đồng lẻ
Sờ bụng, thấy không một chữ gì.

Học đã sôi cơm nhưng chưa chín
Thi không ăn ớt thể mà cay
Bốn kỳ trọn vẹn thêm kỳ nữa
Á, Ồ, U, Ồ, ngọn bút chì.

Mong gì kỳ này đầu đội mũ
Biết rõ anh em, chẳng chắc rồi
Ví phỏng còn thi còn học mãi
Toi cơm tốn vải, hại mà thôi.

Bụng buồn còn biết nói năng chi
Đệ nhất buồn là cái hông thi
Một việc văn chương thôi cũng hông
Trăm năm thân thể có ra gì.

Vô danh.

Bài tập 3. Định dạng văn bản

OUTLINE A DOCUMENT

You can outline a document in two way:

Organize a new document by typing headings in outline view. Word formats the headings with the built-in heading styles. You can numbering the headings if you like. Type topics and subtopics in a numbered list. The text will not be formatted with heading styles.

What do you want to?

- Organize headings in a new document
- Create a numbered list in outline format

ORGANIZE A NEW DOCUMENT IN OUTLINE VIEW

- 1) In a new document, switch to *outline view*
- 2) Type each heading and press ENTER. Word formats the headings with the built-in heading style. Heading1
- 3) To assign a heading to a different level and apply the corresponding heading style, drag the heading's-or + symbol
 - To *demote* to a lower level, drag to the right, or press TAB
 - To *promote* to a higher level, drag to the left, or press SHIFT+ TAB
 - To move a heading to a different location, drag up or down. Subordinate text below the heading moves with the heading
- 4) When you're satisfied with the organization, switch to *normal view* to add detailed text, graphics, and other items

Tip: You can number the headings by using the Heading Numbering command on the Format menu

CREATE A MULTILEVEL LIST

Multilevel lists can have up to nine levels. You can modify the number or bullet format of each level.

Bài tập 4. Tạo các mẫu tài liệu sau:



UBND TP. Hồ Chí Minh
Sở Thương Mại
---oOo---
Số: ____ / ____

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc



TM. BAN THƯƠNG VỤ
ĐOÀN CÔNG TY INEXCO
BÍ THƯ

NGUYỄN ĐỨC NGUYỄN



STT	TÊN THIẾT BỊ	NƯỚC SX	ĐVT	SL	ĐƠN GIÁ	THÀNH TIỀN
1	Máy vi tính Pentium IV	Đài Loan	Cái	05	\$1,050.00	\$5,250.00
2	Máy in HP	Singapore	Cái	01	\$440.00	\$400.00
3	UPS 500A	Nhật	Cái	01	\$150.00	\$150.00

Bài tập 5. Tạo văn bản theo mẫu

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc
---oOo---

SƠ YẾU LÝ LỊCH
---oOo---

Họ và tên: _____ Phái _____

Ngày sinh: _____ Tại: _____

Địa chỉ hiện tại: _____

Quận / Tỉnh: _____

Ngày ... tháng ... năm

Người làm đơn

Bài tập 6. Tạo văn bản theo mẫu

UBND TP HCM
SỞ ABC

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc

**THỐNG KÊ VPĐD CÁC CÔNG TY NƯỚC NGOÀI
TRÊN ĐỊA BÀN THÀNH PHỐ HỒ CHÍ MINH**
(Đến ngày 30/06/1996)

1. Phân theo lĩnh vực hoạt động

<i>Thương mại</i>	786
<i>Dịch vụ, tư vấn</i>	210
<i>Sản xuất</i>	109
<i>Hàng không</i>	30
<i>Ngân hàng</i>	49
<i>Tàu biển</i>	37
<i>Khác</i>	87

2. Số người lao động Việt Nam: 2641

3. Số người lao động nước ngoài: 2010

4. Tổng cộng có 1310 văn phòng đại diện

5. Phân theo thị trường

STT	THỊ TRƯỜNG	SỐ VPĐD
1	Khối ASEAN	346
2	Châu Á (trừ khối ASEAN)	225
3	Châu Âu	466
4	Châu Mỹ	95
5	Khu vực khác	178
	Tổng cộng	1310

Bài tập 7. Tạo văn bản theo mẫu

**INDUSTRIAL LAND I N SUB-DIVISION
AREAS FROM 13,500 SQ. FT. TO 84,600 SQ.FT.**

LOCATION: These sites are ideally located on the northern side of *Ainsley Street*, on the corner of *Cairns Avenue* and *Dudley Road*. Rhodes station is within a few minutes' walking distance.

ZONE: "B" Class Industrial.

DESCRIPTION: This is level site of approximately two acres and has been sub -divided into six smaller industrial sites, with frontages ranging from 75' to 103', and areas ranging from 13,500 sq. ft. to 15,000 sq. ft.

<u>LOT NO:</u>	AREA(sq.ft)	FRONTAGE	PRICE
1	13,500	75'.....	\$44,000
2	13,600	100'.....	\$41,000
3	15,000	103'.....	\$45,000
4	15,000	96'.....	\$45,000
5	14,000	85'.....	\$42,000
6	13,500	100'.....	\$47,000

(See plann overleaf. Lot No.1 also has 175' frontage to Cairns Avenue, and Lot No.6 has 70' frontage to Dudley Road, and lot No.4 must have 69' frontage to Ansley Street.)

COMMENT: To our knowledge these sites are the only ones of their size available in this popular location. Rhodes is generally considered the centre of the metro politan area, making it an ideal distribution point.

For further details or an inspection please contact the

Industrial Department

L. J. HICKSON LIMITED

Bài tập 8. Tạo văn bản theo mẫu

Cộng Hòa Xã Hội Chủ Nghĩa Việt Nam
Độc Lập – Tự Do – Hạnh Phúc

Ngày 17 tháng 12 năm 1997

ĐƠN XIN SẢN XUẤT KINH DOANH

Kính gửi: - Ủy ban Nhân dân Huyện Thủ Đức
- Phòng Công-Thương nghiệp Huyện Thủ Đức

Tôi tên là Lê Minh Phương, hiện đang cư ngụ tại 107/1 bis đường Điện Biên Phủ, P.17, Q. Bình Thạnh.

Vừa qua, tôi có ký hợp đồng thuê mặt bằng với huyện Thủ Đức tại số 5/15 đường Trường Sơn để sản xuất hàng thực phẩm chế biến. Nhưng vì thời gian chuẩn bị sản xuất quá ngắn (1 tháng: do bên huyện quy định trong hợp đồng thuê nhà xưởng) nên việc xin giấy phép đăng ký kinh doanh của chúng tôi không thể tiến hành kịp trong thời gian đó.

Vậy nay kính mong UBND và phòng Công-Thương nghiệp huyện Thủ Đức cho tôi được sản xuất thử trong thời gian đang tiến hành xin giấy phép kinh doanh chính thức. Khi được giấy phép tôi xin nộp đủ hồ sơ pháp lý cho huyện. Tôi xin bảo đảm thực hiện đúng các quyên tắc về quản lý và kinh doanh theo đúng chế độ hiện hành.

Kính mong sự chấp thuận và giúp đỡ của UBND và phòng Công-Thương nghiệp huyện Thủ Đức.

Trân trọng
Kính đơn

Lê Minh Phương

Bài tập 9. Tạo văn bản theo mẫu

HIỆP HỘI XUẤT NHẬP KHẨU
VÀ HỢP TÁC ĐẦU TƯ (INFOTRA)

Cộng Hòa Xã Hội Chủ Nghĩa Việt Nam
Độc Lập – Tự Do – Hạnh Phúc

TRƯỜNG HOA SEN

Số: _____

TP Hồ Chí Minh, ngày 26 tháng 4 năm 1997

Kính gửi: **Giám Đốc Sở Giáo dục và Đào tạo TP HCM**

Trích yếu: **Tờ trình xin mở ngành đào tạo quản lý**

Trong thành phố hiện nay có một nhu cầu rất lớn đào tạo quản lý kinh doanh cho các cán bộ cơ quan, xí nghiệp, công ty thuộc các thành phần kinh tế. Trường Tin học và Quản lý HOA SEN, căn cứ vào Quyết định thành lập trường số 257/QĐ-UB ngày 12/08/1991 của UBND TP HCM, mong muốn được đóng góp vào nhiệm vụ đào tạo này.

Do đó, trường HOA SEN kính xin lãnh đạo Sở Giáo dục và Đào tạo TP HCM **cho phép trường tiến hành đào tạo các khóa ngắn hạn về quản trị kinh doanh trong khuôn khổ hiện nay của trường** với mục tiêu và nội dung đính kèm

Trân trọng kính chào đoàn kết.

KT. Hiệu Trưởng

Phó Hiệu trưởng

Trần Thị Hồng

Nơi nhận

- Như trên
- Chủ tịch HĐQT “đề báo cáo”
- Chủ tịch HĐQT HSP “đề báo cáo”

Bài tập 10. Tạo văn bản theo mẫu

ĐẢNG ỦY KHỐI KTĐN

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc Lập – Tự Do – Hạnh Phúc

Đảng ủy VITEXCO

Ngày 17 tháng 04 năm 1993

THƯ MỜI

Ban Chấp hành Đảng bộ Công ty May mặc xuất khẩu – Vitexco kính mời các đồng chí:

- Ban Giám Đốc,
- Chủ tịch Công đoàn và Bí thư Đoàn TNCS Công ty Vitexco ,
- Trưởng, Phó phòng ban của Công ty và các đơn vị trực thuộc (Xí nghiệp, Xưởng, Cửa hàng, Trạm, ...),
- Bí thư các chi bộ phận,
- Các đồng chí là chuyên viên bậc hai trở lên .

Đến dự buổi báo cáo thời sự về tình hình Đông Âu và Liên Xô do Đảng Ủy Khối Kinh tế đối ngoại tổ chức.

Báo cáo viên: Đồng chí Lê Văn Triết, Phó Giám đốc trung tâm Giáo dục Chính trị thành phố Hồ Chí Minh

Thời gian: Từ 7^h30 đến 11^h30, ngày 21/04/93

Địa điểm: Hội trường Sở Kinh tế Đối ngoại, số 15 – 47 Bến Chương Dương, Q.1, TP HCM

Đề nghị lãnh đạo các đơn vị tạo điều kiện cho các đồng chí thuộc diện trên thu xếp thời gian về dự

ĐẢNG ỦY VITEXCO

Bí thư

LÊ VĂN TRIẾT

Bài tập 11. Tạo văn bản theo mẫu

ĐOÀN TNCS HỒ CHÍ MINH CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
SỞ THƯƠNG MẠI Độc Lập – Tự Do – Hạnh Phúc

ĐOÀN CÔNG TY INEXCO
Số: 0121/QĐCĐ

TP Hồ Chí Minh, ngày 05 tháng 11 năm 1995

NGHỊ QUYẾT

V/v THÀNH LẬP CHI ĐOÀN ĐOÀN TNCS XÍ NGHIỆP VITEXIM
CHỈ ĐỊNH BAN CHẤP HÀNH CHI ĐOÀN VÀ GIAO NHIỆM VỤ

- Căn cứ Điều 7 chương II, Điều lệ Đoàn TNCS Hồ Chí Minh,
- Căn cứ đề xuất của Chi bộ Xí nghiệp VITEXIM và công văn số 120/VITEXIM/95 ngày 24/10/1995 của Ban Giám đốc Xí nghiệp,
- Căn cứ tình hình phát triển lực lượng đoàn viên và nhu cầu của thanh niên hiện nay tại xí nghiệp

BAN THƯỜNG VỤ
ĐOÀN CÔNG TY CÔNG NGHỆ PHẠM XUẤT KHẨU – INEXCO

QUYẾT ĐỊNH

- ĐIỀU 1:** Thành lập Chi đoàn TNCS xí nghiệp VITEXIM là Chi đoàn trực thuộc Đoàn Công ty INEXCO
- ĐIỀU 2:** Chỉ định vào Ban Chấp hành Chi đoàn TNCS xí nghiệp VITEXIM và phân công gồm các đồng chí có tên sau đây :
- | | |
|-----------------------|------------|
| 1. Nguyễn Bảo Sơn | Bí thư |
| 2. Dương Thiều Lệ Thu | Phó Bí thư |
| 3. Võ Tấn Thành | Ủy viên |
- ĐIỀU 3:** Ban Chấp hành Chi đoàn TNCS xí nghiệp VITEXIM có nhiệm vụ tổ chức các hoạt động tại Chi đoàn theo điều lệ Đoàn và định hướng chỉ đạo của Đoàn cấp trên.
- ĐIỀU 4:** Chi đoàn xí nghiệp VITEXIM và các đồng chí có tên trên chịu trách nhiệm thi hành quyết định này.

TM. BAN THƯỜNG VỤ
ĐOÀN CÔNG TY INEXCO
BÍ THƯ

NGUYỄN ĐỨC NGUYỄN

Bài tập 12. Tạo văn bản theo mẫu

MỤC LỤC



TẬP CHÍ SÁNG TÁC NGHIÊN CỨU PHÊ BÌNH VĂN HỌC NGHỆ THUẬT

VẤN

NGUYỄN LINH	Dưới tán lá rừng còn lại.....	Truyện ngắn	5
NGUYỄN QUANG SÁNG	Con khướu sỏ lông	Truyện ngắn	12
TRANG THÈNHUY	Về nhà trước cơn mưa	Truyện ngắn	15
NGUYỄN ĐÔNG THỨC	Chia tay.....	Truyện ngắn	28
VÕ ĐẮC DANH.....	Nơi ấy bây giờ.....	Truyện ngắn	37
HIỀN PHƯƠNG	Trăng	Truyện ngắn	61
CÓ ĐÔNG	Ghé bên.....	truyện ngắn trung Quốc	96
KAWABATA YASUNARI	Cánh tay.....	truyện ngắn Nhật Bản	119

THƠ

VĂN CAO	Gởi người em biển xa.....	25
NGUYỄN LẬP EM	Còn mãi tình yêu.	26
NGUYỄN THÁI DƯƠNG	Chiếc lá tương tư.	30

PHÊ BÌNH – NGHIÊN CỨU – TRAO ĐỔI ĐỌC SÁCH

TRẦN THANH ĐẠM.....	Sinh mệnh của văn chương lãng mạng	107
HOÀNG KIM BẢO	Tìm một hướng phát triển cho tập	110
HUY ANH	Đọc “Nguyễn Trường Tộ - con người	117

CÁC MỤC KHÁC

SƠN NAM.....	Giờ chông báo cũ - Suu tầm.....	105
NHẬT CHIỀU	Kỷ niệm 160 năm ngày sinh Lép tônxtôi	110

Bài tập 13. Đóng khung và tô nền văn bản theo mẫu sau:

LUSCIOUS JAM CO. PTY. LTD

239 Campbell Street, Chippendale, NSW. 2008, Tel: 213489

Công nghệ RamBus tăng khối lượng xuất liệu

RamBus hứa hẹn những bo mạch nhanh hơn, rẻ hơn.

Các chi tiết xin liên hệ với:

Trường Cao Đẳng Bán Công HOA SEN
Khoa Đào tạo Ngắn hạn – Phòng C01
Số 8 Nguyễn Văn Tráng, Q. 1, TP. HCM
☎: 8301 877 – 8324 602 – 8323 224
Fax: 8301 878 - ✉: lotus1@hcm.vnn.vn

HOA SEN

HOA SEN

HOA SEN

HOA SEN

HOA SEN

HOA SEN

HOA SEN

Bài tập 14. Tạo văn bản theo mẫu sau

NỘI DUNG THỰC HÀNH

Hàng năm, cứ vào độ cuối thu, lá ngoài đường rụng nhiều và trên không có những đám mây bàng bạc, lòng tôi lại náo nức những kỷ niệm mơn man của buổi tựu trường

Tôi quên thế nào được những cảm giác trong sáng ấy nảy nở trong lòng tôi như mấy cành hoa tươi mỉm cười giữa bầu trời quang đãng

Những ý tưởng ấy tôi chưa lần nào ghi trên giấy, vì hồi ấy tôi không biết ghi và ngày nay tôi không nhớ

hết. Nhưng mỗi lần thấy mấy em nhỏ rụt rè nấp dưới nón mẹ lần đầu tiên đi đến trường, lòng tôi lại tung bừng rộn rã

Buổi mai hôm ấy, một buổi mai đầy sương thu và đầy gió lạnh, mẹ tôi âu yếm nắm lấy tay tôi đi trên con đường làng dài và hẹp. Con đường này tôi đã quen đi lại lắm lần. Nhưng lần này tôi tự nhiên thấy lạ. Cảnh vật xung quanh tôi đều thay đổi,

vì chính lòng tôi đang có sự thay đổi lớn:
Hôm nay tôi đi học



Bài tập 15. Tạo văn bản theo mẫu sau

Chứng bệnh cực kỳ quái gở:

CHẾT ÓC

Khi M. Watson nhai một quả táo, vị chua và tươi mát của nó đã biến thành ... một hình thoi mà anh ta có thể tận tay sờ được

Đối với anh, mùi của con gà rô-ti là ... một chấm, đĩa mì xào là ... hình tròn. Con người kỳ cục này “xem” mùi vị như các mô hình “hình học”!

Bác sĩ cho rằng: M. Watson mắc chứng “loạn giác”. Căn bệnh quái gở này cho phép anh ... “nghe” được màu sắc, hay “thấy” được âm nhạc bằng hình ảnh...

Bài tập 16. Tạo văn bản theo mẫu sau

ĐỊNH VỊ TRIẾT HỌC TRONG NỀN VĂN MINH TRUNG QUỐC

Đối với người phương Tây thấy rằng đời sống dân chúng Trung Quốc thấm nhuần nho học, thì nho học dưới mắt họ hầu như là một tôn giáo. Nhưng thật ra nho học không phải là tôn giáo, cũng như học thuyết Platon hay học thuyết Aristote. Đành rằng *Tứ thư* vốn là thánh kinh của dân tộc Trung Quốc, nhưng trong *Tứ thư*, ta không tìm thấy lịch sử cấu tạo vũ trụ hay đề cập tới thiên đường, địa ngục.

Thật vậy, những từ ngữ triết học và tôn giáo thường hay bị nhầm lẫn. Triết học

và tôn giáo có thể có ý nghĩa hoàn toàn khác nhau tùy theo người. Khi nói đến triết học hay tôn giáo, người ta có thể có nhiều ý nghĩ khác nhau. Về phần tôi, thì điều mà tôi gọi là triết học, là sự suy nghĩ có hệ thống và có ý thức về cuộc sống. Người ta chưa chết thì còn liên quan đến cuộc sống. Nhưng rất hiếm người biết suy nghĩ có ý thức về cuộc sống, và càng hiếm người suy nghĩ có ý thức mà có hệ thống. Nhà triết học phải suy cứu tức là phải suy nghĩ về cuộc sống và diễn tả ý nghĩ của mình cho có hệ thống. (Theo Đại Cương Triết Học Sử Trung Quốc-Fung Yu Lan)

Bài tập 17. Tạo văn bản theo mẫu

Bến đò ngày xưa

Anh Thơ

Trên bến vắng đăm mình trong lạnh lẽo
Vài quán hàng không khách đứng xò ro
Một bác lái ghé buồm vào hút điếu
Mặt bà hàng xù xụ sặc hơi ho.

Đá vọng Phu

Phạm Đình Tân

Nắng chiều soi đá ra vàng
Trên non còn một mình nắng tro vơ
Ăm con nhìn cỏi mịt mù
Mong chồng biết đến bao giờ mới thôi



Bài tập 18. Tạo bảng dữ liệu theo mẫu sau

HOW YOURS SAVING CAN GROW
When investd in
PROPERTY AND BUILDING TRUST

Investment Per year	Value of Investment at end of		
	10 years	20 years	30 years
\$ 200	\$ 2,976	\$ 5,434	\$ 8,900
400	5,954	10,868	17,802
1,000	14,886	17,174	44,506
2,000	29,772	54,348	89,014

WHEAT QUALITY

Average expectation of poduction and use
(million bushel)

	Local Use and Export Flour		Export Wheat		Total Deliveries		
	Hard	Soft	Hard	Soft	Hard	Soft	Total
New South Wales	16.0	15	9.0	15	25	30	55
Victoria	-	25	-	15	-	40	40
South Autralia	4.0	5	16.0	5	20	10	30
Western Australia	2.0	7	10.0	28	12	35	47
Qeensland	8.5	-	6.5	-	15	-	15
Australia	30.5	52	41.5	63	72	115	187

Bài tập 19. Tạo bảng dữ liệu theo mẫu sau

Góc chuyên môn về mạng

DỰ TÍNH GIÁ THÀNH LẬP ĐẶT MẠNG CỤC BỘ

STT	TÊN THIẾT BỊ	ĐVT	SỐ LƯỢNG	ĐƠN GIÁ (USD)	THÀNH TIỀN
1	RAM	MB	40	38.00	1,520.00
2	LAN Card NE200	Cái	11	37.00	407.00
3	BNC	Cái	30	2.00	60.00
4	T-connector	Cái	11	3.00	33.00
5	Terminator	Cái	02	5.00	10.00
6	Coaxial Cable	Mét	50	0.70	35.00
7	File Server	Bộ	01	2,000.00	2,000.00
8	UPS 500VA	Cái	01	170.00	170.00
TỔNG CỘNG					4,235.00

Ngày 20 tháng 11 năm 1997
NGƯỜI LẬP DỰ TÍNH

Bài tập 20. Tạo dữ liệu theo mẫu sau

SỞ THƯƠNG MẠI TP HCM
CTY BÁCH HÓA

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc Lập – Tự Do – Hạnh Phúc

BÁO CÁO BÁN HÀNG

Ngày 15 – 11 – 1997

STT	TÊN HÀNG	Đơn vị tính	GIÁ		SỐ LƯỢNG	LÃI
			NHẬP	BÁN		
1	Cassette Sony	Cái	918,000	1,120,000	8	1,616,000
2	TV JVC	Cái	4,500,000	5,800,000	5	6,500,000
3	Đồng hồ SEIKO	Cái	350,000	420,000	12	840,000
4	Quạt máy NATIONAL	Cái	450,000	645,000	18	3,150,000
Tổng cộng:						12,106,000

TỔNG CỘNG (Viết bằng chữ): _____

TP. Hồ Chí Minh, ngày 16 tháng 11 năm 1997
Cửa hàng trưởng

Bài tập 21. Tạo dữ liệu theo mẫu sau

Công ty CEB

Số: 001/VT

COMPUTER FOR EVERY BODY

YOU THINK – LET IT DO

Kính gửi: Công Ty Xuất Nhập Khẩu

Địa chỉ: 1234 Xô Viết Nghệ Tĩnh, TP. Bình Minh

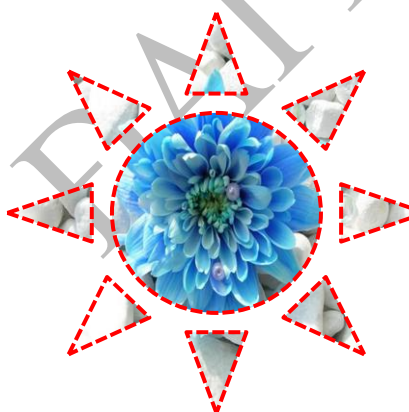
Theo như đề xuất của Quý Công ty, trung tâm chúng tôi sau khi phân tích nhu cầu thực tế của công việc, rất hân hạnh gửi đến Công ty *Bảng dự kiến trang bị phòng máy vi tính* theo như chiết tính các khoản dưới đây để Công ty tiện tham khảo:

STT	THIẾT BỊ	ĐƠN GIÁ	SLG	SỐ TIỀN	TỶ LỆ
1	Pentium 123 MHZ (INTEL), 256 Cache, PCI bus Memory: 32 MB HDD: 2.1 GB FDD: 1.44 MB I/O onboard (1S/2P/1G) Video Card: Trio S364V+ (with 2MB) Monitor: Acer 14" 0,28mm (Non-Interlative) Keyboard: 105 Win95 Enhanced keys Case: Mini Tower w/PS 250W	\$1,500.00	6	\$9,000.00	78.62%
2	EPSON Printer LQ2170 w/cable	\$610.00	2	\$1,220.00	10.66%
3	HP Laser Jet Printer 6P w/toner and cable	\$1,200.00	1	\$1,200.00	10.48%
4	FLOPPY 1.44MB	\$5.50	5	\$ 27.50	0.24%
	TỔNG CỘNG			\$11,447.50	100%

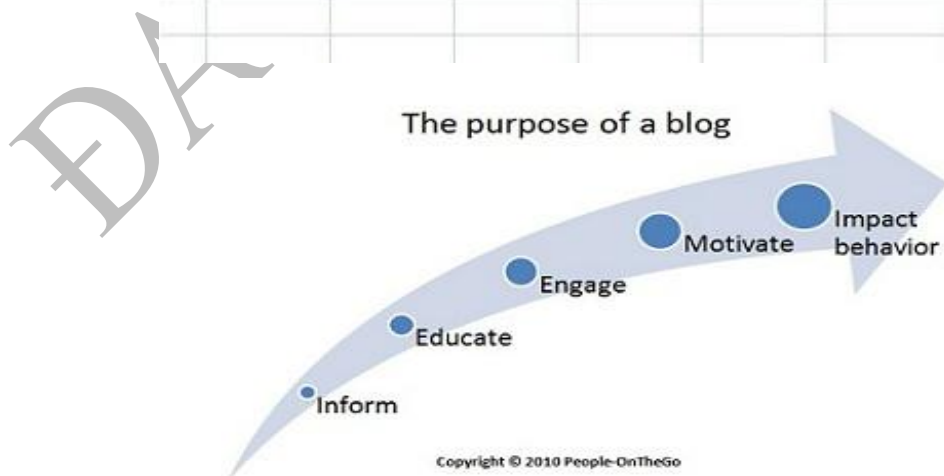
Bài tập 22. Tạo mẫu văn bản sau



Bài tập 23. Sử dụng Shape và WordArt để tạo các mẫu sau:



Bài tập 24. Dùng **SmartArt** để tạo các mẫu sau





ĐẠI HỌC HOA KỲ

Bài tập 25. Sử dụng chức năng Mail Merge để điền số liệu từ bảng vào mẫu giấy chứng nhận sau

Sở Giáo dục và Đào tạo TP HCM Trường THỜI ĐẠI MỚI	CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM Độc Lập – Tự Do – Hạnh Phúc
<div style="border: 1px solid black; width: 80px; height: 60px; margin: 0 auto; display: flex; align-items: center; justify-content: center;"> Ảnh 3x4 </div>	<h2 style="margin: 0;">GIẤY CHỨNG NHẬN</h2>
Hiệu trưởng Trường THỜI ĐẠI MỚI chứng nhận:	
Họ tên:	Phái:
Sinh ngày:	Nơi sinh:
Đã hoàn tất chương trình về Windows NT Đạt loại:	
TP HCM, ngày tháng năm 1997 Hiệu Trưởng	

Họ và tên	Phái	Ngày sinh	Nơi sinh	Kết quả
Nguyễn Văn Bình	Nam	21/01/1978	Đồng Nai	Trung Bình
Trần Nguyên Hải	Nam	05/11/1974	An Giang	Khá
Nguyễn Thị Minh	Nữ	12/05/1976	TP HCM	Khá
Đặng Thị Thanh	Nữ	24/11/1973	TP HCM	Trung Bình
Nguyễn Văn Thắng	Nam	06/08/1969	Vũng Tàu	Giỏi

Bài tập 26. Sử dụng chức năng Mail Merge để trộn dữ liệu từ file **Danh sachungvien.mdb** vào văn bản sau



ĐẠI HỌC HOA SEN

R O D R I G U E Z , I N C .

2255 West 189th Street, New York, NY 11250

November 21st, 2002

(Per)
(Address)
(State), (Zip)

Dear *(Per), (Name)*

Thank you for applying for a position in our *(Career)* Department.

Although there are no openings as this time, we will keep your applications on file. Should a position become available within the next 12 months that requires someone with your background and qualifications, we will get in touch with you.

Once again, *(Per), (Name)*, thank you for your interest in our company. We wish you much success with your job search.

Sincerely

Marilyn Harnett
Employment Manager

ĐẠI HỌC HOA SEN



TIN HỌC ĐẠI CƯƠNG



PHẦN 1.

CÁC KHÁI NIỆM CĂN BẢN

1. THÔNG TIN

1.1. Khái niệm về thông tin

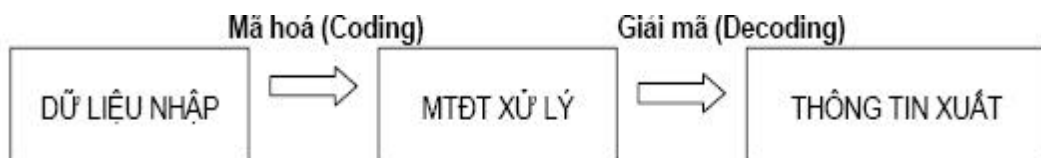
- **Dữ liệu (data)** là các sự kiện không có cấu trúc, không có ý nghĩa rõ ràng, cho đến khi chúng được xử lý theo một quá trình nào đó.

Ví dụ: Một sự kiện nào đó ghi lại trong nhận thức của một người, một mảnh giấy viết tay, một ký tự lưu trong bộ nhớ máy tính,....

- **Thông tin (Information)** là dữ liệu đã được xử lý xong, mang ý nghĩa rõ ràng.

Ví dụ: Văn bản (sách báo, công văn, thông báo,..), các loại số liệu (thống kê nhân sự, dự báo thời tiết,..), âm thanh, hình ảnh,...

- **Hệ thống thông tin (information system)** là tiến trình ghi nhận (nhập) dữ liệu, xử lý nó, và tạo nên (xuất) dữ liệu có ý nghĩa thông tin.



Thông tin có thể được thể hiện qua các thông báo, các biểu hiện vật chất hết sức đa dạng do tự nhiên và xã hội tạo ra theo qui tắc nào đó. Chúng ta có thể phân loại thông tin như sau: văn bản (sách, báo, truyện, thông báo, công văn,...), các loại số liệu (số liệu về thống kê nhân sự, dự báo thời tiết, dự báo kinh tế,...), âm thanh, hình ảnh,

1.2. Đơn vị đo thông tin

BIT (BInary digiT: Số nhị phân) là đơn vị cơ bản dùng để đo thông tin. Một bit có thể chứa hoặc là giá trị 0 hoặc là giá trị 1.

Một bit là giá trị nhỏ nhất của thông tin mà máy tính có thể sử dụng. Nhưng một bit thì quá nhỏ để có thể biểu diễn ký tự nên người ta thường sử dụng đơn vị byte (một nhóm 8 bit dùng để biểu diễn 1 ký tự) và các bội số của byte:

Tên gọi	Ký hiệu	Giá trị
Byte	B	8 bit
KiloByte	KB	2^{10} bytes = 1024 bytes
MegaByte	MB	2^{10} KB
GigaByte	GB	2^{10} MB
TetraByte	TB	2^{10} GB

1.3. Mã hóa thông tin rời rạc

Tất cả các thông tin ở dạng văn bản (text), chữ (character), số (number), ký hiệu (symbol), đồ họa (graphic), hình ảnh (image) hoặc âm thanh (sound) đều được gọi là các tín hiệu (signals). Tín hiệu có thể là liên tục hay rời rạc, hầu hết các dữ liệu mà chúng ta có được thường ở dạng tín hiệu rời rạc, khi đưa các tín hiệu này vào máy tính, chúng sẽ được mã hóa thành tín hiệu số (digital signal) nhằm giúp máy tính có thể hiểu và xử lý được.

2. XỬ LÝ THÔNG TIN

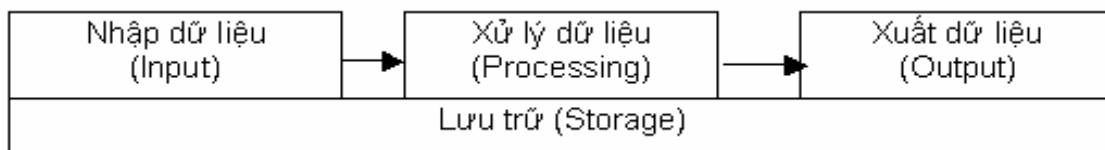
2.1. Sơ đồ tổng quát của một quá trình xử lý thông tin

Quá trình xử lý thông tin là quá trình biến đổi các dữ liệu thu thập được ở dạng rời rạc thành thông tin chuyên biệt phục vụ cho những mục đích nhất định. Mọi quá trình xử lý thông tin bằng máy tính hay bằng con người đều được thực hiện theo sơ đồ sau:

VÀO → XỬ LÝ → RA và LƯU TRỮ
(INPUT → PROCESSING → OUTPUT and STORAGE)

Quá trình này có thể được tóm tắt như sau:

Dữ liệu được nhập ở đầu vào, máy tính hay con người sẽ thực hiện quá trình xử lý, sau đó xuất dữ liệu ở đầu ra.



Quá trình nhập, xử lý và xuất dữ liệu đều cần được lưu trữ lại để dùng cho các nhu cầu về sau.

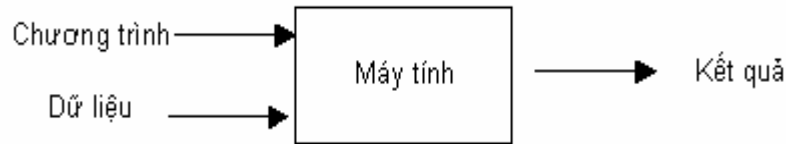
2.2. Xử lý thông tin bằng máy tính điện tử

Máy tính điện tử (MTĐT) hay máy tính là thiết bị điện tử cho phép lưu trữ (store) và xử lý (processing) thông tin một cách tự động theo chương trình (program) cho trước mà con người không cần can thiệp vào trong quá trình xử lý. Tức là con người phải cung cấp đầy đủ, ngay từ đầu cho máy tính các mệnh lệnh, chỉ thị nhằm hướng dẫn máy tính thực hiện các công việc cụ thể theo yêu cầu đề ra.

Các thành phần cơ bản của một MTĐT:

- Các thiết bị nhập như bàn phím, chuột, máy quét,... cung cấp dữ liệu cho máy tính.
- Dữ liệu và chương trình được chứa trong bộ nhớ, bộ xử lý trung tâm (CPU) thực hiện các thao tác xử lý và lưu tạm thời kết quả vào bộ nhớ trong.
- Các kết quả này được hiển thị thông qua các thiết bị xuất như màn hình, máy in,... hoặc được lưu trữ lại vào bộ nhớ ngoài.

Sơ đồ tổng quát xử lý dữ liệu trên máy tính:



Ưu điểm của việc xử lý thông tin bằng máy tính:

1. *Về phương diện lưu trữ*: máy tính có khả năng lưu trữ một lượng thông tin rất lớn trên một diện tích rất nhỏ.
2. *Về phương diện truy xuất*: máy tính có thể thực hiện các thao tác tìm kiếm, thêm bớt thông tin một cách dễ dàng tiện lợi.
3. *Về phương diện xử lý*: máy tính có tốc độ xử lý rất cao (hàng trăm triệu phép tính/1 giây) nhưng vẫn đảm bảo độ chính xác cao.
4. *Về phương diện ứng dụng*: với những ưu điểm trên cùng với sự phát triển như vũ bão của ngành công nghệ thông tin, ngày nay máy tính đã có mặt ở hầu hết các lĩnh vực.

2.3. Máy tính điện tử và lịch sử phát triển

Do nhu cầu giảm thời gian tính toán và tăng độ chính xác, con người đã chế tạo các công cụ tính toán từ xưa: bàn tính tay của người Trung Quốc, máy cộng cơ học của nhà toán học người Pháp Blaise Pascal (1623-1662), máy tính cơ học có thể cộng trừ nhân chia của nhà toán học người Đức Gottfried Wilhelm von Leibniz (1646-1716), máy sai phân để tính các đa thức toán học, máy phân giải điều khiển bằng máy đục lỗ của Charles Babbage (1792-1871),....

Tuy nhiên, máy tính điện tử thực sự hình thành bắt đầu vào thập niên 1950 và đến nay đã trải qua 5 thế hệ được phân loại theo sự tiến bộ về công nghệ điện tử và vi điện tử cũng như các cải tiến về nguyên lý, tính năng và loại hình của nó:

- *Thế hệ 1 (1950-1958)*: Máy tính sử dụng các bóng đèn điện tử chân không, mạch riêng rẽ, vào số liệu bằng phiếu đục lỗ, điều khiển bằng tay. Máy tính cồng kềnh, dễ hỏng, tốn hao nhiều năng lượng, tốc độ rất chậm (khoảng 300-3000 phép tính/1 giây), độ tin cậy thấp. Ví dụ: EDVAC (Mỹ) hay BESM (Liên xô cũ),...

- *Thế hệ 2 (1958-1964)*: Máy tính sử dụng các transistor. Máy đã có chương trình dịch như Cobol, Fortran và hệ điều hành đơn giản. Máy có kích thước ít cồng kềnh, bền hơn, ít hao năng lượng. Độ tin cậy cao, tốc độ được cải thiện hơn (có khả năng tính khoảng 10.000-100.000 phép tính/1 giây). Điển hình như loại IBM-1070 (Mỹ) hay EC (Liên xô cũ),...

- *Thế hệ 3 (1965-1974)*: Máy tính sử dụng các bộ xử lý bằng vi mạch điện tử cỡ nhỏ. Kích cỡ máy gọn, bền, ít hao năng lượng hơn hai thế hệ trước, tốc độ nhanh (hàng trăm nghìn phép tính/giây), độ tin cậy rất cao. Máy đã có các hệ điều hành đa chương

trình, nhiều người dùng đồng thời hoặc đa kiểu chia thời gian. Kết quả từ máy tính có thể in ra trực tiếp ở máy in. Điển hình như loại IBM 360 (Mỹ) hay MinSk (Liên xô cũ),...

- *Thế hệ 4 (1974 -1990)*: Máy tính sử dụng các vi mạch có độ tích hợp cao, đa xử lý có khả năng thực hiện hàng triệu phép tính/1 giây. Giai đoạn này hình thành 2 loại máy tính chính: máy tính cá nhân để bàn (Personal Computer - PC) hoặc xách tay (Laptop hay Notebook computer) và các loại máy tính chuyên nghiệp thực hiện đa chương trình, đa vi xử lý,... hình thành các hệ thống mạng máy tính (Computer Networks) và các ứng dụng phong phú, đa phương tiện.

- *Thế hệ 5 (bắt đầu từ 1990 đến nay)*: Các nhà sản xuất đã nghiên cứu chế tạo ra được các máy tính mô phỏng các hoạt động, hành vi của con người, có trí khôn nhân tạo với khả năng tự suy diễn phát triển các tình huống nhận được và giải quyết được các yêu cầu đa dạng.

Các loại máy tính điện tử:

Máy tính có rất nhiều loại, mỗi loại đáp ứng một mục đích cụ thể và dành cho các đối tượng người dùng khác nhau.

- Siêu máy tính (Super Computer).
- Máy tính lớn (Mainframe Computer).
- Máy tính mini (Mini Computer).
- Máy vi tính/máy tính cá nhân (Micro Computer/Personal Computer).
- Máy tính xách tay (Handle Computer/Laptop).

Trong đó chiếm số lượng nhiều nhất là *máy vi tính* vì nó phục vụ cho công việc hàng ngày của rất nhiều đối tượng người dùng.

3. TIN HỌC

3.1. Khái niệm

Tin học (informatics) được định nghĩa là ngành khoa học nghiên cứu các phương pháp, công nghệ, kỹ thuật lưu trữ và xử lý thông tin tự động. Công cụ chủ yếu của tin học là máy tính điện tử và các thiết bị truyền tin.

3.2. Các lĩnh vực nghiên cứu của tin học

Việc nghiên cứu chính của tin học tập trung chủ yếu vào 2 kỹ thuật ***phát triển song song*** nhau:

- **Kỹ thuật phần cứng (hardware engineering)**: nghiên cứu, chế tạo các thiết bị, linh kiện điện tử, công nghệ vật liệu mới,... hỗ trợ cho máy tính và mạng máy tính đầy mạnh khả năng xử lý toán học và truyền thông tin.

- **Kỹ thuật phần mềm (software engineering)**: nghiên cứu phát triển các phần mềm hệ điều hành, ngôn ngữ lập trình cho các bài toán khoa học kỹ thuật, mô phỏng điều khiển tự động, tổ chức dữ liệu và quản lý hệ thống thông tin.

3.3. Ứng dụng của tin học

Tin học hiện đang được ứng dụng ngày càng rộng rãi trong tất cả các lĩnh vực khác nhau của đời sống xã hội như: khoa học kỹ thuật, y học, kinh tế, công nghệ sản xuất, giáo dục, khoa học xã hội, giải trí,...

4. CÁC THÀNH PHẦN CỦA MỘT HỆ THỐNG MÁY TÍNH

Một hệ thống máy tính được chia làm hai phần: phần cứng (hardware) và phần mềm (software).

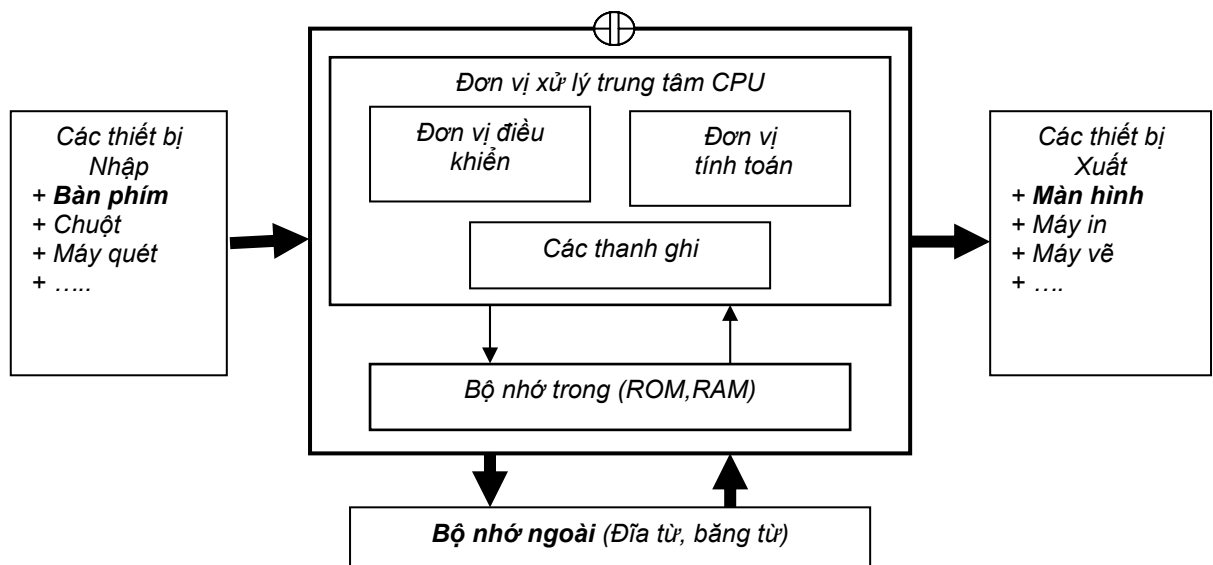


Hình 1.1. Các thành phần của một hệ thống máy tính

4.1. Phần cứng

Phần cứng có thể được hiểu đơn giản là bất kỳ thành phần nào trong một hệ thống máy tính mà chúng ta có thể thấy và sờ được. Đó chính là các thiết bị, các linh kiện điện tử. Phần cứng thực hiện các chức năng nhập, xuất, xử lý, và lưu trữ dữ liệu.

*** Sơ đồ cấu trúc phần cứng một hệ thống máy tính**



Hình 1.2. Sơ đồ cấu trúc phần cứng máy tính

4.1.1. Bộ nhớ

Bộ nhớ là thiết bị lưu trữ thông tin của máy tính. Khả năng lưu trữ thông tin của bộ nhớ gọi là dung lượng. Bộ nhớ được chia làm hai phần:

- **Bộ nhớ trong:** gồm có ROM và RAM

- **ROM (Read Only Memory)** là **bộ nhớ chỉ đọc**, dùng để lưu các chương trình hệ thống, chương trình điều khiển việc nhập xuất cơ sở (BIOS-Basic Input/Output System). Thông tin được lưu giữ trên ROM thường xuyên ngay cả khi mất điện. Bộ nhớ ROM được các công ty sản xuất máy tính cài đặt sẵn trên máy. *Người sử dụng máy tính không thể thay đổi thông tin trong ROM.*

- **RAM (Random Access Memory)** là **bộ nhớ truy xuất ngẫu nhiên**, được dùng để lưu trữ dữ kiện và chương trình trong quá trình thao tác và tính toán. Muốn thông tin trên RAM không bị mất thì phải luôn có nguồn nuôi để lưu trữ nội dung thông tin. Do đó nếu tắt máy hay ngắt điện thì thông tin trong RAM cũng bị mất đi. Dung lượng **RAM** cho các máy tính hiện nay thường là 128MB, 256MB, 1GB,....



Hình 1.3. Bộ nhớ RAM

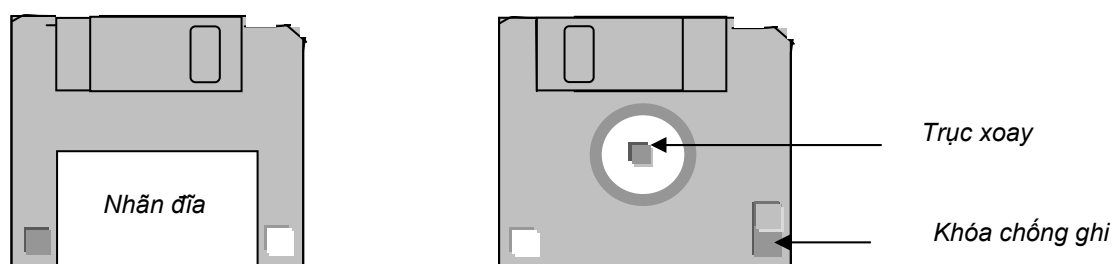
- **Bộ nhớ ngoài:** như đĩa từ, băng từ,.... để lưu trữ thông tin và có thể chuyển các thông tin này qua các máy tính khác, người ta sử dụng các đĩa, băng từ như là các bộ nhớ ngoài. Các bộ nhớ này có dung lượng chứa lớn, không bị mất đi khi không có nguồn điện. Trên các máy vi tính phổ biến hiện nay có các loại đĩa sau:

- **Đĩa cứng (Hard Disk):** có nhiều loại, dung lượng hiện nay từ 10GB, 20GB, 40GB, 80GB hoặc lên đến hàng trăm GB.



Hình 1.4. Ổ đĩa cứng

- **Đĩa mềm (Floppy Disk):** phổ biến là loại đĩa có đường kính 3.5 inches, dung lượng 1.44MB (thường được gọi là “đĩa 1.44MB”). Để sử dụng được đĩa mềm, cần phải có một ổ đĩa mềm (Floppy Drive) gắn trong máy tính.



Hình 1.5. Mặt trước và sau của đĩa 1.44MB

- **Đĩa quang** (CD: Compact Disk): việc đọc ghi dữ liệu đối với loại đĩa này được thực hiện dựa trên công nghệ quang học (sử dụng công nghệ tia laser để đọc và ghi dữ liệu).



Hình 1.6. Ổ đĩa quang và đĩa quang

Hiện nay có các loại đĩa quang sau:

- + **Đĩa CD-ROM** (Compact Disk Read Only Memory): là loại đĩa chỉ đọc.
- + **Đĩa CD-R** (Compact Disk Recordable): là loại đĩa CD trắng (chưa có dữ liệu) cho phép ghi dữ liệu duy nhất một lần.
- + **Đĩa CD-RW** (Compact Disk ReWritable): loại này cho phép đọc ghi nhiều lần (giống như đĩa cứng, đĩa mềm).
- + **Đĩa DVD** (Digital Versatile Disk hoặc Digital Video Disk): là loại đĩa có khả năng lưu trữ lớn, thường được sử dụng để lưu các đoạn phim.
- + **Đĩa flash USB**: là thiết bị giúp sao lưu nhanh gọn nhất bởi tính năng ưu việt của nó trong việc truyền tải dữ liệu. Các loại đĩa flash giao tiếp với máy tính thông qua cổng USB.

Để thực hiện việc đọc/ghi dữ liệu trên các loại đĩa quang, máy tính cần phải có các loại ổ đĩa quang thích hợp cho từng với loại đĩa.

Về dung lượng, các đĩa CDROM, CD-R, CD-RW có thể chứa khoảng 650-700 MB dữ liệu, riêng đĩa DVD có thể lưu trữ từ 4,7-17 GB dữ liệu tùy thuộc vào kỹ thuật ghi và đọc dữ liệu.

4.1.2. Bộ xử lý trung tâm (CPU-Central Processing Unit)

Bộ xử lý trung tâm có nhiệm vụ điều khiển toàn bộ các hoạt động của máy tính và thực hiện các phép tính. CPU có 3 bộ phận chính:

- **Bộ điều khiển** (CU: Control Unit) là trung tâm điều hành máy tính, nó có nhiệm vụ giải mã các lệnh của chương trình và điều khiển các hoạt động xử lý.

- **Bộ tính toán số học và logic** (ALU: Arithmetic-logic Unit) bao gồm các thiết bị thực hiện các phép toán số học (cộng, trừ, nhân, chia,...), các phép tính logic (AND, OR, NOT, XOR) và các phép tính quan hệ (so sánh lớn hơn, nhỏ hơn, bằng nhau,...)

- **Các thanh ghi** (registers) được gắn chặt vào CPU bằng các mạch điện tử, được sử dụng làm bộ nhớ trung gian. Các thanh ghi có chức năng giúp tăng tốc độ trao đổi thông tin trong máy tính.

Ngoài ra CPU còn được gắn với một **đồng hồ** (clock) hay còn gọi là bộ tạo xung nhịp. Tần số đồng hồ càng cao thì tốc độ xử lý thông tin càng nhanh. Thường thì đồng hồ được gắn tương xứng với cấu hình máy và có các tần số dao động là khoảng từ 33 MHz đến vài GHz.

4.1.3. Các thiết bị nhập/xuất (input/output)

Các thiết bị nhập xuất bao gồm các thiết bị nhập (input device), thiết bị xuất (output device) và thiết bị truyền tin. Chúng có tác dụng chuyển dữ liệu (thông tin) từ bên ngoài vào máy tính và ngược lại.

* Các thiết bị nhập chính:

- **Bàn phím (keyboard)**: là thiết bị nhập dữ liệu và câu lệnh, và được coi là *thiết bị nhập thông tin chuẩn*. Bàn phím máy vi tính phổ biến hiện nay có khoảng 101-106 phím, các phím có tác dụng khác nhau, có thể chia làm các nhóm chính như sau:

+ *Nhóm phím dữ liệu (data key)*: gồm các phím chữ số, các phím ký tự: a → z, A → Z, 0 → 9,...

+ *Nhóm phím chức năng (function key)*: gồm các phím từ F1 → F12.

+ *Nhóm phím trạng thái và đệm số*: gồm các phím như NumLock, CapsLock, ScrollLock, Enter, Esc (Escape), Shift, Alt (Alternate), Ctrl (Control),...

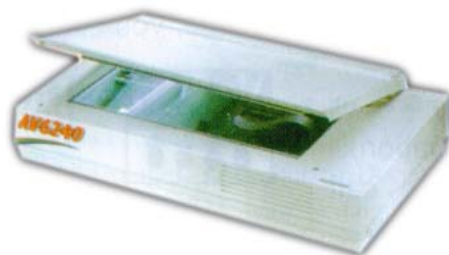
+ Ngoài ra còn có một số phím đặc biệt hỗ trợ cho hệ điều hành Windows và các thiết bị phần cứng

☞ *Lưu ý*: Các phím Shift, Alt, Ctrl thường không sử dụng riêng lẻ mà chúng thường phối hợp với các phím khác tạo thành một *tổ hợp phím* có tác dụng như một phím mới.

- **Chuột (Mouse)**: là thiết bị cần thiết phổ biến hiện nay, nhất là các máy tính chạy trong môi trường Windows. Chuột có kích thước vừa nắm tay di chuyển trên một tấm phẳng (mouse pad) theo hướng nào thì dấu nháy hoặc mũi tên trên màn hình sẽ di chuyển theo hướng đó tương ứng với vị trí của viên bi hoặc tia sáng (optical mouse) nằm dưới bụng của nó. Một số máy tính có chuột gắn trên bàn phím.

- **Máy quét (Scanner)**: là thiết bị dùng để quét văn bản, hình vẽ, ảnh chụp vào máy tính. Thông tin nguyên thủy trên giấy sẽ được quét thành các tín hiệu số tạo thành các tập

tin ảnh (image file). Scanner đi kèm với phần mềm để nhận diện các tập tin ảnh hoặc văn bản.



Hình 1.7. Máy quét

*** Các thiết bị xuất thông tin chính**

- **Màn hình (Screen hay Monitor):** là *thiết bị xuất chuẩn*, dùng để thể hiện thông tin cho người sử dụng xem. Thông tin được thể hiện ra màn hình bằng phương pháp ánh xạ bộ nhớ (memory mapping), với cách này màn hình chỉ việc đọc dữ liệu liên tục từ trong bộ nhớ và hiển thị (display) bất kỳ thông tin nào hiện có trong vùng nhớ ra màn hình. Vì vậy, để xuất thông tin ra màn hình ta chỉ cần xuất ra vùng nhớ tương ứng.

Trong chế độ văn bản, màn hình thể hiện 80 cột ký tự (đánh số từ 0 - 79) và 25 dòng (đánh số từ 0 - 24).

Trong chế độ đồ họa, màn hình được chia thành các phần tử ảnh (pixels). Tích số này càng lớn thì màn hình càng mịn và rõ nét.

Loại màn hình màu	Độ phân giải (resolution) cao nhất
CCA: Color Graphic Adapter	320 x 200 (pixel)
EGA: Enhanced Graphic Adapter	640 x 350
VGA: Video Graphic Array	640 x 480
SVGA: Super VGA	1024 x 768

- **Máy in (Printer):** là thiết bị xuất, dùng để đưa thông tin ra giấy. Máy in phổ biến hiện nay là máy in ma trận điểm (dot matrix) loại 9 kim và 24 kim, máy in phun mực, máy in laser trắng đen hoặc màu.



Hình 1.8. Các loại máy in

Giấy in thường dùng là loại giấy in 80 cột (in được 80 ký tự, in nén được 132 ký tự) và loại giấy in khổ rộng in được 132 cột (in 132 ký tự, in nén được 256 ký tự). Cả hai loại giấy in đều có khả năng in 66 dòng/trang.

✓ Các thiết bị nhớ ngoài như : đĩa từ, băng từ,... (xem phần bộ nhớ).

4.2. Phần mềm (software)

Phần mềm là những chương trình làm cho phần cứng của máy tính hoạt động được. Thông thường, phần mềm chia làm 2 loại chính như sau:

❖ **Hệ điều hành (OS: Operating System):** là phần mềm cơ bản, gồm tập hợp các chương trình điều khiển hoạt động của máy tính cho phép người dùng sử dụng khai thác dễ dàng và hiệu quả các thiết bị của hệ thống. Một số hệ điều hành thông dụng: MS-DOS, Windows, Unix, OS/2, Linux,...

❖ **Phần mềm ứng dụng (Application):** là các chương trình ứng dụng cụ thể vào một lĩnh vực.

- Phần mềm soạn thảo văn bản (Wordprocessing): Microsoft Word, EditPlus,...

- Phần mềm quản trị dữ liệu (Database Management System): Visual Foxpro, Access, SQL Server,...

- Phần mềm đồ họa: Corel Draw, PhotoShop, FreeHand, Illustrator,...

- Phần mềm thiết kế: AutoCad cho ngành xây dựng, cơ khí, Orcad cho ngành điện tử viễn thông.

- Phần mềm chế bản điện tử: PageMaker, QuarkPress,...

- Phần mềm thiết kế trang Web: FrontPage, DreamWeaver,...

- Ngôn ngữ lập trình (Programming Language): dùng xây dựng các phần mềm ứng dụng. Một số ngôn ngữ lập trình: C, Pascal, C++, Visual Basic, Visual C++, Delphi, Java,...

5. MẠNG MÁY TÍNH

5.1. Định nghĩa

Mạng máy tính là hệ thống liên kết hai hoặc nhiều máy tính lại với nhau. Một mạng máy tính thông thường gồm nhiều máy tính, gọi là các máy khách, được kết nối tới một máy tính chính gọi là máy chủ. Máy chủ cung cấp cho các máy khách không gian lưu trữ, chương trình, các dịch vụ gửi nhận thư...

5.2. Ưu điểm của mạng máy tính

- Một số người sử dụng trong mạng không cần phải trang bị máy tính đắt tiền.

- Mạng máy tính cho phép người lập trình ở một trung tâm máy tính này có thể sử dụng các chương trình tiện ích của một trung tâm máy tính khác đang rỗi, sẽ làm tăng hiệu quả kinh tế của hệ thống.

- Rất an toàn cho dữ liệu và phần mềm vì phần mềm mạng sẽ khoá các tệp tin (*files*) khi có những người không đủ quyền hạn truy xuất các tệp tin và thư mục đó.

5.3. Cấu trúc mạng theo phạm vi địa lý

Các máy khách có thể được kết nối đến máy chủ bằng cáp, đường điện thoại hoặc vệ tinh,...

- Một mạng kết nối các máy tính trong một vùng địa lý nhỏ, ví dụ như trong một tòa nhà hay các tòa nhà trong một thành phố, được gọi là mạng cục bộ (LAN : Local Area Network).

- Một mạng kết nối các máy tính trong một vùng địa lý rộng, ví dụ như giữa các thành phố, được gọi là mạng diện rộng (WAN : Wide Area Network).

- Mạng Internet là một mạng máy tính toàn cầu. Trong đó, các máy tính kết nối với nhau thông qua tập chuẩn chung các giao thức gọi là TCP/IP (Transmission Control Protocol/Internet Protocol). Không có máy tính nào làm chủ và điều khiển tất cả.

- Một Intranet là một mạng cục bộ nhưng dùng giao thức TCP/IP để kết nối với các máy trong mạng. Một Intranet của một công ty có thể được kết nối với các Intranet của các công ty khác và kết nối vào Internet.

PHẦN 2.

HỆ ĐIỀU HÀNH MICROSOFT WINDOWS

Chương 1.

HỆ ĐIỀU HÀNH WINDOWS XP

1.1. GIỚI THIỆU

Hệ điều hành Windows do hãng Microsoft sản xuất và được xem là hệ điều hành (viết tắt HĐH) thông dụng nhất hiện nay. HĐH này có rất nhiều phiên bản khác nhau, mỗi phiên bản thường đòi hỏi các *cấu hình phần cứng* tương ứng (phiên bản càng mới đòi hỏi cấu hình phần cứng càng cao). Các phiên bản phổ biến ở nước ta hiện nay là các bản *Windows 98/98SE/ME, Windows 2000/NT, Windows XP, Windows 2003....*

Sơ lược về lịch sử của HĐH Windows:

- Tháng 11/1985, Windows 1.0 ra đời và bị đánh giá kém, giao diện xấu không đa nhiệm.

- Tháng 05/1990, Windows 3.0 xuất hiện với giao diện cải tiến, đẹp và đa nhiệm, do đó được đánh giá tốt hơn và bắt đầu được phổ biến.

- 1991-1993, Windows 3.1 và Windows for Workgroups 3.11 cải tiến dựa trên phiên bản 3.0

- Tháng 09/1995, Windows 95 ra đời đã đánh dấu một bước tiến lớn về giao diện đồ họa cho người sử dụng (GUI: Graphical User Interface), và đặc biệt là không còn chạy trên hệ điều hành MS-DOS như các phiên bản trước. Đây là một *hệ điều hành thật sự*.

- Từ năm 1998 đến nay, hệ điều hành Windows đã có các bản 98/98SE/ME/NT/2000/XP/2003 và phiên bản mới nhất hiện nay là Windows Vista.

Ngoài ra còn có bản Windows 97, nhưng đây thật sự là bản 95 có kèm thêm trình duyệt Web *Internet Explorer*.

Từ các phiên bản Windows 3.x trở về trước đều phải khởi động từ hệ điều hành DOS, còn các phiên bản từ Windows 95 trở đi thì *chạy độc lập* và MS-DOS được xem như một chương trình ứng dụng chạy trên HĐH Windows.

Các đặc điểm của HĐH Windows:

- Giao diện đồ họa cho người sử dụng (GUI: Graphical User Interface): Windows có giao diện đồ họa thân thiện và dễ sử dụng, giao diện tiếp xúc người-máy ở HĐH Windows là giao diện đồ họa, thông qua các hình ảnh được gọi là các **biểu tượng (Icon)**. Điều này đã làm người sử dụng có thể dễ dàng nhận biết các đối tượng qua các biểu tượng của nó.

- Giao diện chương trình ứng dụng (Application Program Interface – API): đặc điểm này cho phép hầu hết các giao diện tiếp xúc người sử dụng–máy tính ở tất cả các chương trình ứng dụng đều tương tự nhau (ví dụ: như các cửa sổ, các nút lệnh, các tùy chọn ...). Điều đó giúp người sử dụng dễ học, dễ thao tác khi tiếp xúc với một ứng dụng mới.

- Khả năng đa nhiệm (Multitasking): Windows cho phép cùng lúc thi hành nhiều chương trình ở từng cửa sổ riêng biệt.


- Là *HHH 32 bit* và chạy nhanh hơn do thành phần chính hoạt động theo chế độ 32 bit này.

- Cho phép tên tập tin/thư mục có chiều dài tối đa đến 255 ký tự, được phép có các ký tự trắng, nhưng không được chứa bất kỳ ký tự đặc biệt nào sau đây: \ / : * ? ' " < > |.

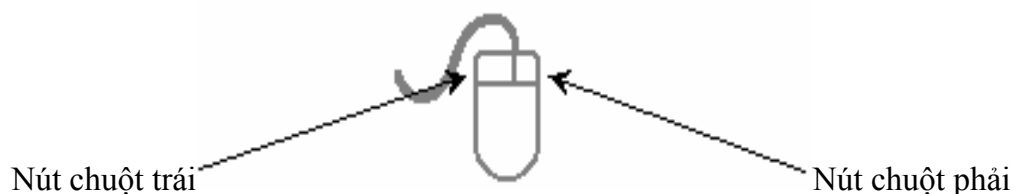
- Hỗ trợ chế độ cho nhiều người dùng chung trên một máy, về mạng, Internet.

- Ngoài ra còn nhiều tính năng khác như: đồ họa, âm nhạc, phim ...

Sử dụng chuột

Trước khi làm việc với Windows, bạn cần phải biết sử dụng thiết bị chuột (*bàn phím và chuột là hai thiết bị nhập chuẩn của Windows*). Với bàn phím bạn có con trỏ của bàn phím (giống như con trỏ của DOS), với chuột bạn cũng có *con trỏ chuột*. Con trỏ chuột thường có hình dạng như một mũi tên  (có thể có hình dạng khác), bạn sẽ điều khiển con trỏ chuột này bằng các thao tác với thiết bị chuột bên ngoài.

Các thao tác cơ bản với chuột (Mouse):



Thông thường một thiết bị chuột thường có 2 nút chính: **trái** – **phải** (*left - right*). Do nút chuột trái được sử dụng thường xuyên hơn, nên khi nói thao tác *nhấp chuột* được ngầm hiểu là *nhấp chuột trái*.

Các thao tác với thiết bị chuột:

- *Di chuyển (Move) trỏ chuột*: Cầm thiết bị chuột rồi di chuyển cho nó trượt trên mặt phẳng, khi đó trỏ chuột trên màn hình cũng di chuyển theo.

- *Nhấp chuột (Click)*: Bấm và nhả nhanh nút chuột trái.

- *Nhấp đúp chuột (Double click)*: nhấp chuột trái hai lần.

- *Nhấp phải chuột (Right click)*: nhấp chuột phải 1 lần.

- *Rê chuột (Drag)*: Bấm và giữ nút trái rồi di chuyển đến vị trí mới, sau đó mới nhả nút trái. Thao tác này còn được gọi là *nắm kéo*.

1.2. KHỞI ĐỘNG VÀ THOÁT KHỎI WINDOWS XP

Muốn sử dụng HĐH Windows cần phải cài đặt nó. HĐH Windows có rất nhiều tập tin và cần nhiều chỗ trống trên đĩa tùy thuộc vào phiên bản đang sử dụng ,do đó HĐH Windows phải được cài trên ổ cứng. Các tập tin/thư mục chính của hệ điều hành Windows thường được lưu trong thư mục C:\WINDOWS.

1.2.1. Khởi động

Bật nút **Power** ở thùng máy để mở nguồn điện. Máy tính sẽ khởi động, kiểm tra các thiết bị phần cứng, sau đó nạp HĐH vào bộ nhớ RAM. Quá trình khởi động sẽ kết thúc khi màn hình tương tự sau xuất hiện.



Hình 2. 1. Màn hình Windows XP

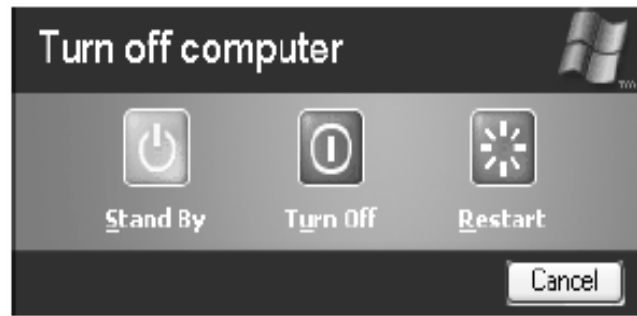
⚠ Lưu ý:

- Mỗi máy sẽ có màn hình hiển thị khác nhau tùy theo ý thích của người sử dụng. Do đó bạn đừng ngạc nhiên khi màn hình của bạn khác với một số hình minh họa trong giáo trình này.

- Hãy quan sát màn hình logo khởi động của Windows, bạn sẽ biết được máy của mình đang sử dụng là phiên bản nào (nhìn vào dòng chữ Microsoft Windows 95 hay 98, ME, NT, 2000 ,XP ...).

1.2.2. Thoát khỏi Windows và tắt máy

Click nút **Start**, click chọn mục **Turn Off Computer**. Trong hộp thoại Turn off computer xuất hiện, click nút **Turn off**.



Hình 2.2. Hộp thoại Turn off computer.

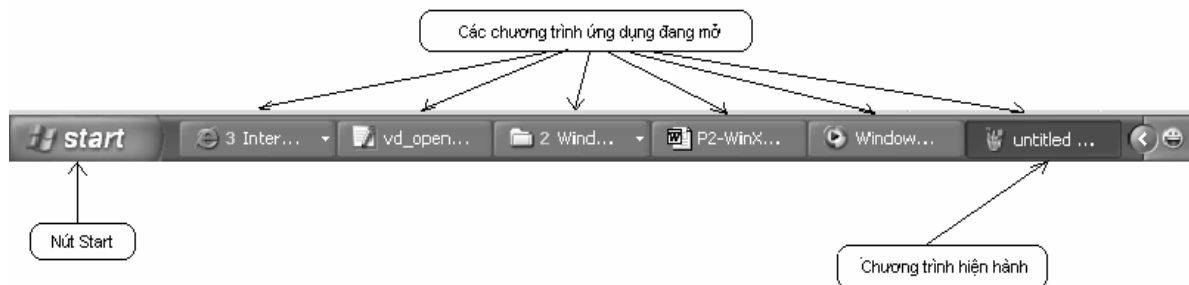
Chú ý: Trước khi thoát khỏi Windows để tắt máy tính, bạn nên thoát khỏi các ứng dụng đang chạy sau đó thoát khỏi Windows. Nếu tắt máy ngang có thể gây ra những lỗi nghiêm trọng.

1.3. CÁC THÀNH PHẦN VÀ CÁCH TỔ CHỨC TRONG WINDOWS XP

1.3.1. Các thành phần cơ bản trong Windows XP

➤ Thanh tác vụ - Taskbar


Thường nằm ở phía dưới của màn hình (có thể ở phía trên, bên trái hoặc bên phải màn hình). Đây là thanh tác vụ chứa menu chính của Windows (*menu Start*) và cũng là thanh trạng thái cho biết những trình ứng dụng nào đang được chạy, và ứng dụng nào là hiện hành.



Hình 2.3. Thanh tác vụ

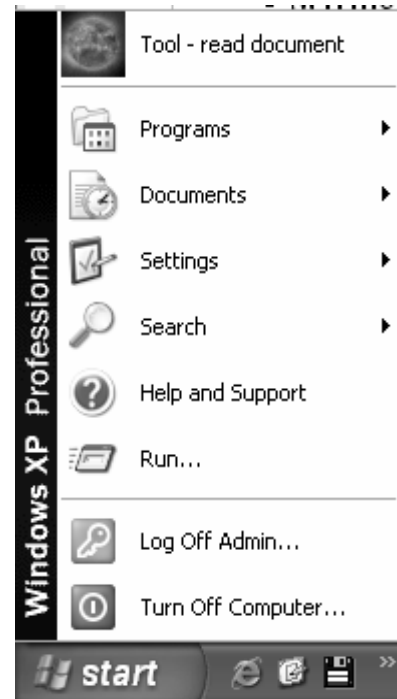
➤ Nút Start và menu Start.

Nút Start là một thành phần trên thanh Taskbar. Bạn có thể làm tất cả các công việc từ nút này. Click chuột vào nút Start bạn sẽ thấy một hệ thống menu phân cấp, các menu con sẽ tự động mở ra khi bạn lướt chuột đến (không nhấn giữ bất cứ nút chuột nào), menu này được gọi là **menu Start**. Bạn chỉ Click chuột khi muốn chọn một lệnh hoặc muốn khởi động một chương trình nào đó.

Bạn có thể truy cập nhanh đến nút Start bằng tổ hợp phím *Ctrl+Esc* hoặc phím *Windows* (phím có biểu tượng ).

* **Các thành phần trong menu Start:**

- **Programs:** hiển thị danh sách tên và nhóm các chương trình ứng dụng mà bạn có thể khởi động.
- **Documents:** hiển thị danh sách các tập tin mà bạn đã mở trong thời gian gần đây nhất.
- **Settings:** chứa danh sách các thành phần, các thiết bị mà bạn có thể thay đổi các thông số thiết định.
- **Search:** cho phép tìm các thư mục, tập tin và kể cả các máy tính khác trong mạng nếu bạn đang là thành viên của mạng đó. Ngoài ra bạn cũng có thể tìm các dữ liệu cụ thể trong các file.
- **Help and Support:** lệnh cho phép hiển thị chương trình giúp đỡ của Windows.
- **Run... :** chạy một chương trình hay khởi động một ứng dụng.
- **Turn off...:** khởi động lại Windows hoặc tắt máy.



Hình 2.4. Menu Start

🔗 **Lưu ý:**

- Các mục có dấu mũi tên ▶ phía sau có nghĩa là còn có thêm một menu con nữa. Nếu ta di chuyển chuột đến các mục này menu con của nó sẽ tự hiện ra.
- Dấu ... (ba dấu chấm) phía sau lệnh có nghĩa là khi chọn lệnh này sẽ xuất hiện một hộp thoại.
- Các mục không có dấu ▶ hoặc dấu ... thì tương ứng với một chương trình sẽ được mở nếu chọn nó.

Desktop

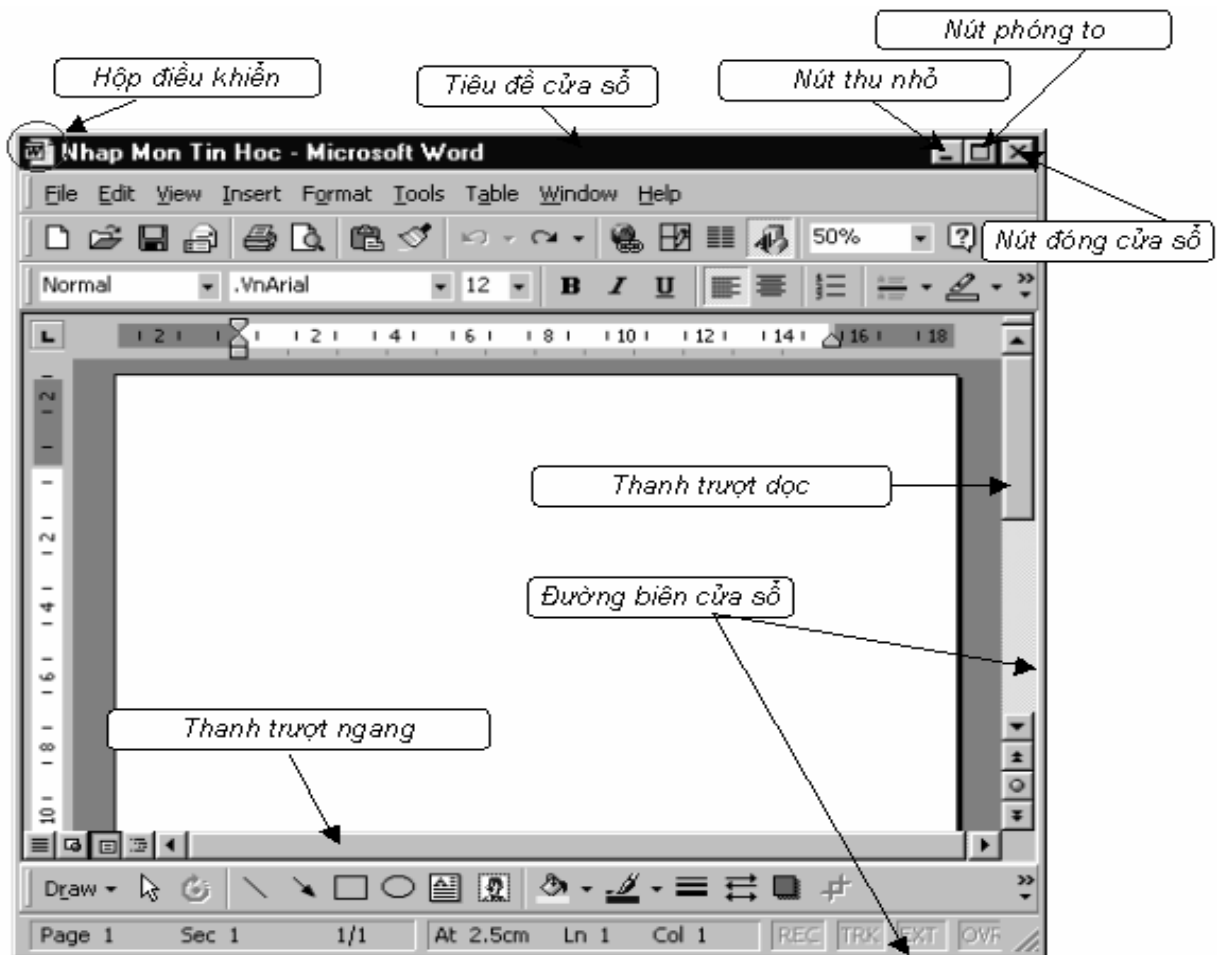
Đây là màn hình nền của Windows, bao gồm tất cả những gì bạn thấy trên màn hình trừ *Taskbar* (hoặc thanh *Microsoft Office Shortcut Bar* nếu có).

Desktop có công dụng như *bàn làm việc* của bạn. Bạn có thể thiết kế và trang trí cho nó theo ý muốn của mình.

Cửa sổ và các thành phần của nó:





Các chương trình hoặc các ứng dụng khi được chạy trên Windows sẽ xuất hiện dưới dạng một cửa sổ (window), tên chương trình sẽ được hiện trên thanh tiêu đề của cửa sổ.

Cửa sổ là một vùng màn hình, dạng chữ nhật. Với cửa sổ ta các thẻ dễ dàng đóng, mở, di chuyển hoặc thay đổi kích thước của nó. Ở một thời điểm, ta có thể mở nhiều cửa sổ một lúc (mở nhiều chương trình ứng dụng), nhưng trong các cửa sổ được mở, chỉ có một cửa sổ hiện hành (cửa sổ nằm trên cùng), các cửa sổ còn lại sẽ bị che khuất.



Hình 2.5. Cửa sổ trong Windows

* Chức năng của các thành phần trên một cửa sổ:

- **Control menu**  (tùy theo ứng dụng nó sẽ có hình biểu tượng của ứng dụng): trình đơn điều khiển, nó chứa các lệnh điều khiển cửa sổ như các nút ở trên.
- **Title bar**: thanh tiêu đề hiển thị tiêu đề của cửa sổ. Bạn có thể di chuyển cửa sổ một cách dễ dàng bằng cách nắm kéo thanh tiêu đề này.
- **Minimize button**  : thu nhỏ cửa sổ, khi nhấn vào nút này cửa sổ sẽ được thu nhỏ lại dưới dạng hộp biểu tượng trên thanh Taskbar.
- **Maximize button**  : phóng to cửa sổ, khi nhấn vào nút này cửa sổ sẽ được mở rộng ra hết màn hình.
- **Restore button**  : phục hồi lại kích thước của cửa sổ theo ý thích của người sử dụng.
- **Close button**  : Đóng cửa sổ, bạn có thể dùng nó để thoát khỏi ứng dụng.
- **Window border** : Đường biên của cửa sổ, nếu bạn đưa trỏ chuột đến các biên này bạn sẽ thấy trỏ chuột có hình mũi tên hai chiều (\updownarrow , \leftrightarrow , \swarrow , hoặc \searrow) và bạn có thể thay đổi kích thước của cửa sổ theo chiều của các mũi tên.

▪ **Horizontal scroll bar** và **Vertical scroll bar**: Thanh trượt ngang và thanh trượt dọc. Chúng được dùng trong trường hợp dữ liệu vượt quá kích thước của cửa sổ.

➤ **Dialog box.**

Khi làm việc với Windows, chúng ta thường xuyên gặp một dạng cửa sổ đặc thù được gọi là hộp hội thoại (dialog box), hay còn gọi là *hộp thoại*. Hộp thoại có chức năng cung cấp thông tin, yêu cầu thông tin hoặc đòi hỏi người sử dụng xác nhận các lựa chọn. Hộp thoại có 3 dạng chính như sau:

• **Dạng 1.**

- Là một cửa sổ được tổ chức theo nhiều trang (*có thể chỉ có 1 trang*) hay còn gọi là *thẻ (Tab page)*; mỗi thẻ có nội dung khác nhau.

- Ở một thời điểm, chỉ có một thẻ hiện hành tức là thẻ đang được chọn (thẻ nằm trên cùng). Nếu ta chọn thẻ khác, lập tức nội dung của thẻ mới được chọn sẽ xuất hiện.

Ví dụ: Trong hộp thoại AutoCorrect (hình 2.6) có bốn thẻ: AutoCorrect, AutoFormat As You Type, AutoText, AutoFormat. Trong đó, thẻ hiện hành là thẻ AutoCorrect.

- Trên một thẻ thường có các thành phần sau:

▪ **Button**: thường có dạng nút nhấn hình chữ nhật, các từ ở giữa chính là tên nút, nó mô tả công việc sẽ thi hành nếu nút được nhấn vào.

▪ **Check box**: có dạng ô vuông nhỏ, đi theo sau là một đoạn văn bản mô tả công việc sẽ được thi hành nếu được đánh dấu (check).

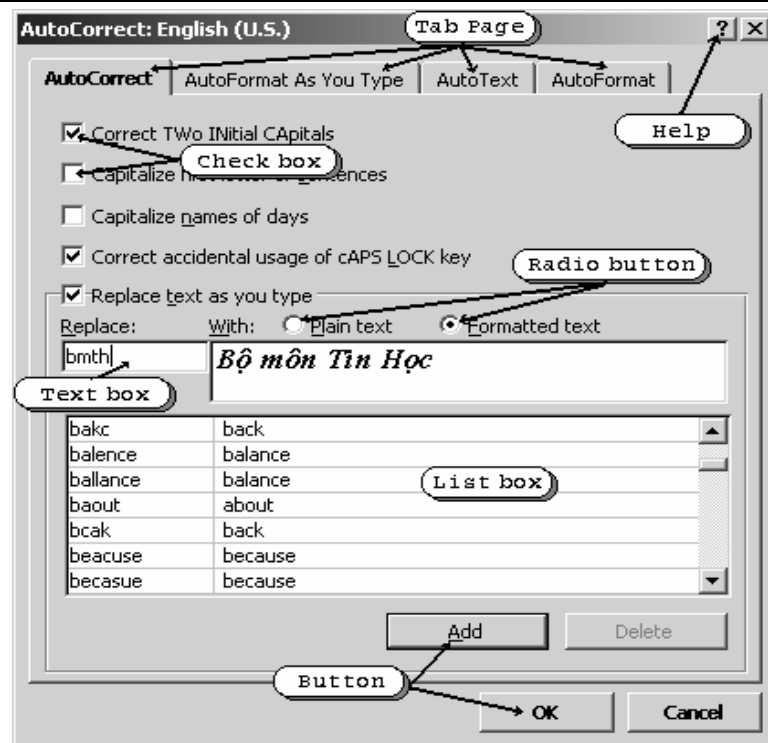
▪ **Radio button**: có dạng ô tròn nhỏ. Giống check box, Radio button cũng có một đoạn văn bản mô tả công việc thực hiện nếu được chọn, nhưng chúng thường đi theo từng nhóm (từ hai nút trở lên). Ta chỉ có thể chọn duy nhất một “thành viên” trong mỗi nhóm.

▪ **List box**: loại này có nhiều dạng, nhưng chúng có đặc điểm chung là *các mục chọn của chúng được hiển thị dưới dạng danh sách*. Muốn chọn mục nào ta di chuyển thanh sáng đến mục đó.

▪ **Text box**: có dạng ô chữ nhật, đây là nơi người dùng có thể nhập vào một dòng văn bản hoặc một dòng lệnh.

- Trong hộp thoại dạng này, luôn luôn có *nút OK* và *nút Cancel* phía dưới. Nhấn nút OK (hoặc *phím Enter*) nếu đồng ý với những mục đã chọn; nhấn nút Cancel (hoặc *phím ESC*) nếu muốn hủy bỏ những mục vừa chọn (như lúc chưa được chọn).

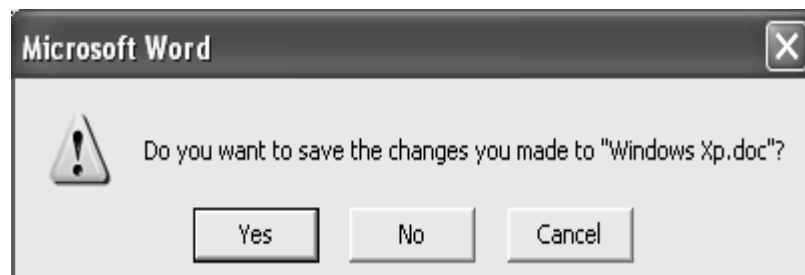
- Ta có thể dùng chuột hay phím Spacebar để *đánh dấu/bỏ đánh dấu* hay chọn/bỏ chọn.



Hình 2.6. Hộp thoại dạng 1

- **Dạng 2:**

- Đây là dạng thông báo, hoặc cảnh báo cho người sử dụng về công việc đang thực hiện và yêu cầu xác nhận của người dùng qua các nút lệnh.



Hình 2.7. Hộp thoại dạng 2 (thông báo).

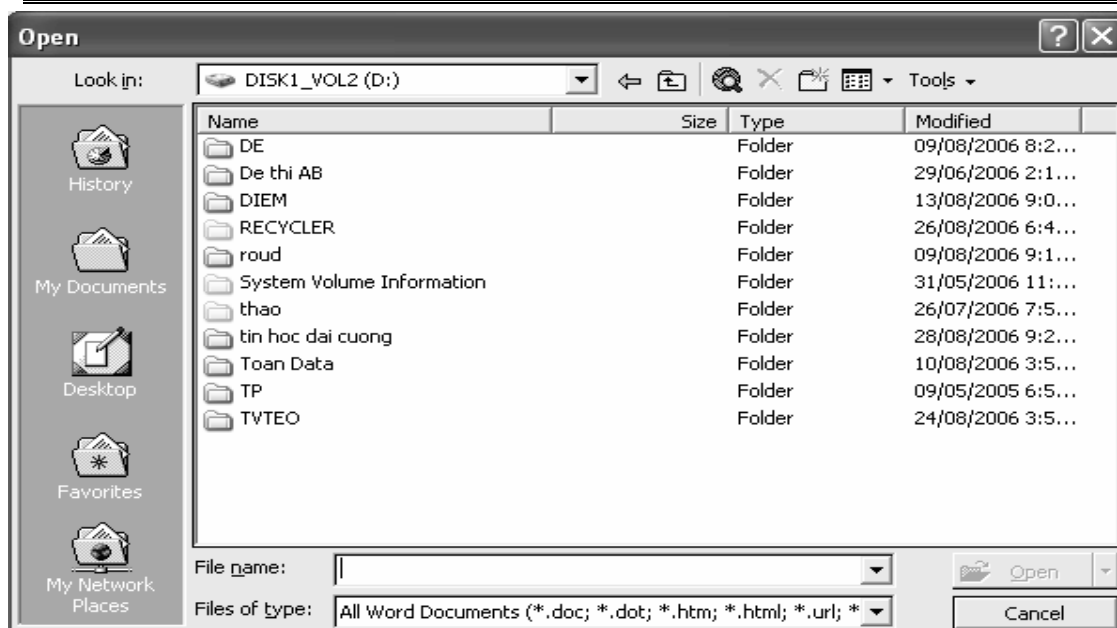
- Dạng này thường có các nút lệnh: nút OK, nút Cancel, nút Yes, nút No, ...

+ *Nút OK* và *nút Cancel* thường xuất hiện trong dạng thông báo.

+ *Nút Yes* và *nút No* luôn đi chung với nhau, thường xuất hiện dưới dạng cảnh báo qua câu hỏi.

- **Dạng 3:**

- Đây hộp thoại được sử dụng để truy xuất đến một hoặc nhiều đối tượng (thường là các tập tin).



Hình 2.8. Hộp thoại dạng 3

- Các thành phần chính trong dạng hộp thoại này:

- *Look in:* chứa danh sách các ổ đĩa và cây thư mục hiện hành. Ta có thể chọn ổ đĩa từ đây.

- *Bảng Contents:* chứa nội dung của thư mục hiện hành (thư mục được hiển thị trong *Look in*). Đối với bảng này nếu ta *double click* hoặc *nhấn phím Enter* vào một thư mục, có nghĩa là chọn thư mục này làm thư mục hiện hành; nếu *double click* hoặc *nhấn phím Enter* vào tập tin có nghĩa là sẽ *truy xuất đến nó* - mở (open), chạy (run), hoặc lưu (save),...

- *File name:* ta có thể truy xuất nhanh đến đối tượng bằng cách gõ *đường dẫn và tên đối tượng* vào text box này.

- *File of type:* chỉ định loại tập tin được hiển thị ở bảng Contents.

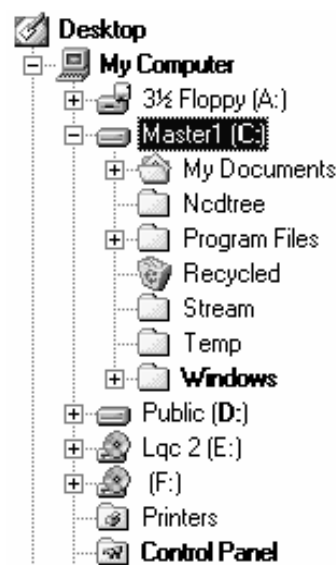
1.3.2. Cách tổ chức các thành phần trong Windows

Các thành phần trong Windows được tổ chức theo dạng cây phân cấp, có cấu trúc chặt chẽ.

Ví dụ: Xem hình bên, ta sẽ thấy thư mục **Windows** được lưu trong thư mục gốc của ổ đĩa **C:**, các ổ đĩa được quản lý bởi **My Computer**. Tất cả đều được tổ chức theo dạng cây phân cấp.

Những đối tượng có dấu cộng (+) phía trước có nghĩa là bên trong nó còn các thành phần con, và các thành phần con này cũng được tổ chức theo dạng cây phân cấp.

Ngoài ra bạn nên xem lại Taskbar và Start menu để hiểu rõ hơn về cách quản lý của Windows.



Hình 2.9. Cấu trúc cây phân cấp

1.4. KHỞI ĐỘNG VÀ THOÁT MỘT CHƯƠNG TRÌNH ỨNG DỤNG

1.4.1. Khởi động một ứng dụng trên Windows

Có 3 cách chính sau:

↳ **Cách 1:** Nếu biết rõ đường dẫn và tên tập tin chương trình của ứng dụng cần mở, thực hiện như sau:

① Click nút *Start*, chọn lệnh *Run*.

② Gõ vào đường dẫn và tên chương trình cần chạy và nhấn *Enter*, hoặc nhấn vào nút *Browse* để tìm đến tập tin chương trình cần chạy.


↳ **Cách 2:** Khởi động từ menu Programs

① Click nút *Start*, chuyển chuột đến menu *Programs*.

② Chuyển chuột đến tên của ứng dụng cần mở và **Click chuột** vào đó (nếu không thấy trong menu Programs bạn hãy tiếp tục tìm trong các menu con).

Ví dụ:

– Khởi động trình soạn thảo văn bản **Word**: *Start\Programs\Microsoft Word*.

☞ Ngoài ra nếu trên máy có thanh *Microsoft Office Shortcut Bar*, ta có thể khởi động từ thanh này bằng cách Click chuột vào biểu tượng  trên thanh này.

– Khởi động chương trình vẽ **Paint**: *Start/Program/Accessories* chọn *Paint*.


– Khởi động trình soạn thảo **Notepad**: *Start/Program/Accessories/ Notepad*.

↳ **Cách 3:** Khởi động từ Desktop, có 2 cách:

Bước 1: Double click vào đối tượng cần mở.

Bước 2: Click chuột phải lên đối tượng, chọn lệnh **Open** (có thể sử dụng cách này ở các vị trí khác trong Windows).

1.4.2. Đóng cửa sổ hay thoát khỏi ứng dụng: Có 4 cách chính

- Nhấn *Alt+F4* (đóng cửa sổ hiện hành).
- Nhấn vào nút *Close* () của cửa sổ.
- Click chuột phải vào biểu tượng của ứng dụng trên Taskbar, chọn *Close*.
- *Double click* vào *Control button*.

1.5. MÀN HÌNH NỀN DESKTOP

1.5.1. Các biểu tượng chuẩn/không chuẩn trên Desktop

a. Các biểu tượng chuẩn:

○ My Computer.

Là một thành phần mặc định, luôn luôn có mặt trên nền Desktop. Tất cả các công việc xử lý và thiết đặt trên máy tính của bạn đều được hiển thị trong cửa sổ của My Computer (như: các ổ đĩa mềm, ổ đĩa cứng, ổ



CD, Control Panel, Printer, thư mục, tập tin...).

o Recycle Bin.

Nằm trên Desktop và còn được gọi là *sọt rác*, đây là nơi tạm quản lý các đối tượng bị xóa (thường là các thư mục và tập tin). Trong Windows, khi ta xóa một đối tượng, nó sẽ được chuyển đến Recycle Bin (có một số trường hợp ngoại lệ). Do đó, chúng ta có thể dễ dàng *phục hồi* lại các đối tượng đã bị xóa từ Recycle Bin.



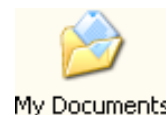
o My Network Places.

Giống như My Computer và Recycle Bin, Network Neighborhood cũng là một thành phần trên Desktop. Nếu bạn là một thành viên trong một mạng, các tài nguyên, thông tin được chia sẻ trên mạng sẽ được hiển thị trong My Network Places.



o My Documents.

Là một folder đặc biệt mà Windows đã tự động “tạo dùm” cho người sử dụng. Ta có thể tổ chức và lưu trữ các tập tin của mình vào My Documents. Tuy nhiên, ta vẫn có thể tạo cho mình một hoặc nhiều folder khác mà không nhất thiết phải sử dụng My Documents.



b. Các biểu tượng không chuẩn

o Folder (Ngăn xếp)


Là một đối tượng được tạo ra để chứa các ngăn xếp con hoặc các tập tin. (xem chương 3)



o Shortcut (lối tắt)

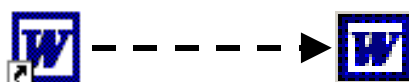
- Để giúp chúng ta có thể mở nhanh đến một trình ứng dụng, hoặc một Folder hay một đối tượng bất kỳ trên Windows, Microsoft đã đưa ra đối tượng shortcut. Nội dung của shortcut chính là đường dẫn chỉ đến đối tượng cần truy xuất nhanh đó. Khi ta mở shortcut chính là ta đang truy xuất đến đối tượng được chỉ định trong nó.

- Shortcut hoàn toàn khác với một trình ứng dụng, shortcut chỉ là một liên kết chứa đường dẫn đến một đối tượng nào đó (có thể là một tập tin, folder, drive, ...); do đó, nếu bạn xóa shortcut đi, bạn vẫn có thể tạo lại một cách dễ dàng mà không ảnh hưởng gì đến "đối tượng gốc".

- Biểu tượng của một Shortcut thường có hình  nằm góc dưới bên trái.

- Shortcut có thể được đặt trên desktop, trên menu Start, trên một folder bất kỳ.

Ví dụ: Biểu tượng của trình ứng dụng Microsoft Word và Shortcut của nó:



- **File (tập tin).**

Là các đối tượng chứa thông tin . Có nhiều loại tập tin và mỗi loại có một dạng biểu tượng khác nhau.

Ví dụ: , , , 

1.5.2. Các kiểu Desktop

Desktop có hai kiểu hiển thị: dạng thông thường (normal) và dạng trang web (web page).

Dạng thông thường là dạng mặc nhiên, khi cài đặt hoàn tất Windows Desktop được hiển thị ở dạng này. Ở dạng thông thường, muốn mở cửa sổ của các biểu tượng chuẩn hoặc các đối tượng được đại diện bởi shortcut ta phải nhấp đúp chuột (double click) vào biểu tượng hay shortcut đó.

Ở dạng Web, tên của các biểu tượng, shortcut được gạch dưới. Ta rê trỏ chuột đến đối tượng nào, trỏ chuột sẽ đổi thành hình bàn tay, đối tượng được chọn sẽ sáng lên. Chỉ cần nhấp chuột (click) để kích hoạt đối tượng, thay vì double click.

- **Thiết lập Desktop dạng web**

Bước 1: Chọn **Start / Settings / Control Panel/Folder Options**.

Bước 2: Trong hộp thoại **Folder Options** chọn thẻ **General**, chọn **Single-click to Open an item(point to select)**.

Bước 3: Click **OK**.

- **Chuyển Desktop về dạng thông thường (normal)**

Thực hiện tương tự trên nhưng chọn **Double-click to open an item (single- click to select)**.

1.5.3. Tạo, xóa và đổi tên folder/shortcut trên Desktop.

a. Tạo folder.

Bước 1: Click phải vào một điểm trống trên Desktop, chọn **New\Folder** từ menu tắt.

Bước 2: Nhập tên mới cho folder vừa tạo, nhấn **Enter**.

b. Tạo shortcut.

① Click chuột phải vào vùng trống trên Desktop, chọn **New**.

② Trong menu **New** chọn **Shortcut**, ta được hộp thoại **Create Shortcut**.

③ Nhấn vào nút **Browse** để xác định *thư mục chứa đối tượng* cần tạo shortcut, nhấp đúp vào đối tượng cần tạo để quay về hộp thoại **Create Shortcut**.

④ Nhấn nút **Next**. Gõ vào tên của shortcut (mặc nhiên sẽ lấy tên của đối tượng) và nhấn phím **Enter**.

c. Đổi tên folder/shortcut.

- ① Click phải vào folder/shortcut cần đổi tên
- ② Chọn lệnh **Rename** từ menu tắt
- ③ Nhập tên mới vào và nhấn **Enter**.

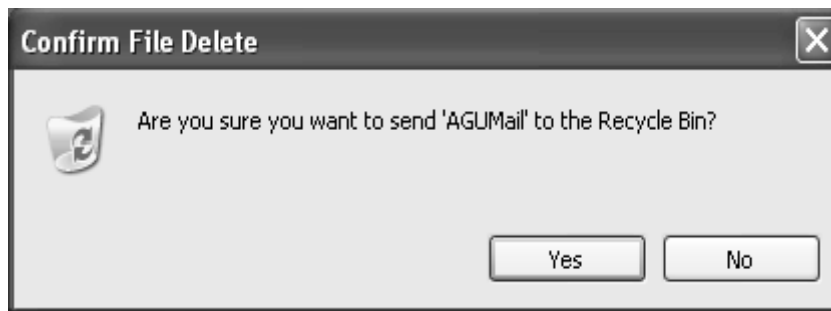
d. Xóa folder/shortcut.

Trong Windows khi xóa một đối tượng thì mặc nhiên sẽ đồng nghĩa với di chuyển đối tượng đó vào sọt rác (Recycle Bin).

↳ *Bước 1:*

- Cách 1: Chọn đối tượng cần xóa và nhấn phím *Delete*
- Cách 2: Nhấn chuột phải lên đối tượng cần xóa và chọn *Delete*.
- Cách 3: Nắm kéo đối tượng cần xóa đến *Recycle Bin*.

↳ *Bước 2:* Windows sẽ hiển thị một hộp thoại (tương tự hình 2.10) yêu cầu bạn xác nhận lại trước khi xóa. Chọn nút *Yes* nếu đồng ý; chọn nút *No* nếu muốn hủy bỏ lệnh xóa.



Hình 2.10. Hộp thoại yêu cầu xác nhận xóa một đối tượng.

1.5.4. Thay đổi ảnh nền cho Desktop.

Bước 1: Click chuột phải lên vùng trống của Desktop, chọn **Properties** bạn sẽ được hộp thoại **Display Properties**.

Bước 2: Chọn thẻ **Desktop** để có được hộp thoại tương tự hình 2.11.

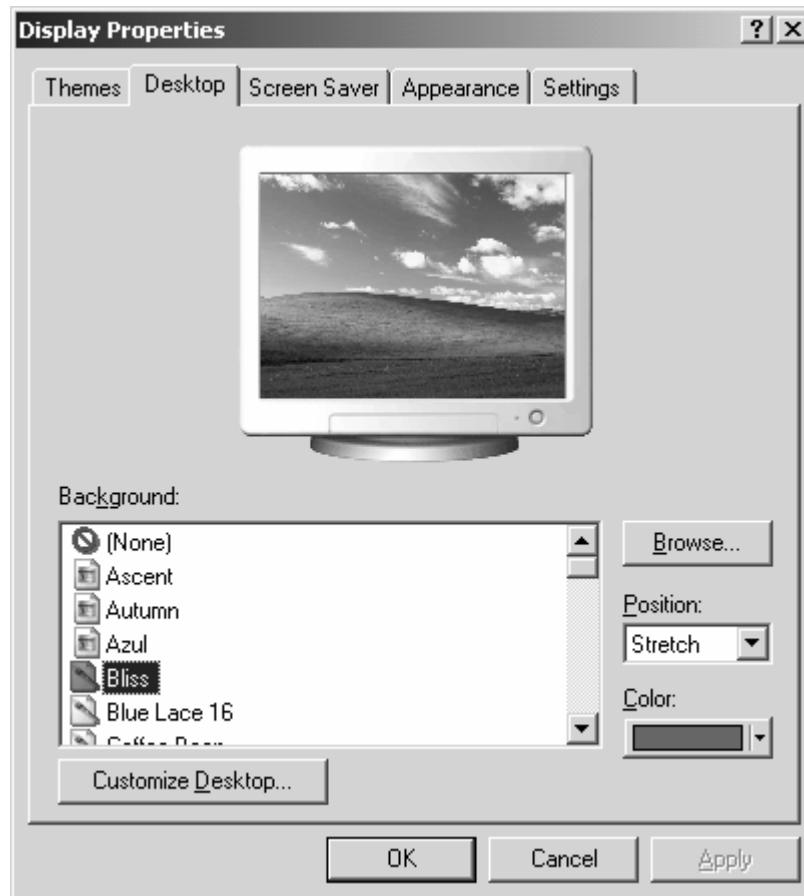
Bước 3: Thực hiện 1 trong 3 thao tác sau:

- Chọn tên tập tin hình ảnh hoặc tập tin HTML cần hiển thị trong danh sách *Select an HTML Document or a picture*.

- Nếu muốn bỏ ảnh nền trên Desktop hãy chọn **None**.

- Nếu muốn chọn một tập tin ảnh khác trên đĩa làm ảnh nền thì click nút *Browse....*

Sau đó xác định đường dẫn đến tập tin muốn dùng làm ảnh nền.



Hình 2.11. Chọn màn hình nền cho desktop

Bước 4: Nhấn nút **OK** hoặc **Apply** để chấp nhận các tùy chọn và đóng hộp thoại hoặc **Cancel** để hủy bỏ việc thay đổi ảnh nền.

1.6. CÁC THAO TÁC CƠ BẢN TRÊN CỬA SỔ

1.6.1. Mở/đóng cửa sổ.

Thao tác mở/đóng cửa sổ cũng chính là thao tác khởi động / thoát khỏi một ứng dụng làm việc ở dạng cửa sổ.

1.6.2. Phóng to, thu nhỏ và phục hồi cửa sổ

+ **Cách 1:** Click nút phóng lớn, thu nhỏ hay phục hồi ở góc trên phải cửa sổ.

+ **Cách 2:** Click vào hộp điều khiển (Control box) ở góc trên trái cửa sổ và chọn Maximize (phóng lớn), Minimize (thu nhỏ) hay Restore (phục hồi).

+ **Cách 3:** Right click biểu tượng của chương trình trên Taskbar và chọn Maximize (phóng lớn), Minimize (thu nhỏ) hay Restore (phục hồi).

1.6.3. Thay đổi kích thước (Resize) - Di chuyển cửa sổ (Move)

Nếu cửa sổ đang ở trạng thái phóng to hoặc thu nhỏ thì phải phục hồi lại mới có thể thay đổi kích thước hoặc di chuyển.

+ **Thay đổi kích thước:** Đưa trỏ chuột đến các cạnh hoặc các góc của cửa sổ cho đến khi trỏ chuột biến thành mũi tên hai đầu. rê chuột để thay đổi kích thước.

+ **Di chuyển:** Click và rê chuột tại thanh tiêu đề để di chuyển.

1.6.4. Chuyển đổi cửa sổ làm việc

Windows cho phép mở nhiều chương trình cùng lúc nhưng tại một thời điểm chỉ có thể làm việc với một chương trình (cửa sổ). Thông thường thanh tiêu đề (Title bar) của cửa sổ đang làm việc có màu đậm hơn các cửa sổ khác. Thực hiện một trong các cách sau để chuyển sang một cửa sổ khác:

Bước 1: Click vào biểu tượng cửa sổ trên Taskbar.

Bước 2: Click bất cứ điểm nào của cửa sổ muốn chọn nếu thấy.

Bước 3: Dùng tổ hợp phím *Alt+Tab* để chọn cửa sổ.

1.7. THAO TÁC VỚI TASKBAR

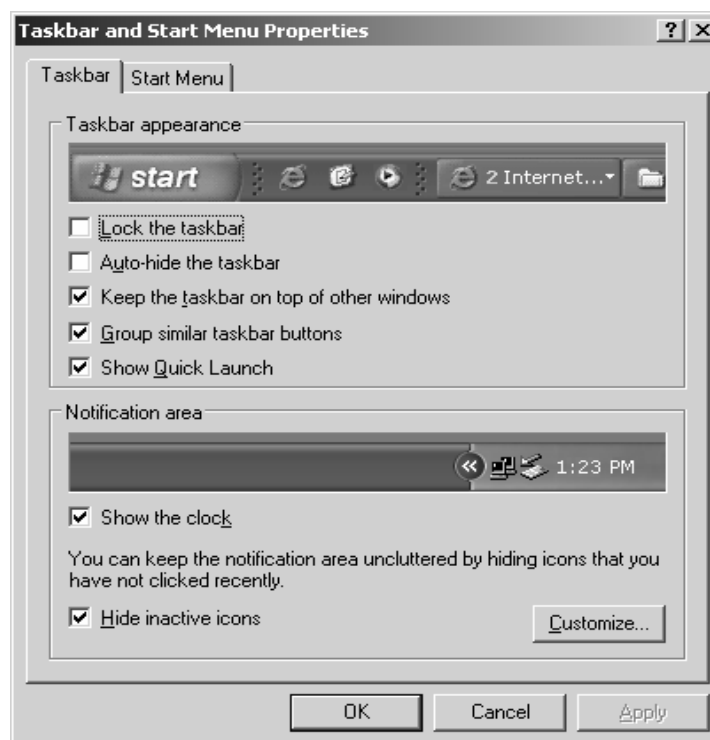
1.7.1. Di chuyển và thay đổi kích thước

Ngoài vị trí mặc định (cạnh dưới màn hình), bạn có thể di chuyển Taskbar đến các cạnh còn lại bằng cách chuyển chuột đến vùng trống trên Taskbar và nắm kéo đến vị trí mong muốn.

Ngoài ra bạn có thể chỉnh lại kích thước của Taskbar bằng cách đưa trỏ chuột ra biên của Taskbar, khi đó chuột sẽ có hình mũi tên hai đầu (↓ hoặc ↔) và bạn có thể kéo rộng hoặc thu nhỏ Taskbar như ý muốn.

1.7.2. Điều chỉnh Taskbar

Click chuột phải lên vùng trống của Taskbar (lưu ý không click trên các nút nằm trên thanh), chọn mục *Properties* để mở hộp thoại **Taskbar Properties**.



Hình 2.12 Hộp thoại Taskbar Properties

1.7.3. Thêm/xóa thanh công cụ trên Taskbar

Các thanh công cụ có thể thêm vào thanh Taskbar gồm: **Address** (Thanh địa chỉ của trình duyệt web), **Links** (Thanh chứa các liên kết của trình duyệt web), **Desktop** (Các biểu tượng trên desktop) và **Quick Launch** (Chứa các shortcut đến các chương trình thường dùng để khởi động nhanh).

Để thêm/xóa thanh công cụ trên Taskbar, làm theo các bước sau:

Bước 1: Nhấp chuột phải trên một khoảng trống (vị trí không có các nút hay các biểu tượng hiển thị) trên Taskbar, chọn **Toolbars** từ menu tắt.

Bước 2: Click vào thanh công cụ muốn thêm/xóa.

Bước 3: Nếu muốn tạo thanh công cụ mới thì chọn **New Toolbar**. Thanh công cụ mới mà bạn có thể tạo gồm nội dung của một thư mục, ổ đĩa hoặc một địa chỉ Internet.

1.7.4. Thêm/xóa shortcut trên Taskbar

Taskbar chỉ chứa shortcut cho các chương trình ứng dụng. Shortcut trên Taskbar được đặt trên thanh công cụ Quick Launch. Do đó, muốn thêm shortcut vào Taskbar thì trước tiên phải thêm thanh công cụ Quick Launch vào Taskbar.

○ *Thêm shortcut cho chương trình ứng dụng vào Taskbar:*

Bước 1: Thêm vào Taskbar thanh công cụ Quick Launch nếu chưa có.

Bước 2: Tìm tập tin thực thi của chương trình ứng dụng muốn tạo shortcut (có phần mở rộng là exe) rồi kéo và thả vào thanh công cụ Quick Launch.

○ *Xóa shortcut cho chương trình ứng dụng vào Taskbar:*

Bước 1: Nhấp chuột phải vào shortcut của chương trình ứng dụng muốn xóa trên Taskbar, chọn **Delete**.

Bước 2: Nhấp nút **Yes** để xóa.

1.8. PHỤC HỒI LẠI CÁC ĐỐI TƯỢNG ĐÃ XÓA TỪ RECYCLE BIN

Bước 1: Mở **Recycle Bin**.

Bước 2: Click chuột phải vào các đối tượng cần phục hồi, chọn **Restore** (các đối tượng sẽ được phục hồi lại vị trí trước khi bị xóa).

☞ **Lưu ý:** - Trong Windows, bạn có thể thực hiện các thao tác xóa, phục hồi, mở, ... với nhiều đối tượng cùng một lúc bằng cách giữ phím **Shift** hoặc **Ctrl** trong khi chọn hoặc dùng chuột để sơn sáng các đối tượng.

1.9. CÁC THAO TÁC TRÊN Ổ ĐĨA

1.9.1. Định dạng đĩa (Format)

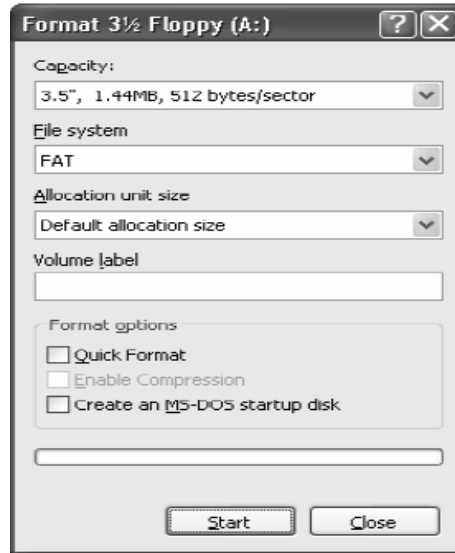
Đưa một đĩa mềm vào ổ đĩa A:

Bước 1: Click phải vào mục 3½ Floppy (A:) và click mục **Format** (hình 2.13)

Bước 2: Click nút Start để định dạng. Nếu đĩa hiện đang chứa dữ liệu sẽ xuất hiện một hộp thoại cảnh báo dữ liệu sẽ bị xóa, click nút Ok để yêu cầu định dạng.

Bước 3: Khi xuất hiện thông báo định dạng hoàn tất, click nút Ok.

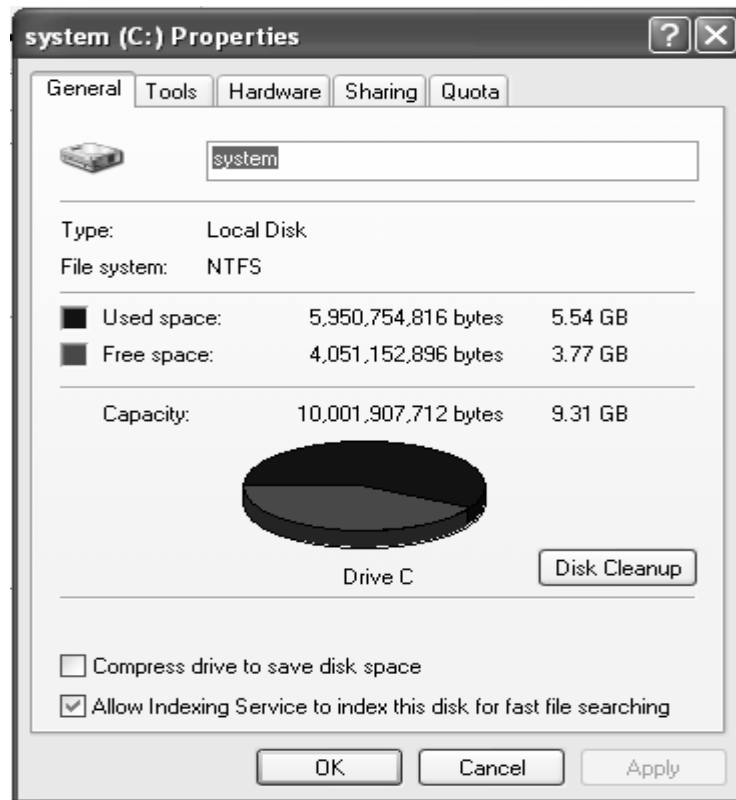
Bước 4: Click nút Close.



Hình 2.13. Định dạng đĩa mềm

1.9.2. Đổi tên đĩa và xem các thông số về ổ đĩa

Sau khi đặt tên cho đĩa, bất cứ lúc nào muốn đặt tên lại, bạn chỉ cần mở cửa sổ My Computer hoặc Windows Explorer, right click tại biểu tượng của ổ đĩa và chọn **Properties** để mở hộp thoại **Properties** (hình 2.14) , trong khung **Label** nhập vào tên mới và chọn **OK**.



Hình 2.14. Các thuộc tính của ổ đĩa

Trong hộp thoại **Properties** của ổ đĩa cũng cho biết các thông tin sau:

- **Biểu đồ hình tròn:** biểu thị số phần trăm dung lượng đĩa đã sử dụng và phần còn lại (Phần đã sử dụng có màu xanh, phần còn trống có màu hồng).
 - **Used space:** dung lượng đã sử dụng.
 - **Capacity:** tổng dung lượng đĩa.
 - **Free space:** dung lượng còn trống.

1.10. Regional Setting

Ta có thể qui ước các ký hiệu thập phân; ký hiệu phân cách ngày, tháng, năm; ký hiệu phân cách ngày, giờ; ký hiệu tiền tệ ... và định dạng các cách hiển thị cho chúng. Đây là những điều chỉnh hệ thống, nó sẽ ảnh hưởng đến các chương trình ứng dụng khác (như Microsoft Word, Microsoft Excel, ...).

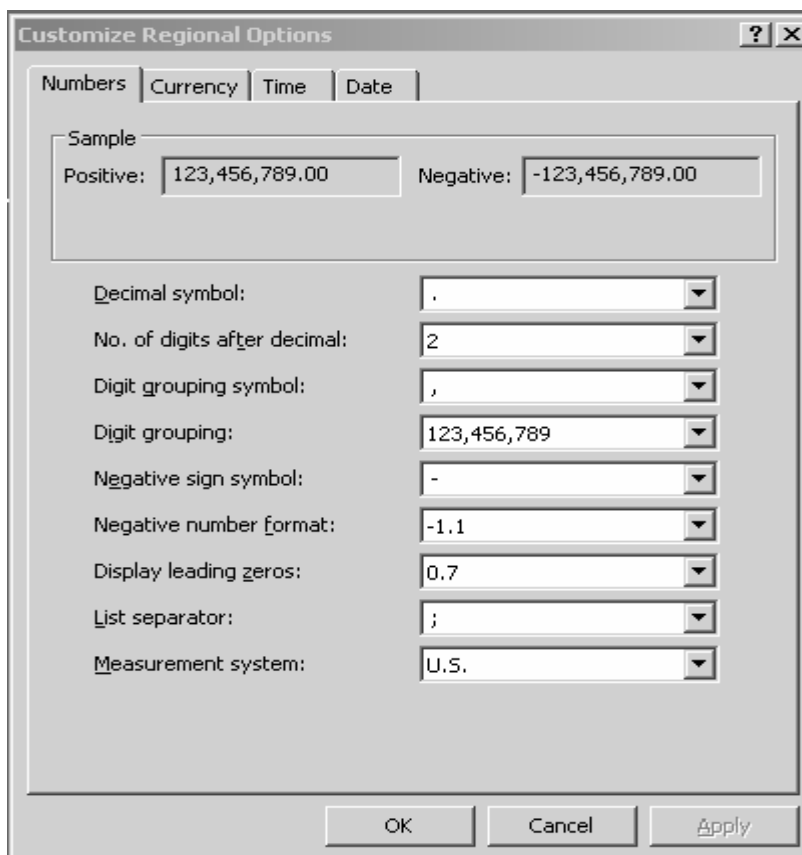
Bước 1: Vào Start\Settings\Control Panel.

Bước 2: Nhấp đúp vào biểu tượng Regional and Language Options .

Bước 3: Thiết lập các tùy chọn trong các thẻ sau:

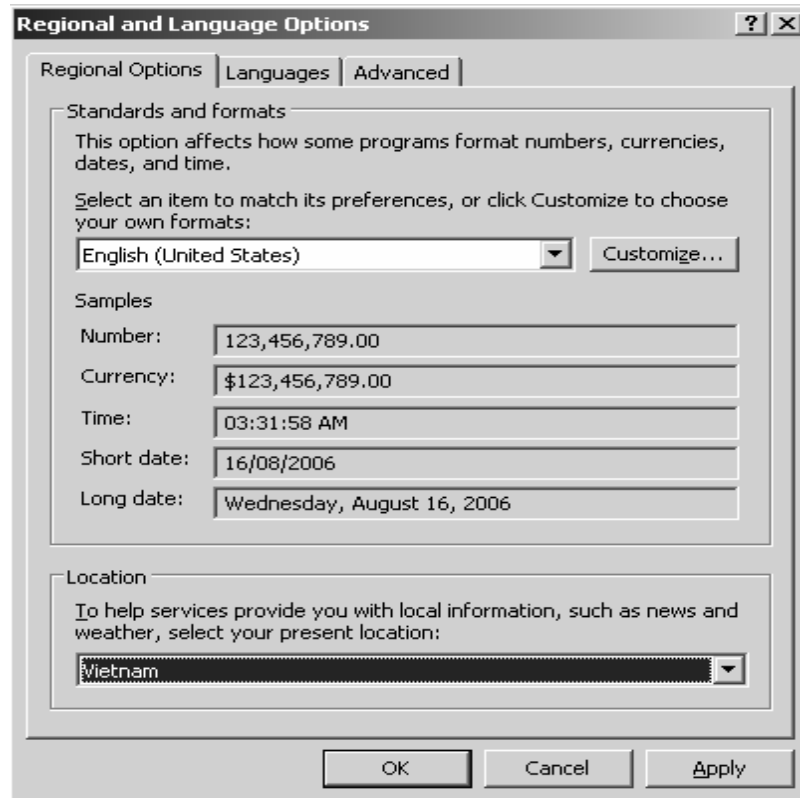
- Thẻ **Number** (hình 2.15):
 - *Decimal Symbol:* qui định ký hiệu thập phân.
 - *Digit grouping symbol:* ký hiệu phân cách hàng ngàn (ký hiệu phân cách nhóm các ký số).

- *No. of digits in group*: số ký số trong một nhóm.
- *List separator*: Dấu phân cách giữa các phần tử trong danh sách (đây cũng là dấu dùng để phân cách các đối số của một hàm).



Hình 2.15 Thẻ Number

- Thẻ **Date** (hình 2.16): Qui định cách hiển thị ngày tháng năm.
- Short date style: chọn kiểu hiển thị ngày theo dạng ngắn.
- Long date style: chọn kiểu hiển thị ngày theo dạng đầy đủ.
- Date separator: chọn ký hiệu phân cách ngày, tháng và năm.



Hình 2.16 Thẻ Date

CHƯƠNG 2.

CÁC VẤN ĐỀ LIÊN QUAN ĐẾN TẬP TIN, THƯ MỤC

2.1. TẬP TIN (FILE)

Tập tin là nơi tổ chức và lưu trữ thông tin dùng cho máy tính. Có nhiều kiểu tập tin chẳng hạn như: tập tin văn bản (document), tập tin chương trình (program),...

Mỗi tập tin đều có tên riêng, vị trí lưu trên đĩa và những thuộc tính riêng. Tên tập tin do người sử dụng đặt ra gồm có 2 phần phân cách nhau bằng dấu chấm là phần tên và phần mở rộng.

- **Phần tên** (name): Là phần bắt buộc phải có, có thể được cách nhau bởi khoảng trống (space), có chiều dài tối đa là 255 ký tự và không được chứa các ký tự sau: \ / : * ? " < >.

- **Phần mở rộng** (Extension): Là phần không bắt buộc phải có, dùng để phân biệt loại tập tin.

* Ký tự đại diện (Wildcard) :

- Dấu ? : Đại diện cho một ký tự bất kỳ trong tên tập tin tại vị trí nó xuất hiện.
- Dấu * : Đại diện cho một dãy ký tự bất kỳ trong tên tập tin kể từ vị trí nó xuất hiện đến hết phần tên hoặc phần mở rộng.

Ví dụ: Giả sử trên đĩa có các tập tin: THO1, THO2, THO3, THOCA, VANBAN.

. Ký hiệu THO? Sẽ đại diện 3 tập tin: THO1, THO2, THO3.

. Ký hiệu THO* sẽ đại diện 4 tập tin: THO1, THO2, THO3, THOCA.

. Ký hiệu *.* sẽ đại diện cho tất cả các tập tin.

Vài loại tập tin thông dụng có phần mở rộng:

- **EXE** hoặc **.COM**: Tập tin các chương trình khả thi.
- **BMP, .JPG, .GIF, .PSD**: Tập tin hình ảnh.
- **DOC**: Tập tin văn bản của chương trình Microsoft Word.
- **XLS**: Tập tin dữ liệu của bảng tính Microsoft Excel.

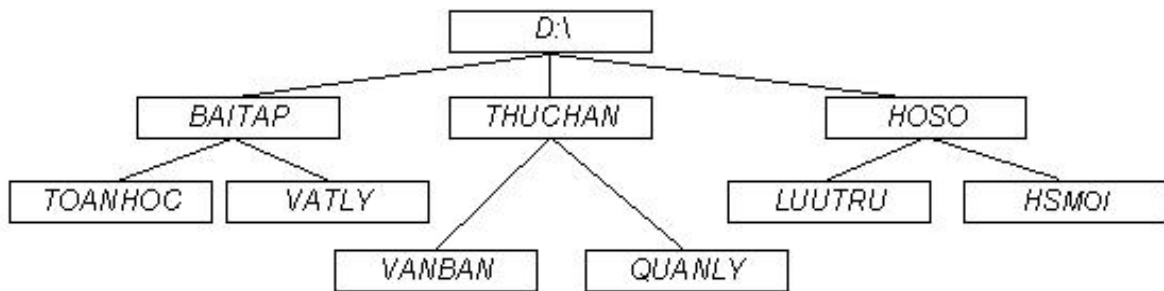
Ba tập tin hệ thống cơ bản của hệ điều hành: COMMAND.COM, IO.SYS, MSDOS.SYS. Đây là ba tập tin cần thiết để khởi động máy tính vào chế độ dấu nhắc lệnh của DOS.

2.2. THƯ MỤC/NGĂN XẾP (FOLDER)

Để dễ dàng quản lý tập tin người ta để các tập tin có chung đặc tính nào đó (như cùng một chủ đề, cùng tác giả, cùng một chương trình....) vào một nhóm riêng và cất giữ trong một “vùng riêng” trên đĩa gọi là *Thư mục/ngăn xếp*.

Một đĩa có thể được chia thành nhiều thư mục, các thư mục có thể chứa nhiều tập tin và các thư mục con khác... Như vậy cấu trúc thư mục trên đĩa có dạng hình cây.

Ví dụ:



Hình 2.17 Cây thư mục

Các khái niệm về thư mục:

- **Tên thư mục:** tuân thủ theo qui tắc đặt tên của tập tin. Tuy nhiên, không nên đặt tên thư mục có phần mở rộng vì như vậy có thể nhầm lẫn với tên tập tin. Trong cùng thư mục không thể có hai tập tin, hoặc hai thư mục con, hoặc một thư mục và một tập tin trùng tên nhau.

- **Thư mục gốc:** trên mỗi đĩa có ít nhất một thư mục, đó là thư mục gốc. Thư mục gốc là thư mục do hệ điều hành tạo ra khi định dạng đĩa. Thư mục gốc là nơi chứa tất cả các thư mục con và tập tin khác trên đĩa.

- **Thư mục con:** là thư mục được chứa bên trong thư mục khác.

- **Thư mục cha:** thư mục nào có chứa thư mục con bên trong đều được gọi là thư mục cha của các thư mục con đó.

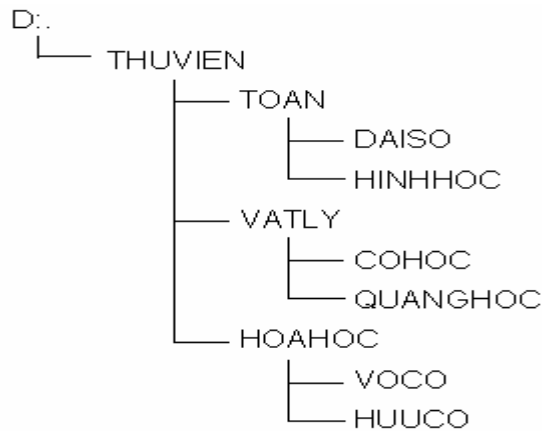
- **Thư mục hiện hành:** là thư mục đang được chọn.

- **Thư mục rỗng:** là thư mục không có chứa tập tin và thư mục con khác.

2.3. ĐƯỜNG DẪN (PATH)

Khi sử dụng thư mục nhiều cấp thì cần phải chỉ rõ cho thư mục nào mà bạn muốn truy xuất, để chỉ định thư mục người ta dùng đường dẫn. Đường dẫn là một chuỗi các tên thư mục được viết phân cách với nhau bởi dấu \.

Ví dụ: Giả sử ta có cây thư mục của ổ đĩa D: được tổ chức như sau:



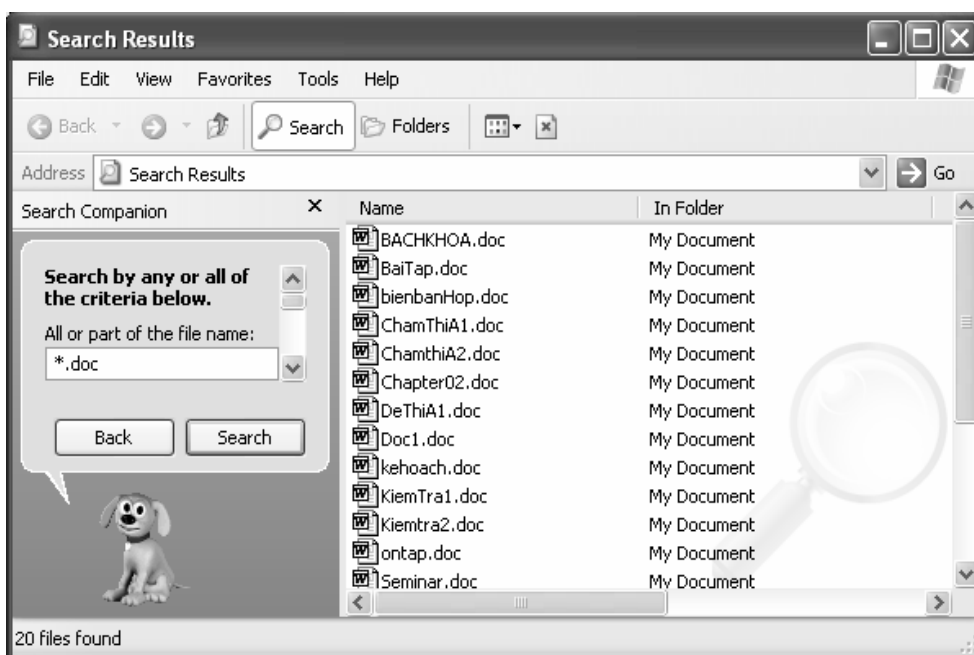
Hình 2.18

☞ Qua cây thư mục trên ta thấy:

- THUVIEN là thư mục con của thư mục gốc D:\
- TOAN, VATLY, HOAHOC đều là thư mục con của thư mục THUVIEN, chúng còn được gọi là thư mục cùng cấp.
- Tương tự, DAISO và HINHHOC là thư mục con cùng cấp của thư mục TOAN; COHOC và QUANGHOC là thư mục con cùng cấp của thư mục VATLY ...Nhu vậy đường dẫn của thư mục TOAN là: D:\THUVIEN\TOAN, tương tự:
 - Thư mục COHOC sẽ có đường dẫn là: D:\THUVIEN\VATLY\COHOC
 - Thư mục HUUCO sẽ có đường dẫn là: D:\THUVIEN\HOAHOC\HUUCO

2.4. TIỆN ÍCH TÌM KIẾM TẬP TIN/THƯ MỤC

Bước 1: Vào Start\Search\ For Files & Folder để mở hộp thoại Search Results.



Hình 2.19 Tìm kiếm tập tin hoặc thư mục

Bước 2: Click mục All files and folders

Bước 3: Nếu muốn tìm theo tên tập tin hay thư mục, hãy gõ vào đầy đủ hay một phần của tên file trong hộp **All or part of file name**

Bước 4: Hoặc nếu muốn tìm trong nội dung file gõ vào một từ hay một cụm từ đại diện cần tìm trong hộp **A word or phrase in the file.**

Bước 5: Bạn có thể chỉ ra nơi cần tìm vào bằng cách click vào mũi tên hướng xuống trong hộp **Look in**, sau đó chọn ổ đĩa hay thư mục.

Bước 6: Click nút **Search.**

CHƯƠNG 3.

TRÌNH WINDOWS EXPLORER

Windows Explorer là một công cụ được tích hợp sẵn trong Windows giúp bạn quản lý và khai thác những tài nguyên của máy tính. Với Windows Explorer, bạn có thể dễ dàng thực hiện các thao tác khởi động các ứng dụng, tạo, xóa, sao chép, di chuyển thư mục và tập tin, ...

3.1. KHỞI ĐỘNG VÀ THOÁT

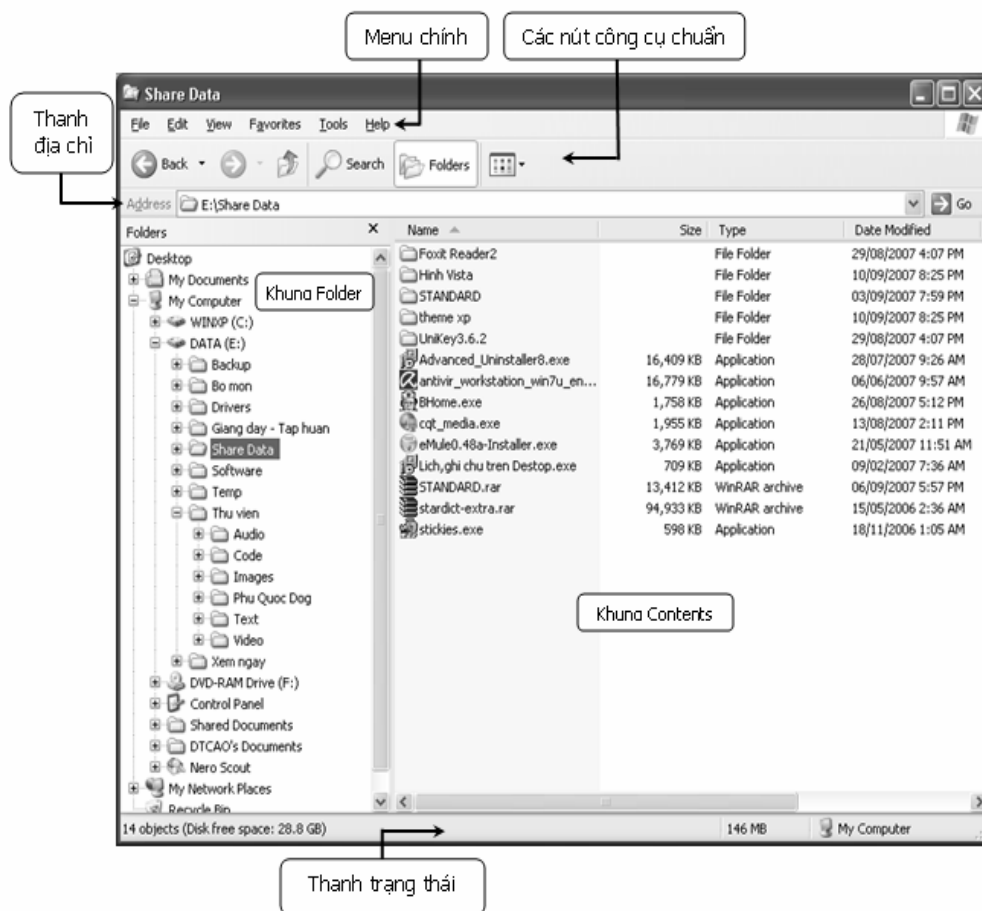
3.1.1. Khởi động

Có 3 cách chính:

- Chọn **Start\Programs\Accessories\Windows Explorer**.
- Chọn **Start\Run**, gõ **Explorer** và nhấn **Enter**.
- Click chuột phải vào các Folder hoặc các đối tượng có đặc tính như Folder (My Computer, Network Neighborhood,...) *trên desktop* và chọn **Explorer**.

3.1.2. Giao diện

Sau khi khởi động, cửa sổ của Windows Explorer sẽ có dạng tương tự sau:



Hình 2.20 Giao diện Windows Explorer.

Cửa sổ được chia thành hai phần: phần bên trái được gọi là *bảng cây thư mục (Folders pane)*, phần bên phải được gọi là *bảng nội dung (Contents pane)*. Phía trên cửa sổ là các thanh công cụ với các nút lệnh giúp cho người dùng có thể thực hiện nhanh các thao tác.

🔗 **Lưu ý:** Nếu không thấy bảng cây thư mục, bạn hãy thực hiện như sau: vào menu **View/Explorer Bar** và chọn **Folders**.

🔗 **Các thành phần trong cửa sổ Windows Explorer:**

- **Thanh trình đơn (Menu):** chứa các lệnh trong Explorer.
- **Thanh công cụ chuẩn (Standard Buttons):** thanh công cụ chứa các nút lệnh thường được sử dụng nhất.

Thao tác	Lệnh	Nút lệnh	Phím tắt
Tạo folder/files	File\New\...		
Sao chép	Edit\COPY	Copy	Ctrl + C
Cắt	Edit\Cut	Cut	Ctrl + X
Dán	Edit\Paste	Paste	Ctrl + V
Đổi tên	File\Rename		F2
Xóa	File\Delete	Delete	Delete


• **Thanh địa chỉ (Address Bar):** cho biết đường dẫn của thư mục hiện hành. Bạn có thể chuyển nhanh đến các ổ đĩa bằng cách chọn từ danh sách, hoặc có thể mở một Folder hay một tập tin bất kỳ từ thanh này bằng cách *gõ vào đường dẫn và tên đối tượng cần mở*.

Phím tắt: **F4** (hoặc **Alt + D**)

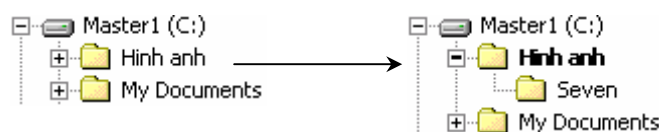
• **Bảng cây thư mục (Folder pane):** hiển thị các thành phần, các folder trên máy theo dạng hình cây. Muốn xem nội dung của folder nào bạn chỉ việc click chuột vào folder đó, nội dung của folder này sẽ hiện trên bảng Contents.

🔗 **Lưu ý:**

- Bảng Folder chỉ hiển thị cấu trúc các folder theo dạng hình cây, bảng Content sẽ hiển thị nội dung folder.

- Folder đang mở (folder hiện hành) sẽ có biểu tượng: 

- Các folder có hình dấu cộng (+) phía trước có nghĩa là bên trong nó còn có chứa folder con khác. Click chuột vào dấu cộng này bạn sẽ thấy được các folder con, khi đó nó sẽ đổi thành hình dấu trừ (-) và nhấp dấu trừ sẽ có kết quả ngược lại.



• **Bảng nội dung (Contents pane):** đây là nơi hiển thị nội dung của folder hiện hành. Trong bảng này, *muốn mở đối tượng nào bạn hãy double click vào đối tượng đó*. Tức là

muốn chọn một folder hiện hành ta double click vào nó; để mở *một tập tin* ta double click vào tập tin đó.

- **Thanh trạng thái (Status Bar):** cho biết thông tin về các đối tượng đang chọn.



3.1.3. Thoát.

Để thoát khỏi Windows Explorer ta có thể sử dụng một trong 4 cách đóng một cửa sổ trong Windows.

3.2. CÁC THAO TÁC CƠ BẢN

3.2.1. Đóng/mở các folder.


Bên khung trái màn hình Windows Explorer hiển thị các đề mục, ổ đĩa và các ngăn xếp (folder). Các ngăn xếp có dấu (+) đứng trước cho biết trong đó có các ngăn xếp con chưa được hiển thị. Muốn hiển thị, click vào dấu (+) các ngăn xếp con sẽ xuất hiện và khi đó dấu (+) biến thành dấu (-). Muốn giấu các ngăn xếp con click dấu (-) để chuyển thành dấu (+).

Để mở một ngăn xếp bên khung trái click vào ngăn xếp đó, biểu tượng của ngăn xếp sẽ đổi từ  sang  và nội dung hiển thị ở khung phải.

3.2.2. Chọn chế độ hiển thị

Có bốn dạng hiển thị ngăn xếp/tập tin:

- **Large icons:** biểu tượng lớn.
- **Small icons:** biểu tượng nhỏ.
- **List:** danh sách.
- **Details:** chi tiết.

Bạn có thể chọn một kiểu hiển thị khác bằng cách mở menu **View** hoặc click biểu tượng **View**  trên thanh công cụ, sau đó click chọn một trong các cách trên.

3.2.3. Sắp xếp các ngăn xếp/tập tin

Chọn **View\Arrange icons**, hoặc right click tại khoảng trống giữa các ngăn xếp/tập tin trong cửa sổ My Computer hay khung phải Windows Explorer và chọn **Arrange icons**, sau đó chọn một trong các cách sắp xếp sau:

- **By Name:** theo tên.
- **By Type:** theo loại tập tin.
- **By Size:** theo độ lớn.
- **By Date:** theo ngày tháng được tạo ra.

3.2.4. Chọn ngăn xếp/tập tin

Để làm việc với các ngăn xếp và tập tin nhất thiết bạn phải chọn chúng trước. Có nhiều cách để chọn (thao tác trong cửa sổ My Computer hoặc khung phải Windows Explorer).

- Chọn một ngăn xếp hoặc một tập tin: Chỉ cần click trỏ chuột vào ngăn xếp hoặc tập tin, ngăn xếp hoặc tập tin đổi màu và được chọn.

- Chọn nhiều ngăn xếp hoặc tập tin đứng liền nhau: Có thể thực hiện một trong những cách sau:

+ Giữ phím **Shift** và mở rộng khối chọn bằng các phím mũi tên.

+ Click chuột vào đối tượng đầu khối, giữ phím **Shift**, click chuột vào đối tượng cuối khối.

- Chọn những ngăn xếp hoặc tập tin không liền nhau: Giữ phím **Ctrl** và click các đối tượng muốn chọn.


3.4. CÁC THAO TÁC TRÊN NGĂN XẾP/TẬP TIN

3.4.1. Tạo ngăn xếp

Bước 1: Mở folder sẽ chứa folder cần tạo, click chuột phải lên vùng trống bên khung Contents và chọn lệnh **New\Folder**.

Bước 2: Gõ vào tên folder và nhấn **Enter**.


3.4.2. Sao chép ngăn xếp/tập tin

Bước 1: Chọn đối tượng cần sao chép nhấn **Ctrl + C** (hoặc nút ).

Bước 2: Chọn folder đích (nơi cần chép đến), nhấn **Ctrl + V** (hoặc nút .

3.4.3. Di chuyển ngăn xếp/tập tin

Bước 1: Chọn đối tượng cần di chuyển nhấn **Ctrl + X** (hoặc nút .


Bước 2: Chọn folder đích (nơi cần chép đến), nhấn **Ctrl + V** (hoặc nút .

3.4.4. Đổi tên ngăn xếp/tập tin

Bước 1: Nhấn chuột phải chọn **Rename** hoặc chọn đối tượng cần đổi tên và nhấn **F2**.

Bước 2: Nhập tên mới vào và nhấn **Enter**.

3.4.5. Xóa ngăn xếp/tập tin

Bước 1: Chọn đối tượng cần sao chép nhấn phím **Delete** (hoặc nút .


Bước 2: Chọn **Yes** trong hộp thoại xác nhận xóa (tương tự hình 2.10).

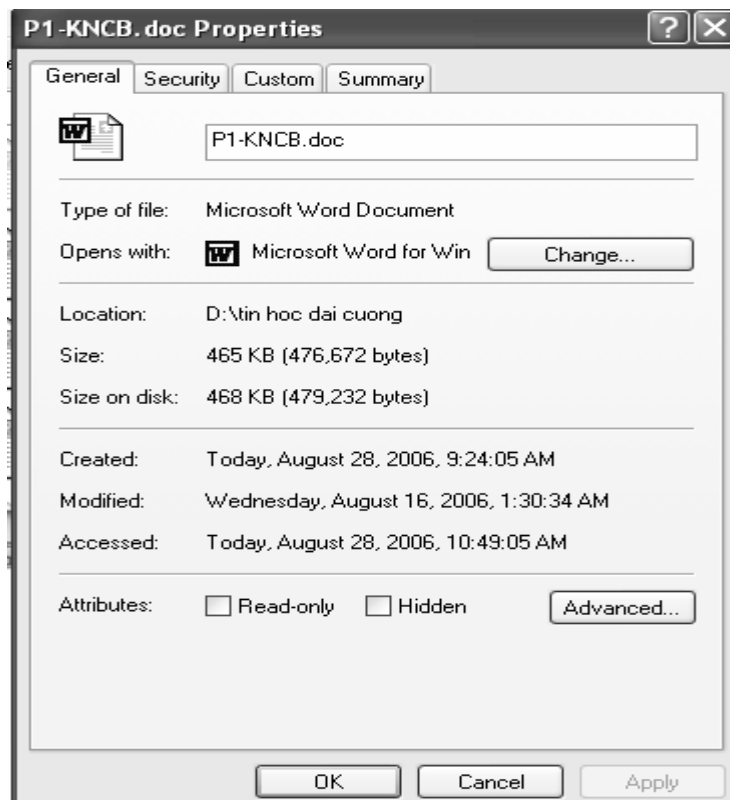
3.4.6. Tạo shortcut cho ngăn xếp/tập tin

- **Cách 1:** Sử dụng chuột phải, nắm kéo đối tượng đến nơi cần tạo shortcut và chọn lệnh **Create Shortcut**.
- **Cách 2:** Chọn đối tượng cần tạo, click chuột phải và chọn lệnh **Create Shortcut**. Sau đó chép shortcut này đến nơi cần thiết.
- **Cách 3:** Cách này thực hiện giống như tạo folder nhưng ta chọn **New/Shortcut**. Sau đó thực tiếp thêm 2 bước xác định đường dẫn đến folder/file cần tạo shortcut và đặt tên cho shortcut (xem phần tạo shortcut trên Desktop - phần 1.5.3 chương 1 - phần 2).

3.4.7. Xem và thay đổi thuộc tính ngăn xếp/tập tin

① Mở hộp thoại Properties (hình 2.21) bằng một trong các cách sau:

- Cách 1: Click chuột phải lên đối tượng, chọn **Properties**.
- Cách 2: Chọn đối tượng cần xem nhấn **Alt + Enter** (hoặc click nút ).



Hình 2.21

② Windows sẽ hiển thị các thuộc tính của đối tượng như sau:

- Biểu tượng và tên tập tin.
- **Type:** loại tập tin.
- **Location:** vị trí của tập tin/ngăn xếp trên đĩa.
- **Size:** kích thước của tập tin/ngăn xếp.
- **Created:** ngày tạo.

- **Modified:** ngày chỉnh sửa gần nhất,....

Và các thuộc tính truy xuất (Attribute):

- **Read-only:** chỉ xem, không được thay đổi hoặc xoá nội dung.
- **Hidden:** Ẩn; **Archive:** Lưu trữ; **System:** Hệ thống.

Bạn có thể thay đổi các thuộc tính này bằng cách đánh dấu vào vị trí tương ứng.

CHƯƠNG 4.

INTERNET EXPLORER

4.1. CÁC KHÁI NIỆM

4.1.1. Khái niệm

Internet là hệ thống gồm các mạng máy tính được liên kết với nhau trên phạm vi toàn thế giới theo giao thức chuẩn TCP/IP, thông qua hệ thống các kênh truyền thông.

Internet cung cấp nhiều dịch vụ dùng để tạo, duyệt xét, tìm kiếm, truy cập, xem và trao đổi thông tin trên nhiều lĩnh vực khác nhau, từ nghiên cứu khoa học đến các hoạt động giải trí, thương mại,...

4.1.2. Các dịch vụ cơ bản trên Internet

- **Dịch vụ Web (WWW - World Wide Web):** Là dịch vụ quan trọng nhất trên Internet ngày nay. Dịch vụ Web cho phép bạn dùng các Web Browser (như Internet Explorer) truy cập thông tin từ bất cứ nơi nào trên thế giới, miễn là bạn được phép xem thông tin đó. Thông tin trả về trong Web Browser dưới dạng các trang Web bao gồm văn bản, hình ảnh, âm thanh và có cả các đoạn phim (multimedia) theo ngôn ngữ định dạng văn bản HTML.

- **Dịch vụ thư điện tử (E-mail):** Cho phép bạn sử dụng các chương trình gửi và nhận thư (Mail Client) điện tử trên mạng. Đây là dịch vụ ra đời sớm nhất và lâu đời nhất.

- **Dịch vụ truyền tập tin FTP (File Transfer Protocol):** Cho phép dùng các FTP Client để truyền và nhận các tập tin trên mạng. Đây là dịch vụ chính trong việc chia sẻ chương trình, tập tin trên mạng. Giao thức được sử dụng ở đây là giao thức truyền nhóm file FTP (File Transfer Protocol).

- **Dịch vụ tán gẫu trên Internet (Chat):** Là hình thức hội thoại trực tuyến (trò chuyện) trên Internet. Với dịch vụ này, hai hay nhiều người có thể cùng trao đổi thông tin trực tuyến qua bàn phím máy tính, qua các tập tin hay qua các whiteboard (giống như dùng phấn viết lên bảng cho mọi người xem).

- **Các diễn đàn (Forum):** Diễn đàn thảo luận cho phép bạn trao đổi thông tin với những người quan tâm ở tất cả mọi nước. Nếu không đăng nhập, bạn có thể truy cập trực tiếp vào các diễn đàn này để đọc bài. Nhập tên truy cập và mật khẩu (sử dụng tên và mật khẩu đăng ký trên vol.vnn.vn).

- **Dịch vụ tìm kiếm thông tin (Search Engine):** Tìm kiếm thông tin trên mạng dựa vào một từ khóa nhập vào từ bàn phím.

4.2. GIỚI THIỆU GIAO DIỆN INTERNET EXPLORER

Internet Explorer là trình duyệt web của hãng Microsoft. Đây là trình duyệt được tích hợp sẵn trong Windows.

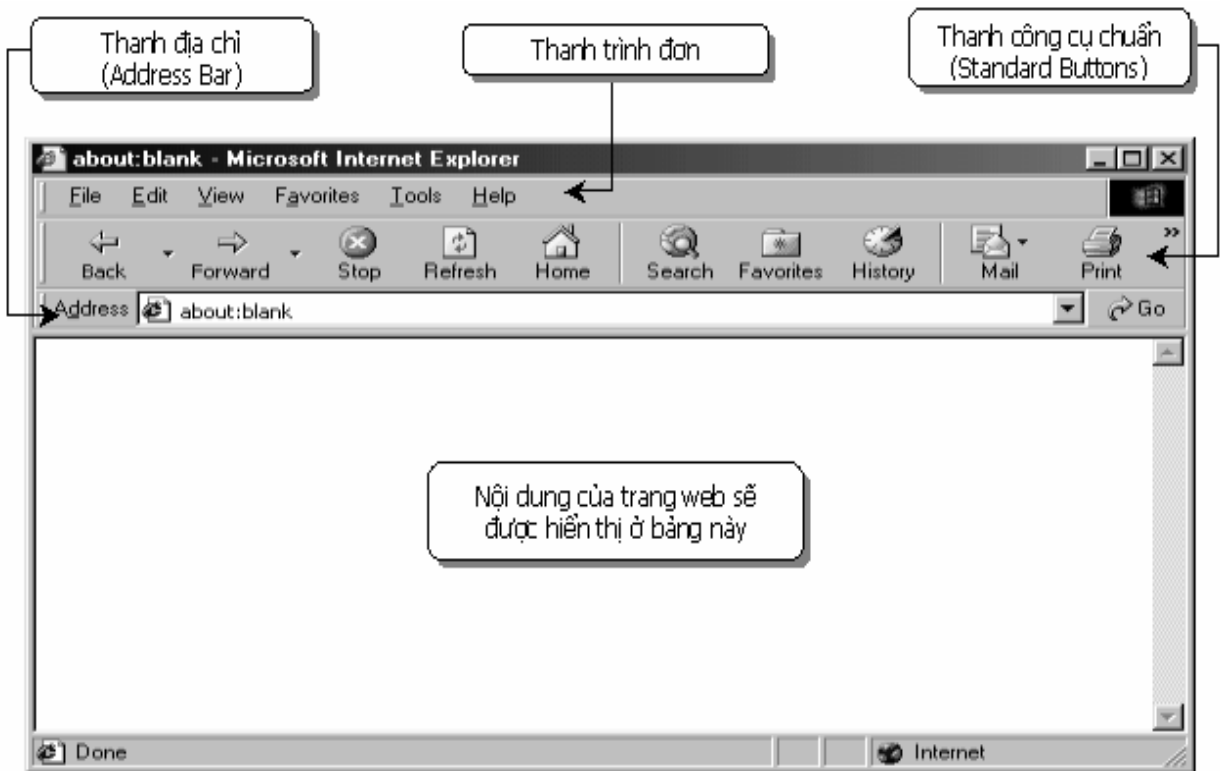
4.2.1. Khởi động

- **Cách 1:** click vào biểu tượng của Internet Explorer  trên thanh **QuickLaunch** của Taskbar hoặc trên Desktop.

- **Cách 2:** vào menu **Start\Programs\Internet Explorer**.

4.2.2. Giao diện

Sau khi khởi động, cửa sổ của Internet Explorer sẽ có dạng tương tự sau:



Hình 2.22. Cửa sổ Internet Explorer

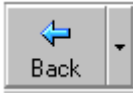






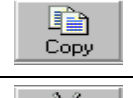
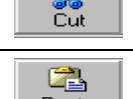
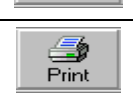
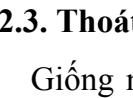
Cửa sổ của IE cũng có các thành phần tương tự như các cửa sổ ứng dụng khác trong Windows như: Thanh tiêu đề, các nút Minimize, Maximize, Restore, Close, các thanh cuộn,... Ngoài ra còn có thanh trình đơn, thanh thanh địa chỉ (Address Bar), thanh công cụ chuẩn của IE (Standard Buttons).

- **Thanh trình đơn (menu):** chứa các lệnh của IE, tùy theo chức năng mà chúng được tổ chức theo từng nhóm.

- **Thanh địa chỉ (Address Bar):** đây là nơi mà bạn sẽ gõ vào địa chỉ của trang web mà bạn muốn xem. Ngoài ra, nó còn có ý nghĩa là cho biết địa chỉ của trang đang được xem trong trường hợp mở nó từ một trang liên kết khác. Ta có thể truy xuất nhanh đến thanh này bằng phím F4 hoặc tổ hợp phím **Alt-D**.

- **Thanh công cụ chuẩn (Standard Buttons):** đây là công cụ chính chứa các nút lệnh thường xuyên được sử dụng nhất. Bao gồm các nút chính sau:

Nút lệnh	Phím tắt	Công dụng
----------	----------	-----------

	Alt - ←	Quay trở lại trang trước đó
	Alt - →	Đi tới trang sau
	Esc	Ngưng tải xuống trang hiện tại
	F5	Cập nhật lại trang hiện hành
	Alt-Home	Quay trở về trang chủ
	Ctrl-H	Xem danh sách các trang web vừa truy cập
	Ctrl-I	Xem danh sách các địa chỉ ưa thích
	Ctrl-C	Chép thông tin đang được chọn vào Clipboard
	Ctrl-X	Chuyển thông tin được chọn vào clipboard
	Ctrl-V	Dán thông tin từ clipboard ra ngay vị trí của con trỏ
	Ctrl-P	In trang web hiện hành ra máy in

4.2.3. Thoát

Giống như các cửa sổ khác trên Windows, bạn có thể nhấn **Alt+F4** từ bàn phím hoặc click chuột vào nút **Close** ở góc trên bên phải cửa sổ.

4.3. CÁC THAO TÁC CƠ BẢN TRONG IE

4.3.1. Mở một trang web

☞ Cách 1:

- Chuyển trỏ đến thanh địa chỉ (Click chuột lên thanh hoặc nhấn **Alt-D**).
- Gõ vào đường dẫn và tên trang web cần mở.

☞ Cách 2:

- Nhấn **Ctrl-O** hoặc vào menu **File\Open**.
- Gõ vào đường dẫn và tên trang web cần mở (nhấn vào nút Browse để tìm nếu đó là một trang được lưu trên đĩa của máy).

4.3.2. Các thao tác cơ bản trong một trang Web

- Xem thông tin bị khuất trên trang web: Sử dụng các cách sau:

- Dùng chuột để điều khiển các thanh cuộn.
 - Dùng các phím mũi tên (lên, xuống, trái, phải) để di chuyển từng dòng.
 - Sử dụng phím Page Up/Down để di chuyển lên/xuống một trang màn hình.
 - Dùng phím Home, End để di chuyển nhanh đến đầu trang hoặc cuối trang.
- Chọn văn bản, hình ảnh,... trong trang web: dùng chuột để quét khối hoặc kết hợp **Shift** với các phím mũi tên để chọn.

4.3.3. Chọn bảng mã hiển thị cho trang web

Nếu bạn mở một trang web mà thông tin trên đó toàn là các ký tự lạ, bạn hãy chọn lại bảng mã hiển thị cho nó:

Vào menu View/Encoding (hoặc nhấn vào nút Encoding trên thanh công cụ) chọn bảng mã tương ứng trong danh sách. Thông thường bạn nên chọn thử một trong ba mục sau:

- Auto-Select.
- User Defined.
- Unicode (UTF-8).

4.3.4. Chép các thông tin trong trang web sang Microsoft Word

Bước 1: Chọn các thông tin cần chép.

Bước 2: Nhấn **Ctrl-C** (hoặc nhấn vào nút Copy trên thanh công cụ).

Bước 3: Chuyển sang cửa sổ Word, nhấn **Ctrl-V**.

✎ **Lưu ý:**

- Thông tin cần chép có thể là đoạn văn bản thuần túy, hoặc hình ảnh hay một đoạn văn bản có chứa hình.

- Có thể sử dụng cách này để chép sang các ứng dụng khác.

4.3.5. Lưu một hình ảnh trong trang vào đĩa

Bước 1: Click chuột phải lên hình cần lưu, chọn **Save picture as ...**

Bước 2: Trong hộp thoại **Save picture as**, chọn ổ đĩa và thư mục chứa tập tin hình ảnh.

Bước 3: Gõ vào tên mới ở hộp **File name** nếu muốn đổi tên tập tin, nhấn **Enter**.

4.3.6. Lưu trang web vào đĩa

Bước 1: Chọn menu **File\Save as**.

Bước 2: Trong hộp thoại **Save Web page**, chọn ổ đĩa và thư mục chứa trang web.

Bước 3: Gõ vào tên mới ở hộp **File name** nếu muốn đổi tên, nhấn **Enter**.

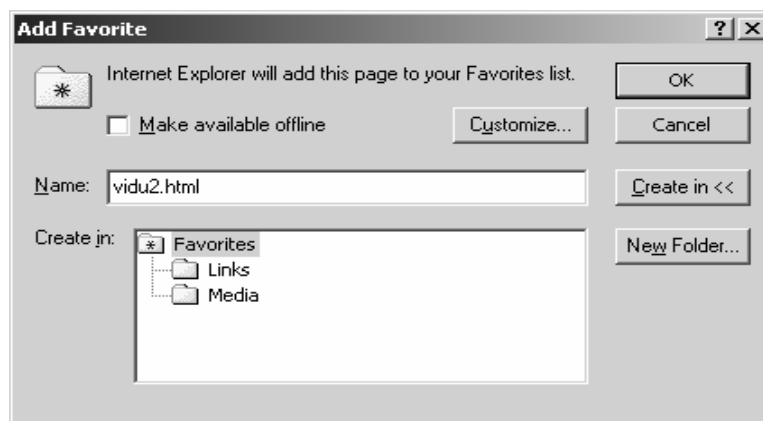
4.3.7. Đưa trang web hiện hành vào danh sách các trang ưa thích

Bước 1: Nhấn vào nút Favorites, để hiển thị khung Favorites bên trái cửa sổ trình duyệt.



Hình 2.23

Bước 2: Trên thanh Favorites, nhấn vào mục **Add...** để mở hộp thoại Add Favorite.



Hình 2.24

⚠ **Lưu ý:** Trong trường hợp hộp thoại Add Favorite không có bảng Create in, hãy nhấn vào nút Create in.

Bước 3: Nhập vào tên trang web cần lưu ở mục Name (mặc định sẽ lấy tên hiện tại của trang).

Bước 4: Đánh dấu vào mục *Make available offline* nếu muốn duyệt trang web sau này mà không cần vào mạng.

Bước 5: Chọn folder chứa địa chỉ trang web trong bảng *Create in*. (Sử dụng nút New Folder nếu muốn tạo Folder mới).

Bước 6: Nhấn OK.

4.3.8. Download một tập tin

Bước 1: Click vào liên kết đến tập tin cần download, IE sẽ mở hộp thoại File Download



Hình 2.25

Bước 2: Chọn mục **Save this file to disk**, và nhấn nút **OK**.

Bước 3: Trong hộp thoại **Save as**, chọn thư mục chứa tập tin và nhấn nút **Save**.

Lưu ý:

- Bạn nên cẩn thận khi download một tập tin trên Internet về. Tốt nhất bạn nên download các tập tin từ các web site đáng tin cậy để tránh “rủi ro” bị virus.

- Nếu bạn chọn mục *Open this file from its current location* có nghĩa là bạn đồng ý “mở nó” ngay lúc đó, điều này rất nguy hiểm trong trường hợp tập tin này bị nhiễm virus. Do đó bạn nên lưu nó trên máy trước, sau đó dùng một trình quét virus để kiểm tra. Nếu không có vấn đề gì mới mở nó.

4.4. THIẾT LẬP CÁC TÙY CHỌN CƠ BẢN

Để sử dụng có hiệu quả Internet Explorer, cần phải đặt các thông số cho trình duyệt trong hộp thoại **Internet Options** (vào trình đơn **Tools** và chọn **Internet Options**).

4.4.1. Thay đổi trang mặc định

Đây là trang mà trình duyệt sẽ tự động mở mỗi khi được khởi động. Cách thực hiện như sau:

- Mục *Address*: Gõ vào địa chỉ của trang mặc định.
- Nút *Use Current*: Chọn trang đang xem là trang mặc định.
- Nút *Use Blank*: Sử dụng một trang trắng làm trang mặc định.




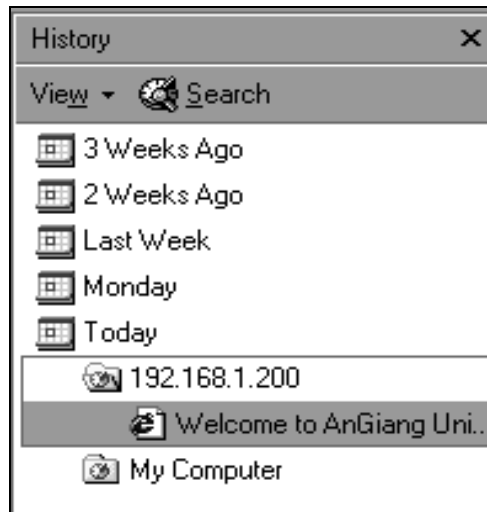
Hình 2.26 Hộp thoại Internet Options

4.4.2. Xem lại các trang web vừa mới viếng thăm (mục History)

Trong khi truy cập các trang web, Internet Explorer lưu lại các địa chỉ bạn đã truy cập thành một danh sách giúp cho việc truy cập nhanh đến các trang này.

Bạn có thể chỉ định giới hạn thời gian lưu trữ các tập tin tạm trong thời gian bao nhiêu ngày bằng cách thay đổi số ngày trong ô *Days to keep pages in history* của khung *History* (hình 2.27) hoặc xoá danh sách này (nếu muốn) bằng cách nhấp chuột vào nút lệnh *Clear History*.

Để tìm một trang lưu lại trong danh sách History, trên thanh công cụ bạn nhấp chuột vào nút  để mở khung History (hình 2.28) ở bên phải khung duyệt. Theo mặc nhiên, danh sách History sẽ được lưu trữ theo thứ tự thời gian từ xa đến hiện tại. Ví dụ: Cách đây 3 tuần (3 Weeks Ago), tuần rồi (Last week), ngày hôm nay (Today),... Trong khung này bạn chọn ngày và nhấp vào địa chỉ trang web cần truy cập (nếu có).

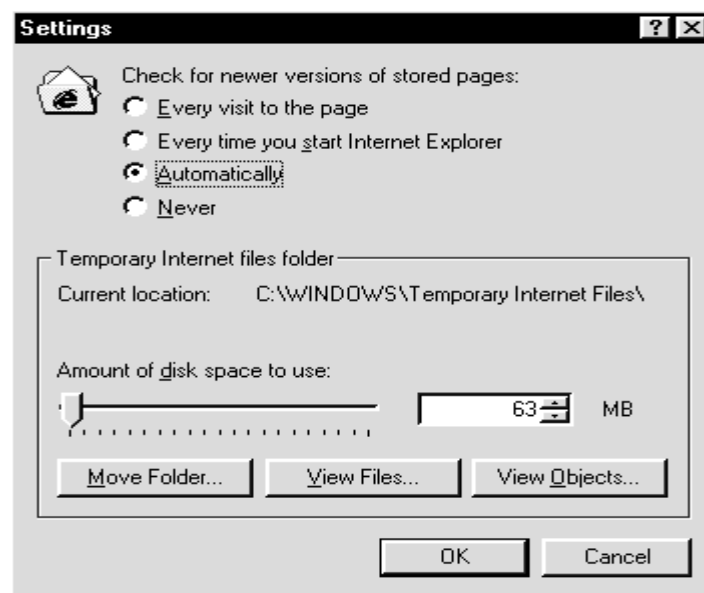


Hình 2.27

4.4.3. Xác định dung lượng đĩa cứng dùng để lưu các tập tin tạm

Thông tin trên các trang web được tải về và lưu trữ tạm thời trên đĩa cứng khi bạn duyệt qua. Điều này giúp tăng tốc độ truy cập đối với các trang web mà bạn thường đến thăm và đồng thời bạn có thể xem lại các trang Web đã xem qua mà không phải nối kết vào mạng.

Để thiết lập dung lượng đĩa cứng để lưu trữ tạm thời các trang web, bạn nhấp vào nút lệnh *Settings* (Hình 2.27) để mở hộp thoại *Settings* (Hình 2.29), sau đó bạn dùng chuột để di chuyển con chạy ở mục *Amount of disk space to use* hoặc nhập vào số chỉ dung lượng (tính bằng MB) ở hộp nhỏ bên phải con chạy và nhấp **OK** để lưu lại thiết lập của bạn.



Hình 2.28

4.5. GIỚI THIỆU DỊCH VỤ E-MAIL

4.5.1. Các khái niệm

- **Thư tín điện tử (E-mail - Electronic mail):**

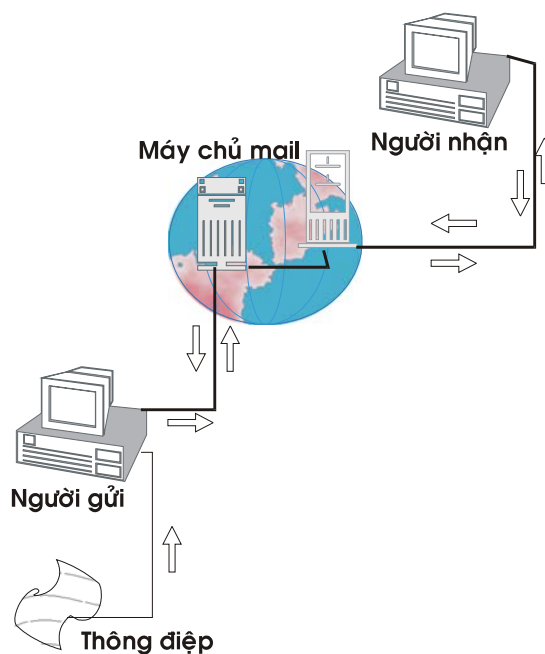
E-mail là một dịch vụ cho phép người sử dụng có thể chuyển và nhận các thông điệp với nội dung và phạm vi không giới hạn, thông qua mạng Internet và các phần mềm quản lý E-mail.

□ **Thông điệp (Message):**

Là những thông tin mà người sử dụng cần trao đổi với nhau. Thông điệp được chuyển vào máy tính thông qua các thiết bị nhập như: bàn phím, máy quét,... và được máy vi tính chuyển thành dạng thông tin đặc biệt để có thể truyền đi thông qua các hệ thống vô tuyến hoặc hữu tuyến.

□ **Mail server:**

Là trung tâm điều khiển và quản lý dịch vụ thư tín điện tử (có thể xem như là trung tâm bưu điện ngoài thực tế). Khi người gửi chuyển một thư tín đi, thư này sẽ được chuyển đến máy chủ quản lý thư và sau đó máy chủ Mail sẽ chuyển đến đúng địa chỉ của người nhận được ghi trong thư thông qua hệ thống mạng máy tính.



Hình 2.29 Chu trình gửi-nhận E-mail

□ **Người gửi (Sender):**

Là người trực tiếp muốn thông tin của họ được chuyển đến người khác thông qua một chương trình chuyển nhận thư tín điện tử (E-mail Program) được cài đặt trong máy tính.

Trên thực tế có rất nhiều loại chương trình cho phép người dùng gửi và nhận thư bằng máy tính như: Outlook Express, Pegasus Mail, Netscape Messenger,... là những chương trình phải cài đặt vào máy; hoặc các web mail không phải cài lên máy, chỉ sử dụng thông qua trình duyệt web như: Yahoo mail, Hotmail, Vol.vnn.vn mail, FPT mail ... Tùy

theo thị hiếu của người dùng và tính năng của từng chương trình mà người dùng có thể chọn cho mình một chương trình thích hợp.

□ **Người nhận (Receiver):**

Là đối tượng mà người gửi muốn chuyển thông tin đến thông qua chương trình chuyển và nhận thư tin như đã nói trên.

□ **Tài khoản (Account):**

Tài khoản được xem như là một nơi chứa các thư tin bạn gửi đi và nhận về, hay xem như là một cái tủ. Chìa khóa để mở tủ này chính là tên của tài khoản (Username) và mã số tài khoản mà nhà cung cấp dịch vụ Internet đã cấp cho bạn (hay được gọi là “mật khẩu” - Password).

□ **Cấu tạo của địa chỉ E-mail:**

ĐỊA CHỈ E-MAIL = TÊN ĐẠI DIỆN HỘP THƯ + @ + TÊN CỦA MÁY CHỦ MAIL

Vi dụ: teache@freemail.agu.edu.vn, agu@hcm.vnn.vn, webadmin@agu.edu.vn, hv01@agu.edu.vn, charles@yahoo.com,...

4.5.2. Cấu tạo của E-mail

From	Gửi từ	Tên và địa chỉ người gửi.
To	Gửi đến	Địa chỉ người nhận. Nếu gửi cho nhiều người thì các địa chỉ cách nhau bởi dấu phẩy hoặc dấu chấm phẩy.
Cc (Carbon copy)	Đồng gửi	Địa chỉ của người đồng nhận với các địa chỉ trong hộp To. Cũng giống như To, nếu có nhiều địa chỉ thì mỗi địa chỉ cách nhau bởi dấu phẩy hoặc dấu chấm phẩy.
Bcc (Blind Carbon Copy)	Đồng gửi ẩn danh	Giống như Cc, chỉ khác là các địa chỉ được liệt kê ở đây sẽ không xuất hiện trong phần tiêu đề thông báo khi người nhận nhận được.
Subject:	Chủ đề	Chủ đề của bức thư.
Attachment:	Kèm file	Những tập tin gửi kèm theo thư.
Contents	Nội dung	Nội dung của thư muốn gửi đi

4.5.3. Các điều kiện cần thiết để sử dụng dịch vụ E-mail

Bước 1: Có một máy vi tính hoạt động được nối với Internet

Bước 2: Có một chương trình gửi/nhận thư tin được cài đặt sẵn trong máy, hoặc một trình duyệt web để sử dụng web mail.

Bước 3: Có một tài khoản E-mail.

4.5.4. Giới thiệu Web Mail

4.5.4.1. Giới thiệu

Web mail là trình gửi nhận thư điện tử được tích hợp trên một trang web của một website nào đó.

Khi sử dụng web mail máy tính chỉ cần có một trình duyệt web (IE, Netscape, hay bất kỳ trình duyệt nào hỗ trợ tải file và hình) và có kết nối Internet.

Tất cả các tác vụ liên quan đến thư như đọc, viết và gửi thư đều được thực hiện trong trang web của nhà cung cấp dịch vụ. Tất cả các email đều được lưu và quản lý trên server của nhà cung cấp dịch vụ e-mail.

❖ Những lợi ích của Web mail:

Bước 1: Miễn phí: Hầu như các web mail trên Internet đều miễn phí.

Bước 2: Có khả năng truy cập ở bất kỳ nơi nào: Có thể truy cập Internet và có trình duyệt là có thể sử dụng hộp thư.

Bước 3: Dễ học và sử dụng: Bạn không phải lo lắng về những thiết lập server và không cần cài đặt client e-mail. Chương trình e-mail được trình bày sẵn, do nhà cung cấp web mail thiết kế, thống nhất trên mọi máy tính và mọi hệ điều hành.

❖ Nhược điểm của Web e-mail:

Bước 1: Có một số hạn chế cho tài khoản mail: Kích thước tối đa của hộp thư, số lượng người nhận thư, kích thước tối đa của thư đến và thư đi, kích thước của file đính kèm,...

Bước 2: Tốc độ không ổn định: Vì web mail dựa trên trình duyệt và thường chứa nhiều hình ảnh (bắt buộc phải tải xuống cả quảng cáo), số lượng người sử dụng lại rất đông vì vậy tốc độ của nó thường không ổn định.

Bước 3: Thời gian truy cập bị giới hạn: Các nhà cung cấp web mail sẽ xoá các tài khoản không dùng đến trong thời gian nhất định. Do đó bạn phải truy cập thường xuyên để duy trì hoạt động của tài khoản.

4.5.4.2. Cách gọi/mở một web mail

Điều kiện: đã kết nối Internet

Bước 1: Mở trình duyệt web.

Bước 2: Nhập địa chỉ của web mail vào hộp Address (chính là địa chỉ của trang web trên đó có chạy chương trình e-mail).

Ví dụ: Yahoo mail: <http://mail.yahoo.com.vn>, Gmail: <http://www.gmail.com>.

Bước 3: Nhập tên tài khoản (username) và mật khẩu (password) để truy cập hộp thư (nếu đã có tài khoản) hoặc đăng ký một tài khoản mail mới nếu chưa có tài khoản trên web mail đó (tùy theo web mail mà bạn phải cung cấp những thông tin cá nhân để tạo được tài khoản).

BÀI TẬP

BÀI 1:

1/. Chọn tập tin HOME.JPG trong thư mục C:\Windows làm ảnh nền của desktop (chọn kiểu hiển thị là Stretch).

2/. Trên Desktop hãy tạo một folder có tên là <họ và tên của bạn>.

3/. Mở folder mang tên của bạn (đã tạo ở trên), chỉnh kích thước của cửa sổ của folder này sao cho bằng khoảng $\frac{1}{4}$ của màn hình. Sau đó di chuyển cửa sổ này sao cho nó không che khuất các biểu tượng trên Desktop.

4/. Xóa folder mang tên của bạn.

5/. Vào Recycle Bin phục hồi lại folder vừa bị xóa.

BÀI 2:

1/. Dùng chức năng tìm kiếm tập tin và thư mục (Start\Search\For Files or Folders) của Windows, tìm và tạo shortcut trên màn hình nền cho các tập tin Winword.exe, Readme.txt, notepad.exe.

2/. Thực hiện các yêu cầu sau:

① Đổi tên các shortcut vừa tạo ở bài tập 1 thành Soanthetaovanban, huongdan và taovanbandongian tương ứng.

② Xóa các shortcut vừa đổi tên.

③ Phục hồi các shortcut vừa xóa.

④ Xóa một lần nữa rồi làm trống thùng rác (Recycle Bin).

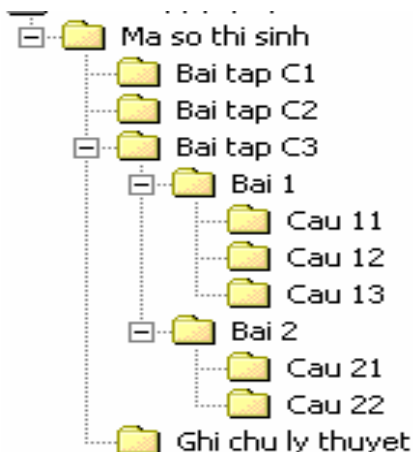
BÀI 3:

① Di chuyển thanh Taskbar về các cạnh màn hình.

② Đặt các thuộc tính: Luôn hiện phía trên các cửa sổ khác, tự động ẩn và hiển thị đồng hồ trên thanh Taskbar.

BÀI 4:

① Tạo cấu trúc folder sau trong ổ đĩa D: hoặc Z:.

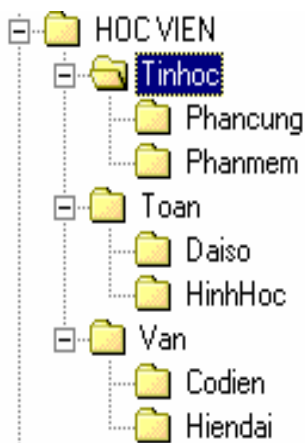


- ② Đổi tên folder Bai tap C1, Bai tap C2 thành BaitapTH, Bai tapUngdung.
- ③ Sao chép folder Bai 1 và Bai 2 vào folder Ghi chu ly thuyet.
- ④ Tạo thêm một số folder mới có tên TH Win98 là folder con của folder Ma so thi sinh.
- ⑤ Di chuyển folder Bai 1 và Bai 2 ở folder Bai tap C3 lần lượt vào hai folder BaitapTH, BaitapUngdung .
- ⑥ Xoá folder Bai tap C3.

Chú ý: Yêu cầu sử dụng nhiều cách khác nhau (sử dụng right-click; nút lệnh trên thanh công cụ; các tổ hợp phím tắt và chọn lệnh trên thanh menu).

BÀI 5:

- ① Tạo cấu trúc folder sau trong ổ đĩa Z: hoặc ổ đĩa D:



② Tạo thêm folder BAITAP là folder con của folder HOCVIEN, sau đó dùng lệnh **Copy** để chép folder Tinhoc và lệnh **Paste** để dán nó vào folder BAI TAP, xem folder BAI TAP.

③ Tạo folder THUCHANH ngang cấp với folder BAITAP rồi dùng lệnh Cut để cắt folder BAITAP và lệnh Paste dán nó vào folder mới vừa tạo, xem folder THUCHANH, BAITAP.

④ Đổi tên folder Phan Cung thành Hardware, Phan Mem thành Software trong folder \HOCVIEN\Tinhoc.

⑤ Xem nội dung folder HOCVIEN, THUCHANH.

⑥ Tạo shortcut cho folder HOCVIEN trên màn hình nền. HD: dùng chuột phải chọn, giữ và kéo nó ra màn hình nền. Xem nội dung folder HOCVIEN trên màn hình nền.

⑦ Xoá Shortcut của folder HOCVIEN trên màn hình nền. Xem lại nội dung folder HOCVIEN trong Windows Explorer.

⑧ Kéo các folder vừa tạo vào thùng rác (Recycle Bin).

⑨ Phục hồi các đối tượng và chọn cùng lúc tất cả folder bạn đã tạo và xoá bằng lệnh Delete.

BÀI 6:

① Tạo cấu trúc folder sau trên ổ đĩa **D:**:



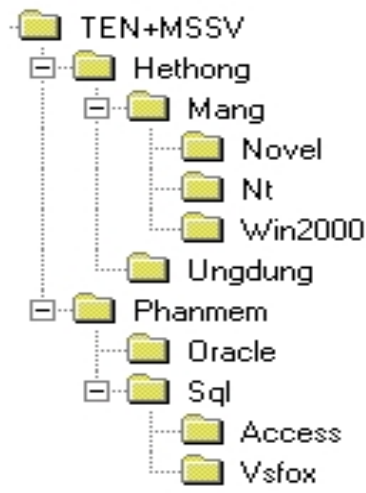
② Tạo folder mang tên của bạn trong ổ đĩa **D:**, sau đó di chuyển folder **THI KIEM TRA** vào folder bạn vừa mới tạo.

③ Đổi tên folder **THI KIEM TRA** thành **THI WINDOWS**.

④ Tạo Shortcut trên màn hình nền cho folder tên của bạn được tạo ở câu ②.

BÀI 7:

① Tạo cấu trúc folder sau trong ổ đĩa **Z:**:



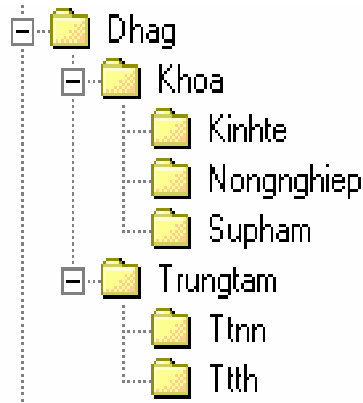
② Đặt thuộc tính chỉ đọc và ẩn cho folders **Hethong**.

③ Di chuyển folder **Ungdung** vào folder **Phanmem**.

④ Di chuyển các folder **Oracle** và **Sql** vào folder **Ungdung**.

BÀI 8:

① Trên ổ đĩa **D:** tạo cấu trúc Folder như hình bên.

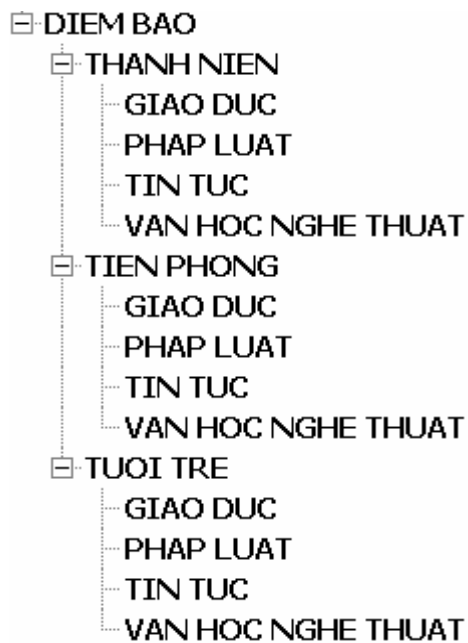


- ② Sao chép folder **Trungtam** vào folder **Supham**.
- ③ Đổi tên các **folder TTNN** thành **CFL**, **TTTH** thành **CICT**.
- ④ Xóa các folder con của folder **Khoa**.
- ⑤ Phục hồi các folder vừa xoá.
- ⑥ Xoá cây thư mục vừa tạo khỏi ổ đĩa.

BÀI 9:

1. Dùng trình duyệt mở trang web có địa chỉ <http://enews.agu.edu.vn>, sau đó lưu nội dung trang web đang mở về đĩa cục bộ thành 2 file - với tên file **tintuc.mht** và **tintuc.html**.
2. Copy nội dung mới nhất trên trang **enews** sang MS Word, rồi lưu lại với tên **tintuc.doc**
3. Mở trang web có địa chỉ <http://www.tuoitre.com.vn>, rồi lưu về đĩa cục bộ một hình ảnh bất kỳ. Sau đó, bạn hãy dùng hình ảnh này để làm ảnh nền cho Desktop.
4. Chọn trang web có địa chỉ <http://www.agu.edu.vn> làm trang chủ (Home page) của trình duyệt.
5. Tạo một folder tên TIN TỨC trong Favorites, rồi lưu những địa chỉ trang web tin tức vào. Ví dụ:
 - a. <http://www.thanhvien.com.vn> đặt tên là BÁO THANH NIÊN
 - b. <http://www.tuoitre.com.vn> đặt tên là BÁO TUỔI TRẺ
 - c. <http://www.dantri.com.vn> đặt tên là BÁO DÂN TRÍ
6. Dùng trang tìm kiếm Google (www.google.com.vn) tìm hình ảnh của một danh nhân mà bạn kính trọng, sau đó lưu 1 hình về đĩa cục bộ.
7. Dùng tài khoản mail của mình gửi đến giáo viên nội dung sau:
 - a. To: <địa chỉ mail của GV đang dạy >
 - b. Chủ đề: <gửi Thầy/Cô>
 - c. Nội dung:
 - i. MSSV
 - ii. Họ và Tên
 - iii. Lớp

8. Dùng tài khoản mail của mình gửi đến giáo viên nội dung sau:
- To: <địa chỉ mail của GV đang dạy>
 - Cc: <địa chỉ mail của 1 bạn trong lớp>
 - Bcc: <địa chỉ mail của 1 bạn khác trong lớp>
 - Chủ đề: <Họ và tên sinh viên>
 - Nội dung:
 - Sở thích hoặc ước mơ sau khi tốt nghiệp bạn sẽ làm gì, ở đâu?
 - Gửi kèm file ảnh đã tìm được ở câu 6

BÀI 10:

- Trên đĩa Z: tạo cấu trúc thư mục như hình bên.
- Trên đĩa **Z:**, tạo tập tin **TRALOI.TXT** có nội dung cho biết:
 - Có bao nhiêu loại phần mềm chính trong hệ thống phần mềm máy tính?
 - Hãy phân loại các phần mềm sau đây theo các loại mà bạn kể trên:
Windows 98, Photoshop, Microsoft Word, Linux.
- Hãy tìm trên máy hai tập tin có phần mở rộng **.ini** và có dung lượng nhỏ hơn **10KB**, chép hai tập tin này về thư mục **DIEM BAO**.
- Trên ổ **Z:** tạo shortcut cho các đối tượng: ổ đĩa **C:**, **My Documents**, **My Computer**.
- Đặt thuộc tính chỉ đọc (Read Only) cho tập tin **TRALOI.TXT** ở trên.

BÀI 11:

- (Z:)
- ▣ isuzu
 - ▣ hi-lander
 - ▣ tropper
 - ▣ x-streme
 - ▣ mitsubishi
 - ▣ grandis
 - ▣ pajero
 - ▣ toyota
 - ▣ altis
 - ▣ innova
 - ▣ vios
- Câu 1:** Tạo cấu trúc thư mục như **Hình 1**. Cây thư mục này sẽ được sử dụng cho các câu hỏi tiếp theo.
- Câu 2:** Tìm trên đĩa **C:** trong máy tính các tập tin có phần mở rộng là **JPG** và có kích thước tối đa **1MB**. Sao chép 1 tập tin bất kỳ trong số các tập tin tìm được sang đĩa **Z:**.
- Câu 3:** Trên đĩa **Z:**, hãy tạo một tập tin dạng **Text Document** có tên **TRALOI.TXT** để trả lời cho 2 câu hỏi sau:
1. Nêu tất cả các cách để có thể kích hoạt cửa sổ tìm kiếm (**Search Results**) trên hệ điều hành Microsoft Windows?
 2. Chỉ rõ một trong các cách để kích hoạt hộp thoại ‘**System Properties**’ như **Hình 2**.

Hình 1**Hình 2**

Câu 4: Trên đĩa **Z:**, hãy tạo shortcut cho 3 đối tượng: **Control Panel**, **Printers and Faxes** và **My Computer**.

PHẦN 3.

MICROSOFT WORD 2003

CHƯƠNG 1

GIỚI THIỆU

1.1. GIỚI THIỆU

Microsoft Word 2003 là một trong những chương trình ứng dụng để xử lý văn bản cao cấp trong bộ Microsoft Office 2003 của hãng Microsoft. Với Word 2003 ta có thể tạo ra một văn bản từ đơn giản đến phức tạp. Word 2003 cung cấp đầy đủ các kỹ năng soạn thảo và định dạng đa dạng, dễ sử dụng. Ngoài ra Word 2003 có thể kết xuất, nhập dữ liệu với nhiều loại định dạng khác nhau, đặc biệt khả năng chuyển đổi dữ liệu giữa Word 2003 và các phần mềm khác trong bộ Microsoft Office làm cho việc xử lý các ứng dụng văn phòng trở nên đơn giản và hiệu quả hơn và dễ dàng kết chuyển tài liệu thành dạng HTML để chia sẻ dữ liệu trên mạng nội bộ cũng như mạng Internet.

1.2. KHỞI ĐỘNG VÀ THOÁT KHỎI WORD 2003

1.2.1. Khởi động Word


C1: Nhấp đúp lên biểu tượng  hoặc  (nếu có) trên màn hình nền.

C2: Vào **Start\Programs\Microsoft Word 2003**.

1.2.2. Thoát khỏi Word

Trước khi thoát khỏi Word (hay bất kỳ chương trình ứng dụng nào khác) cần lưu trữ các văn bản đang làm việc vào đĩa. Nếu không lưu, văn bản sẽ bị mất.

Có thể thoát khỏi Word theo các cách sau:

- **Cách 1:** Nhấp vào nút **Close**  nằm phía bên phải của thanh tiêu đề.
- **Cách 2:** Chọn lệnh **File\Exit** hoặc nhấn tổ hợp phím **Alt + F4**.

Ghi nhớ: Ta luôn nhớ đặt tên cho văn bản ngay khi bắt đầu vào soạn thảo để đề phòng các sự cố mất điện, sự cố về phần mềm, Virus và các sự cố khác làm mất dữ liệu.





1.3. CÁC THÀNH PHẦN CƠ BẢN CỦA WORD



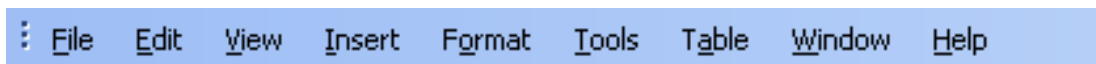
Hình 3. 1 Cửa sổ giao diện của Word

1.3.1. Thanh tiêu đề và tiêu đề (Title bar)



Thanh tiêu đề nằm ở vị trí trên cùng của cửa sổ Word. Thanh tiêu đề chứa biểu tượng , tên tập tin văn bản đang mở và tiêu đề **Microsoft Word** ở bên trái. Bên phải thanh tiêu đề có nút thu nhỏ , nút phục hồi  (Restore Window) và nút đóng cửa sổ  (Close Window).

1.3.2. Thanh lệnh đơn (Menu bar)



Thanh lệnh đơn chứa các lệnh của Word. Các lệnh được bố trí theo từng nhóm tùy theo chức năng như:

- ❖ **File:** nhóm lệnh xử lý tập tin.
- ❖ **Edit:** nhóm lệnh dùng để cập nhật văn bản.
- ❖ **View:** nhóm lệnh dùng để ấn định chế độ hiển thị văn bản.
- ❖ **Insert:** nhóm lệnh dùng để chèn các đối tượng vào văn bản
- ❖ **Format:** nhóm lệnh dùng để định dạng văn bản.
- ❖ **Tool:** nhóm các công cụ hỗ trợ việc soạn thảo văn bản.

- ❖ **Table:** nhóm lệnh xử lý bảng biểu.
- ❖ **Window:** nhóm các lệnh liên quan đến các cửa sổ làm việc của Window.
- ❖ **Help:** nhóm lệnh trợ giúp sử dụng Word.

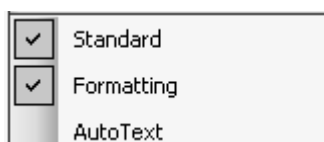
1.3.3. Các thanh công cụ (Toolbar)

Để thuận tiện trong khi soạn thảo văn bản, Word cung cấp sẵn một bộ các thanh công cụ, mỗi thanh công cụ có nhiều nút lệnh để phục vụ cho quá trình soạn thảo. Để biết công dụng của một nút lệnh, trỏ chuột ngay trên nó, khi đó sẽ xuất hiện một hộp màu vàng chứa tên lệnh.

Word 2003 có tất cả 19 thanh công cụ chuẩn, mỗi thanh phục vụ cho một nhóm tác vụ. Do màn hình có giới hạn nên Word không thể bày ra tất cả các thanh công cụ. Tùy theo từng tác vụ mà Word sẽ cho hiển thị thanh công cụ thích hợp phục vụ cho tác vụ đó.

Các thao tác trên thanh công cụ

- **Bật/Tắt các thanh công cụ:**



Chọn lệnh **View\Toolbars**, sau đó chọn thanh công cụ cần bật/tắt. Hoặc click phải chuột vào thanh công cụ bất kỳ danh sách các thanh công cụ sẽ xuất hiện, nhấp chọn thanh công cụ trong danh sách.

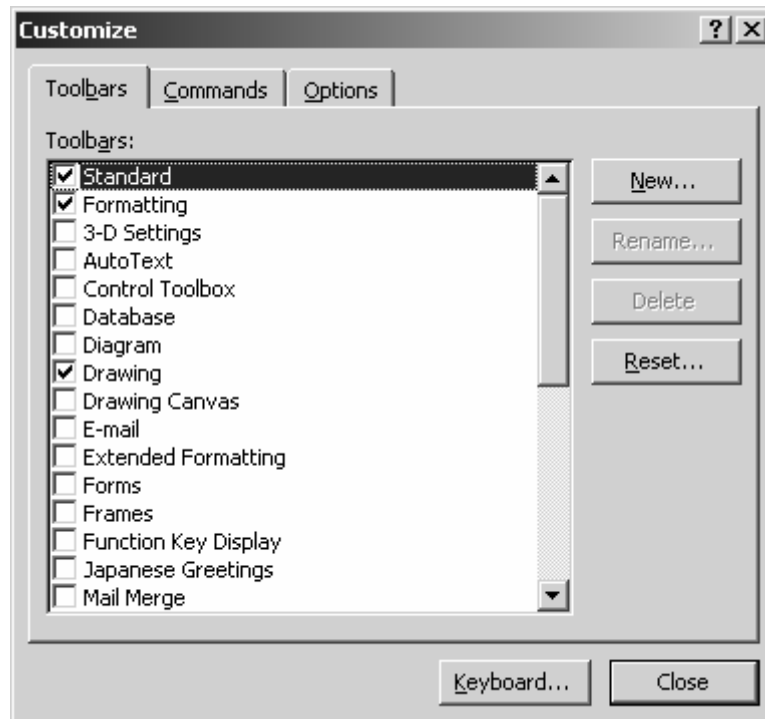
- **Di chuyển các thanh công cụ:**



Muốn di chuyển thanh công cụ đến vị trí khác thì trỏ chuột vào biên của nó không trỏ vào nút lệnh và kéo đến vị trí mới.

- **Tự thiết lập các thanh công cụ (Customize):**

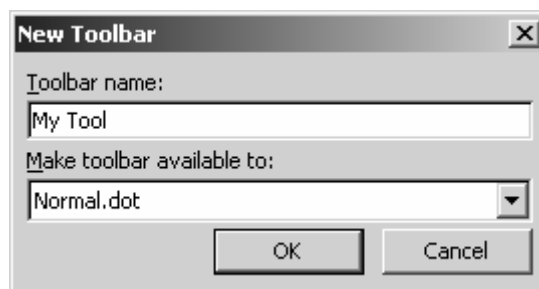
Bên cạnh các thanh công cụ chuẩn có sẵn, Word cho phép người dùng có thể tự thiết kế các thanh công cụ tự chọn để dễ sử dụng và phù hợp với công việc chuyên môn của người sử dụng.


Các bước tạo thanh công cụ mới:**Hình 3.2** Tạo thanh công cụ mới.

Bước 1: Chọn lệnh *View > Customize*

Bước 2: Chọn lớp *Toolbars* sau đó chọn nút *New*.

Bước 3: Trong hộp thoại *New Toolbar* nhập tên thanh công cụ vào hộp *Toolbar name*. Sau đó nhấn **OK**.



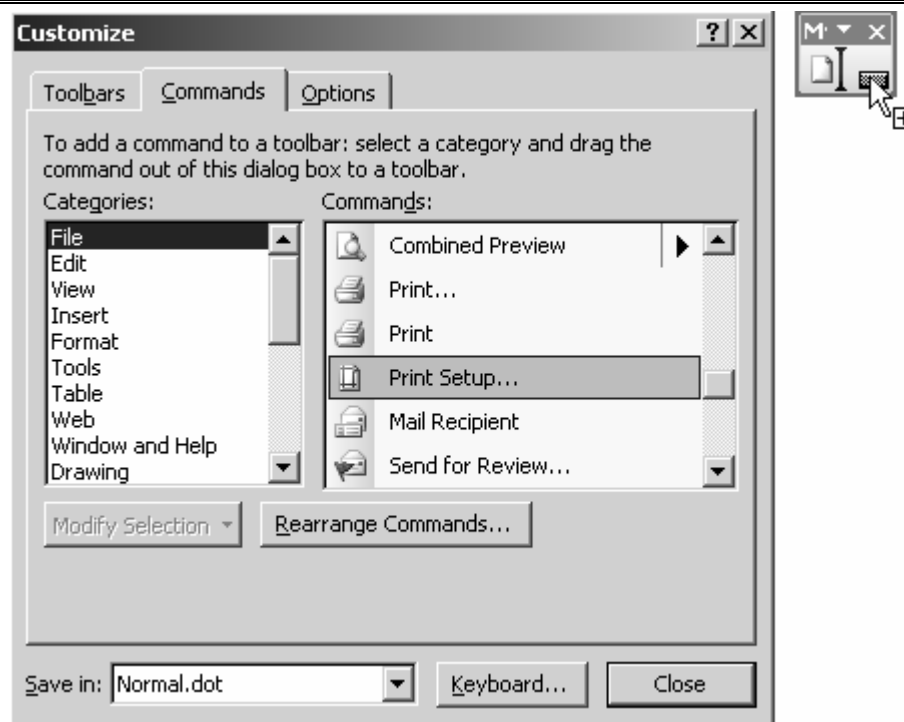
❖ Khi đó thanh công cụ mới được tạo thành có dạng 

Tạo các nút lệnh trên thanh công cụ vừa tạo.

Bước 1: Chọn lệnh *View > Customize*

Bước 2: Chọn lớp *Command* sau đó chọn nhóm lệnh trong danh sách *Categories*. Sau đó gấp gắn lên thanh công cụ như hình 3.3

Bước 3: Nếu muốn gỡ bỏ thì gấp nút lệnh trả về vị trí cũ.



Hình 3.3 Tạo các nút lệnh trên thanh công cụ.

Xóa bỏ thanh công cụ tự tạo

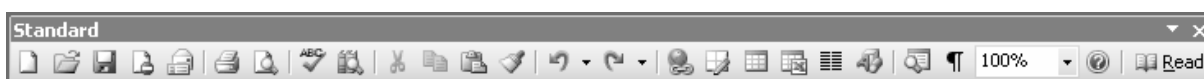
Bước 1: Chọn lệnh *View > Customize*

Bước 2: Chọn lớp *Toolbars* sau đó chọn nút thanh công cụ muốn xóa trong danh sách *Toolbars*.

Bước 3: Nhấp nút *Delete* để xóa. Sau đó Word sẽ đưa thông báo chờ xác nhận, nhấp *OK* để xóa.

- Các thanh công cụ thường dùng:

+ **Thanh công cụ chuẩn (Standard):** Đây là thanh công cụ quan trọng, trên đó chứa các nút lệnh thường sử dụng. Vì vậy nó luôn được hiển thị trên màn hình để tiện sử dụng.



+ **Thanh công cụ định dạng (Formatting):** Thanh công cụ định dạng chứa các nút lệnh dùng cho việc định dạng văn bản như chọn kiểu (Style), font chữ (Font), cỡ chữ (Font size), chữ đậm (Bold), nghiêng (Italic), gạch dưới (Underline), canh lề,...



+ **Thanh công cụ vẽ (Drawing):** Phục vụ cho việc vẽ hình cũng như để tạo các mẫu chữ trang trí, vẽ các text box,... Word cung cấp rất nhiều công cụ tiện lợi. Những nút công cụ này đặt trên thanh công cụ Drawing. Mặc nhiên, thanh công cụ Drawing được hiển thị phía dưới màn hình.

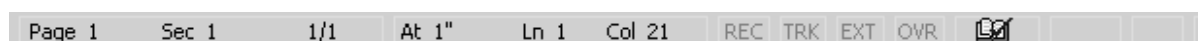


1.3.4. Thước và đơn vị chia trên thước

Thước dùng để kiểm soát các lề, độ lệch so với các lề, điểm dừng của các tab,... Word có hai thước: thước ngang (Horizontal Ruler) nằm phía trên và thước dọc (Vertical Ruler) nằm phía bên trái. Đơn vị của thước có thể là inch hoặc centimeters (cm). Để thay đổi đơn vị chia trên thước dùng tùy chọn: **Tools\Options\General\Measurement units**.

Cũng giống như các thanh công cụ, bạn có thể bật/tắt thước trên màn hình bằng lệnh **View\Ruler**.

1.3.5. Thanh trạng thái (Status bar):



Thanh trạng thái nằm ở đáy của cửa sổ Word. Thanh trạng thái cho biết các thông tin như: trang hiện hành (Page 2), tổng số trang của văn bản hiện hành (2/8), vị trí con trỏ so với mép giấy (ví dụ At 21.1 cm), tọa độ của con trỏ tính theo hàng, cột (ví dụ Ln 31 Col 1), chế độ gõ văn bản là viết chồng hay viết đè (OVR)...

Bật/tắt thanh trạng thái: Chọn lệnh **Tools\Options\View**, trong phần **Show** chọn/bỏ chọn mục **Status bar**.

1.3.6. Thanh cuộn ngang và đứng (horizontal/Vertical scrollbar)

Trên cửa sổ làm việc của Word có hai thanh cuộn: thanh cuộn đứng đặt ở bên phải cửa sổ dùng để cuộn văn bản theo chiều dọc; thanh cuộn ngang nằm ở đáy dùng để cuộn văn bản theo chiều ngang.

Bật/Tắt hiển thị các thanh cuộn:

Bước 1: Chọn lệnh: **Tools \ Options \ View**.

Bước 2: Trong phần **Show** chọn/bỏ chọn **Horizontal scroll bar** để bật/tắt thanh cuộn ngang, chọn/bỏ chọn **Vertical scroll bar** để bật/tắt thanh cuộn đứng.

1.3.7. Vùng soạn thảo văn bản

Vùng soạn thảo văn bản dùng để nhập văn bản vào. Khi nhập văn bản, nếu có từ nào vượt quá lề của văn bản Word sẽ tự động cắt từ đó đem xuống dòng tiếp theo. Muốn chủ động xuống dòng thì phải bấm **Enter** (ngắt đoạn).

Khi nhập văn bản đầy trang, Word sẽ tự động cuộn sang trang mới (ngắt trang mềm). Nếu muốn chủ động qua trang mới trong khi trang hiện hành vẫn còn trống thì gõ tổ hợp phím **Ctrl + Enter** (ngắt trang cứng).


CHƯƠNG 2.

CÁC LỆNH XỬ LÝ TẬP TIN VÀ IN ẤN


2.1. CÁC THAO TÁC TRÊN TẬP TIN

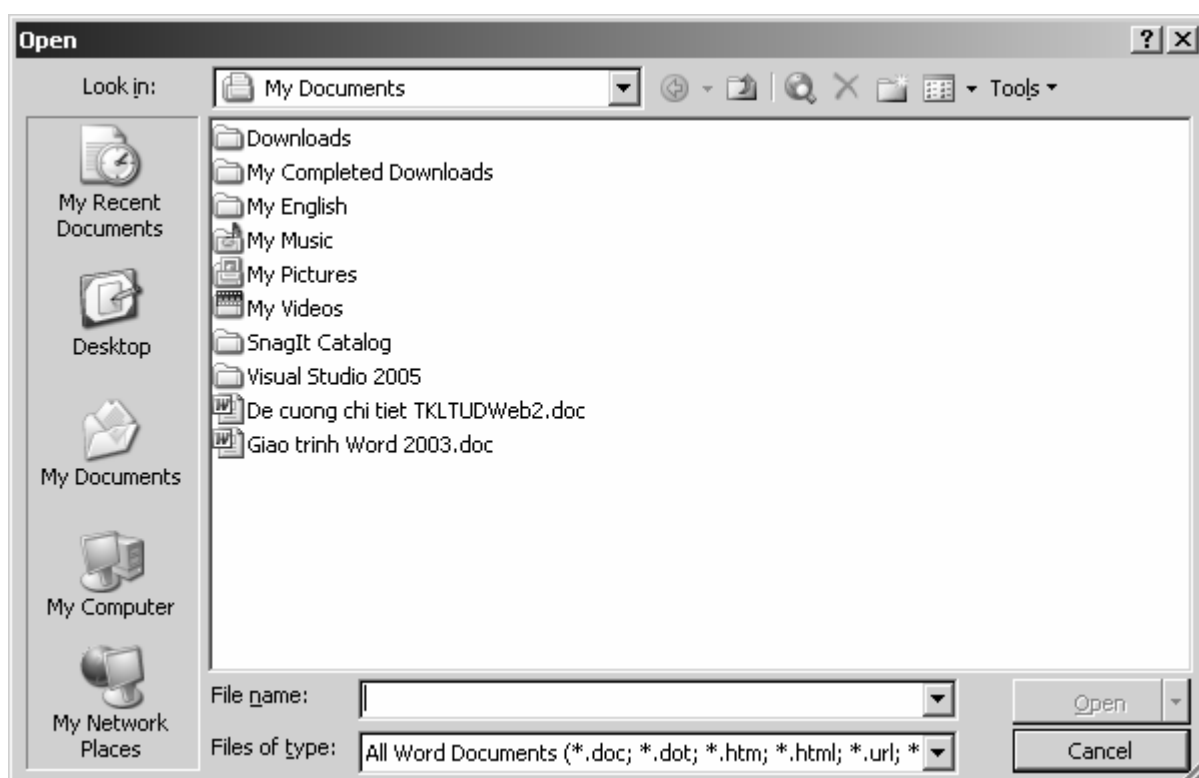
Phần mở rộng mặc nhiên của tập tin Word là **.DOC (Document)**.

2.1.1. Mở tập tin mới

Lệnh **File\New** (nút công cụ , phím gõ tắt **Ctrl+N**) dùng để mở một tập tin mới để bắt đầu soạn thảo.


2.1.2. Mở tập tin đã có trên đĩa

Lệnh **File\Open** (nút công cụ , phím gõ tắt **Ctrl+O**) dùng để mở tập tin đã có sẵn trên đĩa. Bạn cần phải chỉ định vị trí của tập tin (ổ đĩa, đường dẫn) và tên tập tin cần mở trong hộp thoại **Open** như minh họa trong hình sau trước khi click nút lệnh **Open**.



Hình 3.4 Hộp thoại mở một tập tin mới

2.1.3. Lưu tập tin vào đĩa

Lệnh **File\Save** (nút lệnh , phím gõ tắt **Ctrl+S**) dùng để lưu trữ tập tin hiện hành vào đĩa. Khi chọn lệnh này có hai trường hợp xảy ra:

Bước 1: Nếu tập tin chưa có tên thì phải đặt tên. Word sẽ cho hiển thị hộp thoại **Save as**, bạn cần phải xác định thư mục chứa tập tin và đặt tên tập tin trong hộp thoại này.

Bước 2: Nếu tập tin đã có tên (nghĩa là đã có lưu ít nhất một lần) Word sẽ tiến hành lưu lại nội dung đang có mà không hỏi gì thêm.



Hình 3.5 Hộp thoại lưu lại với tên mới

2.1.4. Lưu tập tin vào đĩa với tên khác

Lệnh **File\Save as** dùng để lưu tập tin hiện hành vào đĩa với tên khác. Cách thực hiện giống như khi lưu tập tin chưa đặt tên.

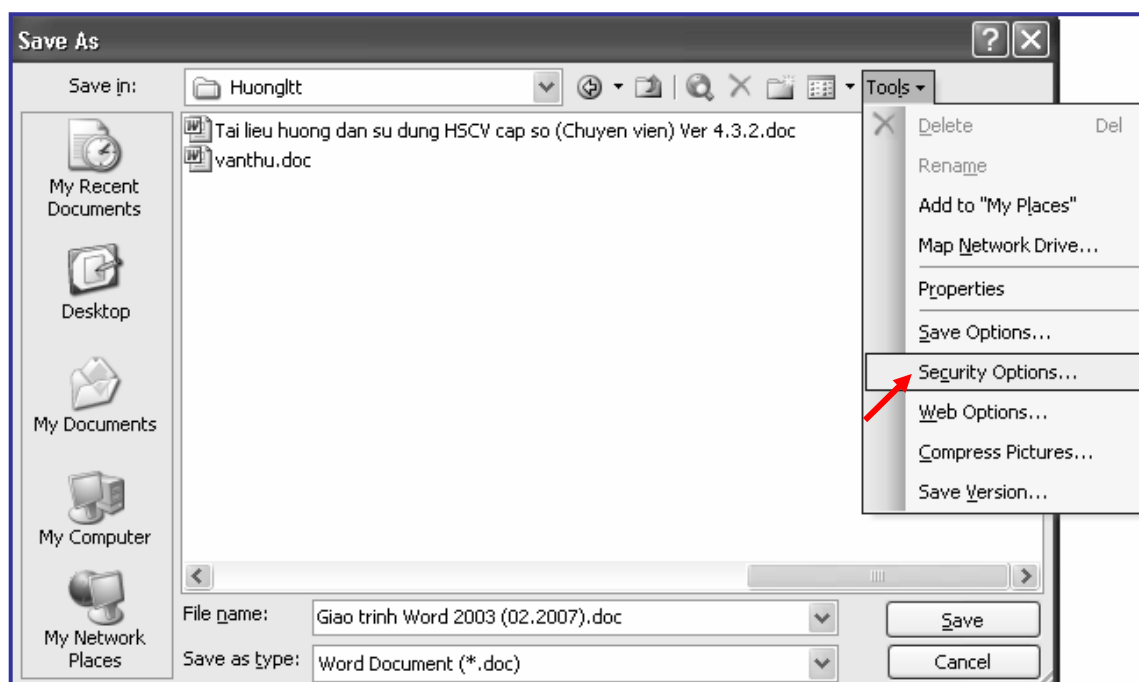
- **Bảo vệ tập tin bằng mật khẩu**

Bước 1: Chọn **File > Save** nếu tập tin chưa được lưu. **File > Save As** nếu tập tin đã lưu.(Hình 3.6)

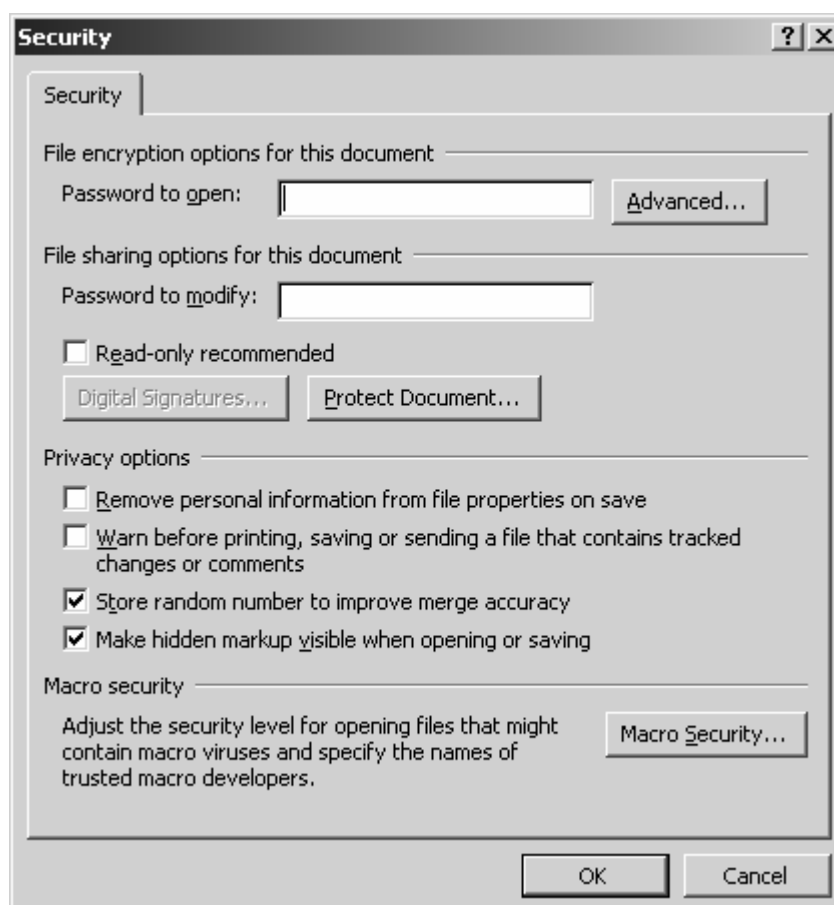
Bước 2: Chọn **Tools > Security Options** (Hình 3.7).

Bước 3: Nhập mật khẩu vào hộp **Password to open** (Hình 3.8)

Bước 4: Chọn **OK** và gõ lại mật khẩu lần nữa.



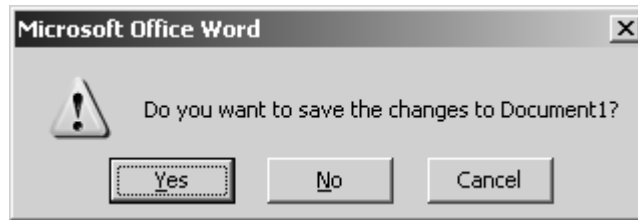
Hình 3.7 Bảo mật tập tin bằng mật khẩu



Hình 3.8 Nhập mật khẩu bảo vệ

2.1.5. Đóng tập tin

Lệnh **File\Close** dùng để đóng tập tin hiện hành. Trước khi đóng tập tin cần phải lưu các cập nhật vào đĩa (nếu cần lưu trữ). Nếu đóng tập tin mà trước đó có cập nhật nhưng chưa lưu, Word sẽ đưa ra cảnh báo sau chờ xác nhận.



Hình 3.6

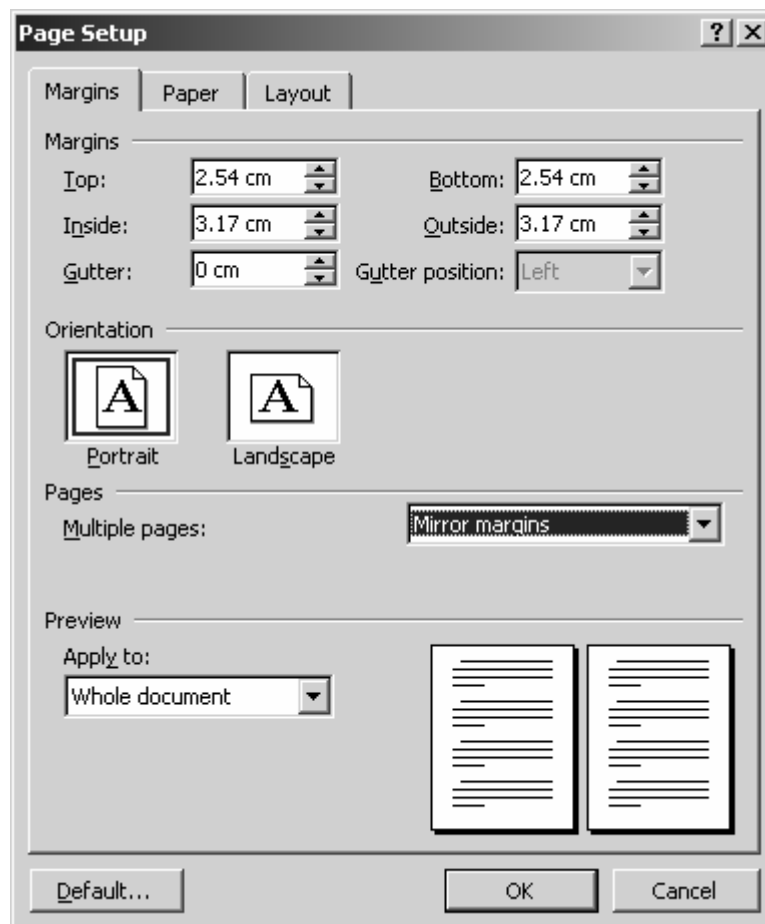
- **Yes**: Lưu những cập nhật.
- **No**: Không lưu.
- **Cancel**: Hủy bỏ lệnh đóng tập tin.

2.2. IN ẤN

2.2.1. Định dạng trang

Bước 1: Dùng lệnh **File\Page Setup** để mở hộp thoại **Page Setup**.

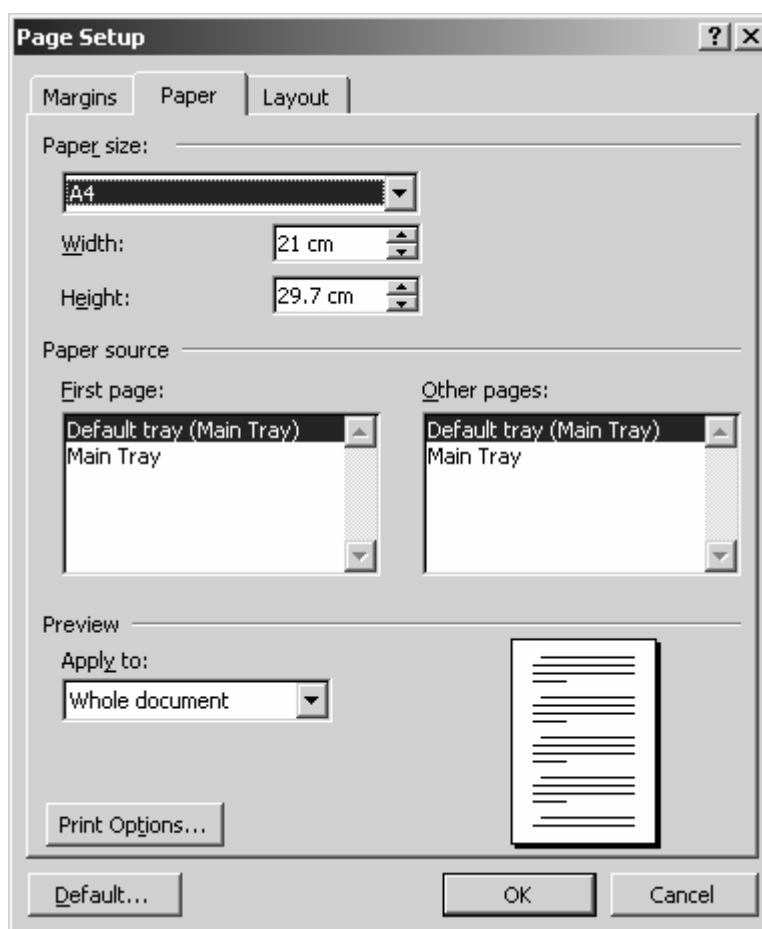
Bước 2: Chọn lớp **Margins** để thiết lập các thông số về lề trang bao gồm:



Hình 3.6 Hộp thoại Page Setup

- **Top/Bottom:** Chừa lề trên/dưới.
- **Left/Right:** Chừa lề trái/phải.
- **Header/Footer:** Chừa từ mép giấy trên đến tiêu đề đầu/cuối trang.
- **Mirror Margins:** dùng trong trường hợp muốn in dữ liệu để đóng thành sách. Khi đóng lại các trang có thể mở ra đối diện nhau. Chọn **Mirror Margins** để lề của hai trang được bố trí đối xứng qua gáy sách.
- **Orientation:** Định hướng in: **Portrait** (in theo chiều dọc) hoặc **Landscape** (in theo chiều ngang).

Bước 3: Chọn lớp **Paper Size** để định khổ giấy sử dụng và hướng in.



Hình 3.7

- **Paper Size:** Chọn khổ giấy. Thông thường là khổ giấy A4 (210x297 mm)

Bước 4: Click **OK**.

Lưu ý: Nếu muốn lưu lại các thiết lập đang chọn làm giá trị mặc nhiên thì bạn click vào nút lệnh **Default** và chọn **Yes**.

2.2.2. Nhập tiêu đề (header) và hạ mục (footer) của trang văn bản

Tiêu đề trang là dòng văn bản ghi ở đầu trang. Hạ mục là dòng văn bản ghi ở cuối trang. Ở các vị trí này chúng ta có thể chèn số trang và các thông tin khác nếu cần.




Bước 1: Chọn lệnh **View\Header and Footer**, khi đó, trên màn hình xuất hiện thanh công cụ **Header and Footer**:



Hình 3.8

Bước 2: Chọn nút  để chuyển qua lại giữa **header** và **footer**.

Bước 3: Nhập dòng văn bản muốn làm tiêu đề/hạ mục. Có thể định dạng văn bản có trong tiêu đề/hạ mục giống như văn bản bình thường.

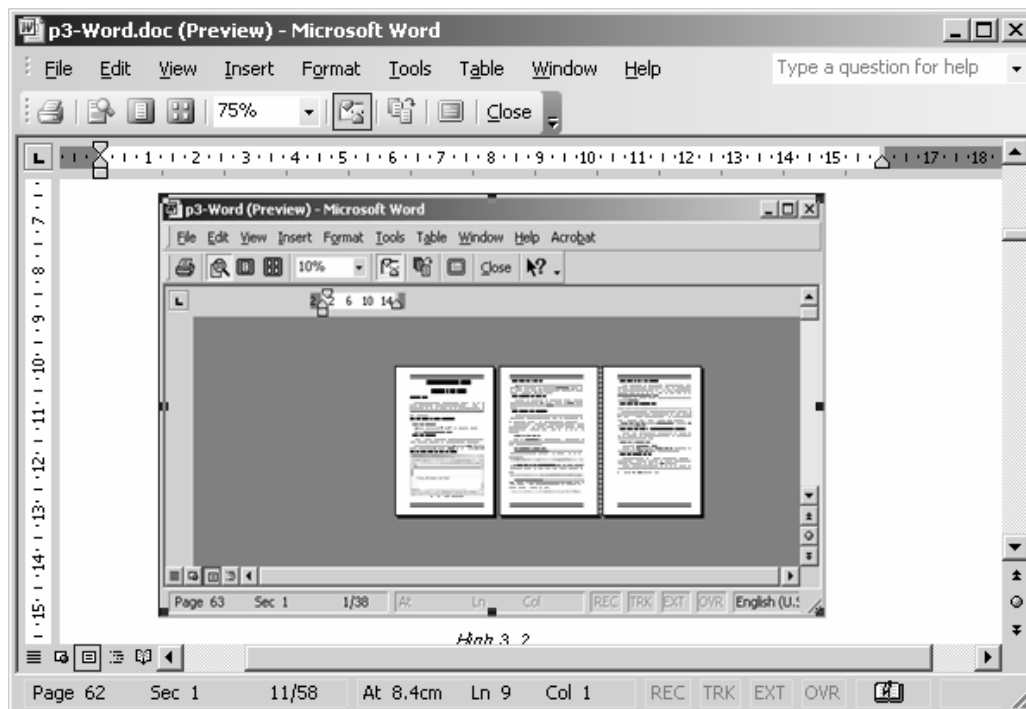
Bước 4: Có thể chọn nút  để điền số trang, nút  để chèn ngày, nút  để chèn giờ vào tiêu đề/hạ mục.

Bước 5: Click nút **Close** để kết thúc việc nhập tiêu đề/hạ mục.

Nếu muốn điều chỉnh lại tiêu đề/hạ mục thì nhấp đúp lên tiêu đề/hạ mục và tiến hành điều chỉnh giống như lúc nhập.


2.2.3. Xem trước nội dung tập tin

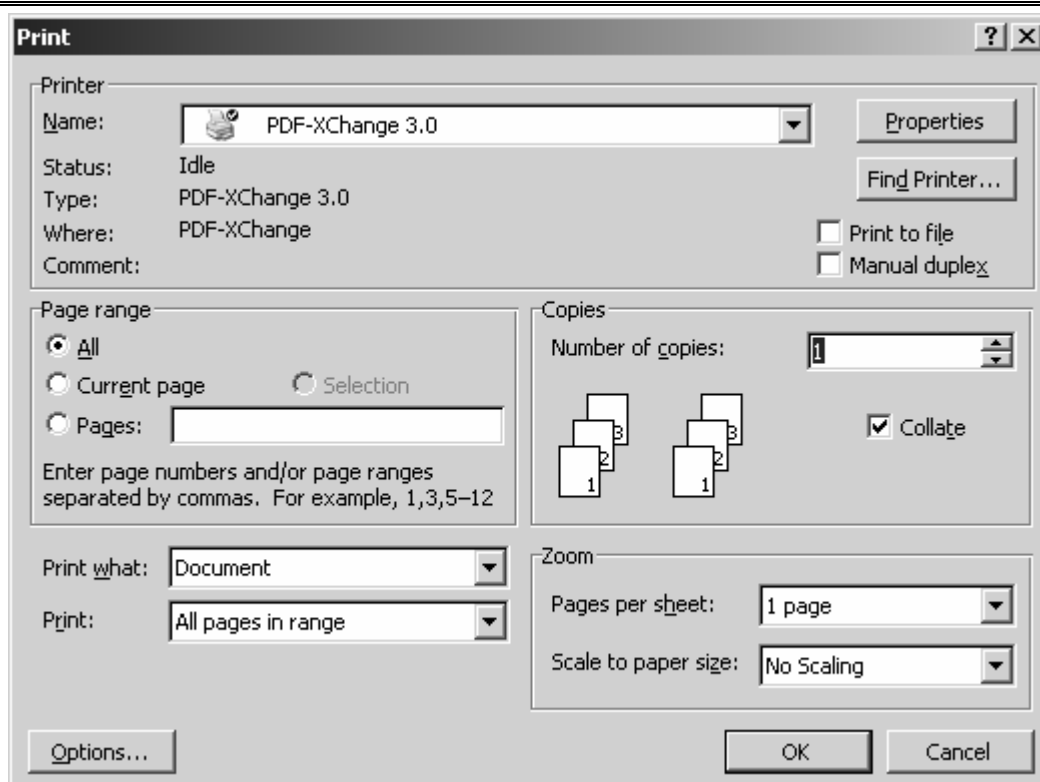
Lệnh **File\Print Preview** dùng để xem trước trang in trên màn hình. Văn bản khi xem ở chế độ này cũng giống như khi in ra giấy.




Hình 3.9 Hộp thoại Print Preview

2.2.4. In nội dung tập tin ra máy in

Dùng lệnh **File\Print**, hoặc click nút công cụ , hoặc dùng tổ hợp phím **Ctrl+P** để in nội dung của văn bản ra máy in.



Hình 3.10 Hộp thoại Print

Nếu nhấn nút  văn bản sẽ được in ra toàn bộ từ đầu đến cuối. Nếu dùng một trong 2 cách còn lại, Word sẽ mở hộp thoại **Print** và bạn có thể thiết lập một số thông số cần thiết sau trước khi click **OK** để in.

- ❖ **Page Range:** xác định phạm vi in.
 - **All:** in tất cả các trang của văn bản hiện hành.
 - **Current page:** chỉ in trang hiện hành (trang đang chứa con trỏ).
 - **Pages:** cho nhập số trang để in. Ví dụ muốn in các trang 1, 2, 3, 4 và 8 thì nhập: 1-6,8.
- ❖ **Print:** Chỉ in trang lẻ (Odd pages), hoặc chỉ in trang chẵn (Even pages) hoặc toàn bộ số trang trong phạm vi đã chọn (All pages in range).
- ❖ **Number of copies:** số bản in, mặc nhiên là 1.

CHƯƠNG 3.

NHẬP VÀ XỬ LÝ VĂN BẢN

3.1. NHẬP VĂN BẢN

3.1.1. Các thành phần của văn bản

- **Ký tự** (character): là các chữ cái, chữ số và các ký tự đặc biệt.
- **Từ** (word): một từ kết thúc bởi một khoảng trống.
- **Câu** (sentence): một câu được kết thúc bởi các ký hiệu: dấu chấm (.), dấu chấm than (!), dấu chấm hỏi (?).
- **Đoạn** (paragraph): một đoạn văn bản kết thúc bởi phím Enter.
- **Trang** (page): một trang kết thúc bởi dấu ngắt trang (page break). Có hai loại dấu ngắt trang:
 - + **Dấu ngắt trang cứng** (hard page) được chèn vào văn bản khi bấm phím Ctrl+Enter.
 - + **Dấu ngắt trang mềm** (soft page) do Word tự động qua trang khi văn bản vượt quá chiều dài trang.

- **Toàn bộ văn bản (document)**

3.1.2. Các chương trình hỗ trợ gõ Tiếng Việt

Để gõ được văn bản Tiếng Việt có dấu, cần phải có một chương trình hỗ trợ gõ Tiếng Việt như Unikey, VietKey, VietSpell,... cài đặt và chạy thường trú trong bộ nhớ máy tính. Ngoài ra, bạn phải chọn bảng mã tiếng Việt phù hợp với font chữ đang sử dụng và biết sử dụng một kiểu gõ dấu Tiếng Việt (VNI, Telex,...).

3.1.2.1. Các bảng mã thường sử dụng

Bảng mã	Font chữ tương ứng
Unicode	Arial, Tahoma, Times New Romans,...
VNI-Windows	Có tên bắt đầu bằng các ký tự VNI-. VD: VNI-Times, VNI-Book,...
TCVN3-ABC	Có tên bắt đầu bằng dấu chấm: .vnTime, .vnArialH,...

3.1.2.2. Các kiểu gõ tiếng Việt

★ **Kiểu gõ VNI:** Dùng các phím số bên trái bàn phím để gõ dấu theo quy ước sau:

Số	Dấu	Số	Dấu
1	Sắc	6	Mũ úp (â, ô, ê)
2	Huyền	7	Móc (ư, ơ)
3	Hỏi	8	Mũ ngựa (ã)

4	Ngã	9	Dấu ngang (Đ, đ)
5	Nặng		

Ví dụ: Sa81c → SắC Huye62n → HuyềN Ho3i → HỏI Nga4 → Ngã
Na85ng → NặNg Hu7o7ng → HươNg




★ **Kiểu gõ TELEX:** dùng các ký tự ít sử dụng để gõ dấu. Qui ước như sau:

Ký tự	Dấu	Ký tự	Dấu
s	SắC	aa, oo, ee	Mũ úp (â, ô, ê)
f	HuyềN	w, uw, ow	Móc (ư, ơ)
r	HỏI	aw	Mũ ngựa (ã)
x	Ngã	DD, dd	Dấu ngang (Đ, đ)
j	NặNg		

Ví dụ: Sawcs → SắC Huyeenf → HuyềN Hoir → HỏI Ngax → Ngã
Nawngj → NặNg Hwowng → HươNg

3.1.2.3. Cách nhập Tiếng Việt bằng chương trình Unikey

Bước 1: Khởi động Word. Chọn font chữ (click nút trên thanh công cụ và chọn một font tùy ý).

Bước 2: Khởi động chương trình Unikey. Nếu có biểu tượng  (đang ở chế độ gõ dấu tiếng Việt) hoặc  (đang tắt chế độ gõ dấu) ở bên phải thanh tác vụ thì chương trình đã được khởi động. Bạn cần phải chuyển sang chế độ gõ dấu Tiếng Việt (click vào biểu tượng ) thì mới có thể gõ dấu được.

Bước 3: Click phải chuột vào biểu tượng chương trình, chọn bảng mã tương ứng font chữ đã chọn.

Bước 4: Tương tự bước 3, chọn kiểu gõ.

Lưu ý:

- Bạn có thể chọn font chữ hay bảng mã trước đều được nhưng phải **tương ứng** như đã trình bày trong phần **3.1.2.1**.
- Nếu sử dụng các chương trình hỗ trợ gõ dấu khác thì bạn cũng thực hiện tương tự như trên.

3.1.3. Nhập văn bản

Khi nhập văn bản, con trỏ tự động dịch chuyển sang phải. Khi con trỏ di chuyển vượt quá lề phải qui định nó sẽ tự động xuống dòng tiếp theo. Muốn kết thúc đoạn thì gõ **Enter**, khi đó Word sẽ tự động chèn vào dấu ngắt đoạn. Khi số dòng vượt quá chiều dài trang thì Word tự động sang trang. Muốn chủ động qua trang khác khi trang hiện hành vẫn còn thì gõ **Ctrl+Enter**.

3.1.3.1. Các phím và lệnh di chuyển con trỏ


Phím	Công dụng
← →	Sang trái, sang phải một ký tự.
↑ ↓	Lên, xuống một dòng.
Home	Về đầu dòng hiện hành.
End	Đến cuối dòng hiện hành.
PageUp	Cuộn lên một trang màn hình.
PageDown	Cuộn xuống 1 trang màn hình.
Ctrl+PageUp	Về trang trước.
Ctrl+PageDown	Đến trang sau.
Ctrl+Home	Về đầu văn bản.
Ctrl+End	Đến cuối văn bản.


Ngoài ra, có thể dùng thanh trượt để cuộn văn bản và click chuột vào vị trí muốn di chuyển đến.

3.1.3.2. Các phím và lệnh xoá

- Phím **Delete**: xoá một ký tự tại vị trí con trỏ, xoá một đối tượng được chọn.
- Phím **Backspace**: xoá một ký tự phía trước (bên trái) con trỏ.
- Lệnh **Edit\Cut** (Ctrl+X): xoá đối tượng được chọn và đưa vào Clipboard.

3.1.3.3. Lệnh Undo và Redo

Lệnh **Edit\Undo** (nút lệnh Undo  hoặc tổ hợp phím Ctrl+Z) cho phép huỷ tác dụng của lệnh vừa thực hiện. Số lần Undo trong Word là không hạn chế cho đến thời điểm lưu tập tin gần nhất.

Lệnh **Edit\Redo** (nút lệnh Redo  hoặc tổ hợp phím Ctrl+Y) có công dụng thực hiện lại các lệnh đã Undo.

3.2. KHỐI VÀ CÁC LỆNH XỬ LÝ KHỐI

Khối là một đoạn văn bản liên tục được xác định bằng bàn phím hay thiết bị chuột. Phần này cung cấp những kỹ năng thao tác trên một khối văn bản bao gồm: sao chép, di chuyển cắt dán nhằm tăng tốc độ soạn thảo văn bản.

3.2.1. Chọn khối

- **Chọn khối bất kỳ:**



- *Dùng bàn phím*: đặt con trỏ ở điểm đầu cần chọn, ấn giữ nút Shift rồi dùng các phím mũi tên để di chuyển đến vị trí cuối cần chọn.

- *Dùng chuột*: trỏ chuột vào đầu khối, ấn giữ chuột trái và kéo đến cuối khối.


- **Chọn từ:** trỏ chuột trên từ và nhấp đúp.
- **Chọn dòng:** trỏ chuột vào đầu dòng muốn chọn, khi chuột có dạng mũi tên ↖ thì nhấp chuột.
- **Chọn câu:** ấn giữ phím Ctrl và nhấp vào vị trí bất kỳ trong câu.
- **Chọn đoạn:** trỏ chuột vào khoảng trống bên trái đoạn và nhấp đúp.
- **Chọn toàn bộ văn bản (Select All):** ấn giữ phím Ctrl nhấp vào đầu dòng bất kỳ, hoặc gõ tổ hợp phím **Ctrl+A**, hoặc chọn lệnh **Edit\Select All**.

3.2.2. Cắt (Cut) hoặc sao chép (Copy) khối đã chọn vào clipboard

Bước 1: Chọn khối muốn cắt hoặc sao chép.

Bước 2: Chọn lệnh **Edit\Cut** (hoặc nút lệnh , hoặc phím gõ tắt **Ctrl+X**) để cắt hoặc **Edit\Copy** (hoặc nút lệnh , hoặc gõ tắt phím **Ctrl+C**) để sao chép.

3.2.3. Dán nội dung trong clipboard vào văn bản (Paste)

Dùng lệnh **Edit\Paste** (hoặc nút lệnh , hoặc tổ hợp phím **Ctrl+V**).

3.2.4. Xóa bỏ đối tượng đang chọn

Muốn xóa bỏ một khối văn bản hay một đối tượng bất kỳ, chỉ cần chọn đối tượng rồi gõ phím **Delete** hoặc **Spacebar**.

3.2.5. Di chuyển các đối tượng

Cách 1:

Bước 1: Xác định khối cần di chuyển.

Bước 2: Cắt văn bản để đưa vào clipboard.

Bước 3: Đặt con trỏ tại vị trí muốn di chuyển đến dán vào.

Cách 2:

Bước 1: Xác định khối cần di chuyển.

Bước 2: Trỏ chuột vào khối vừa chọn và kéo nó đến vị trí mới.

3.3. CÁC KỸ NĂNG ĐỊNH DẠNG VĂN BẢN

Định dạng văn bản là các thao tác giúp cho bạn trang trí làm đẹp văn bản theo ý muốn hoặc theo nhu cầu thực tế.

3.3.1. Định dạng ký tự



Hình 3.11 Hộp thoại Font

Bước 1: Dùng lệnh **Format\Font** (tổ hợp phím **Ctrl+D**) để mở hộp thoại **Font**.

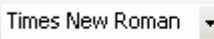




Bước 2: Thiết lập các định dạng sau:

- Chọn font chữ trong hộp Font và màu ký tự trong hộp **Font Color**.
- Chọn kiểu chữ: bình thường (regular), in đậm (Bold), in nghiêng (Italic) trong hộp Font style.
- Chọn kích cỡ ký tự trong các hộp **Size**.
- Chọn kiểu gạch dưới (Underline style) và màu của đường gạch (Underline color).
- Chọn các hiệu ứng đặc biệt trong phần **Effects**.

Bước 3: Click **OK**.

Lưu ý:

- Nếu chọn khối văn bản trước khi định dạng, thì việc định dạng chỉ có tác dụng trên khối văn bản đang chọn. Nếu không chọn khối thì các định dạng sẽ có tác dụng từ vị trí con trỏ trở về sau cho đến khi có một định dạng khác.

- Có thể dùng các nút lệnh    ,  để định dạng nhanh font chữ, cỡ chữ, kiểu chữ (in đậm, in nghiêng và gạch dưới) và màu chữ tương ứng.

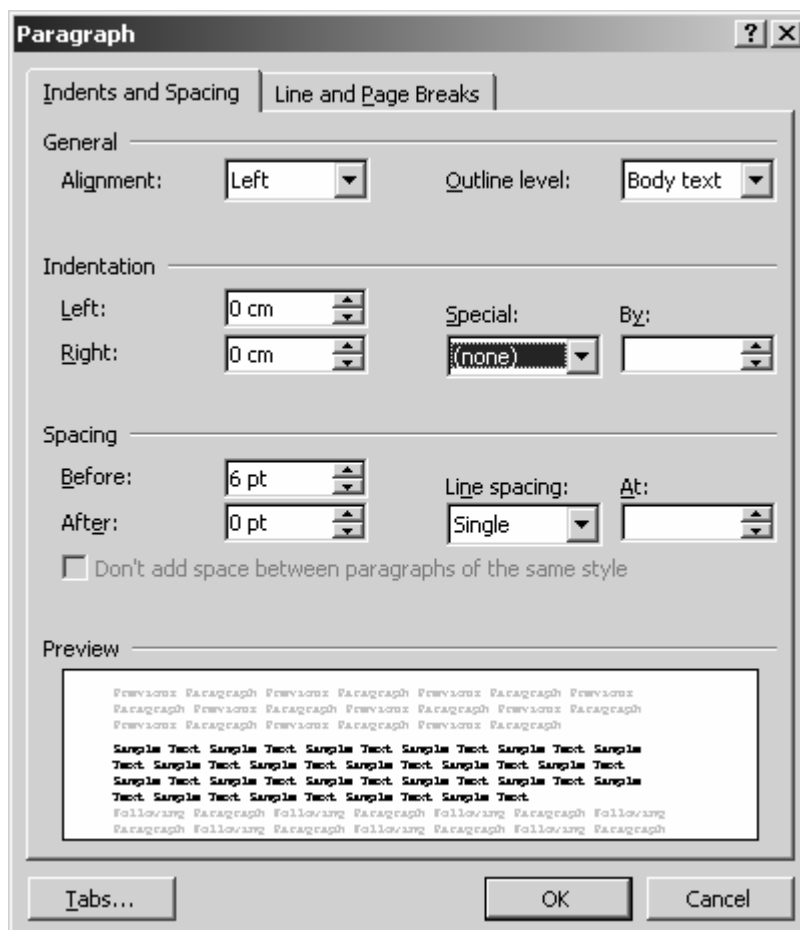
- Có thể dùng các tổ hợp phím tắt sau để định dạng nhanh: Ctrl+B (in đậm), Ctrl+I (in nghiêng) và Ctrl+U (gạch dưới).

3.3.2. Định dạng đoạn văn bản.

Trong phần này chúng ta sẽ tìm hiểu các thao tác định dạng các đoạn văn bản. Mỗi dấu xuống dòng (Enter) sẽ tạo thành một đoạn văn bản. Để mở tính năng định dạng đoạn văn bản thực hiện các bước sau:

Bước 1: Chọn các đoạn muốn định dạng. Nếu muốn định dạng chỉ một đoạn thì chỉ cần đặt con trỏ trong đoạn đó.

Bước 2: Chọn lệnh **Format\Paragraph** để mở hộp thoại **Paragraph**, thiết lập các định dạng sau:







Hình 3.12 Hộp thoại Paragraph

- Chọn chế độ canh lề:
 - Chừa lề trái (**Left**), phải (**Right**) và định dạng dòng đầu tiên (**Hanging**-treo, **First line**-thụt vào, và **None**-bình thường) trong phần **Indentation**.

- Định khoảng cách với đoạn trước (**Before**), sau (**After**) và khoảng cách giữa các dòng trong cùng đoạn (**Line spacing**) trong phần **Spacing**.

Bước 3: Click **OK**.

Các nút công cụ định dạng Paragraph và các phím tắt tương ứng:

	Canh trái (Ctrl+L)		Canh giữa (Ctrl+E)
	Canh phải (Ctrl+R)		Canh đều hai bên (Ctrl+J)

3.3.3. Tạo hoa thị và mục số (Format \ Bullets and Numbering)

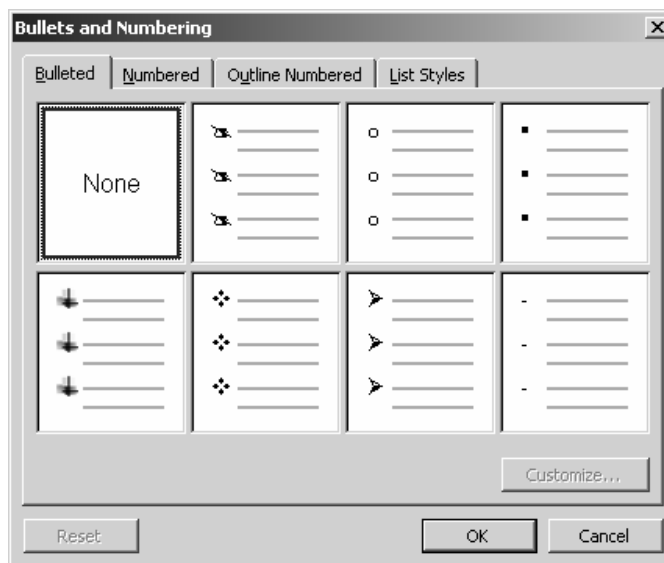
Ở phần này sẽ hướng dẫn các thiết lập đánh dấu đầu đoạn (**Bullets**) và các đánh số chỉ mục (**Numbering**) cho các tiêu đề trên tài liệu Word.

➤ **Thiết lập Bullets.**

Để đánh dấu đầu dòng các đoạn văn bản hãy làm theo cách bước sau đây:

Bước 1: Chọn đoạn muốn đánh dấu đầu dòng.

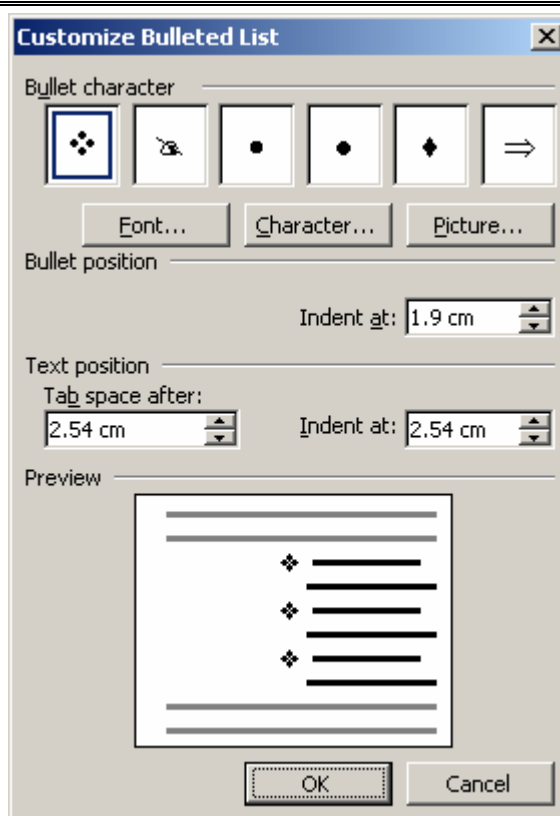
Bước 2: Chọn lệnh **Format\Bullets and Numbering**. Hộp thoại sau đây xuất hiện.



Hình 3.13 a. Hộp thoại thiết lập Bullets


Bước 3: Thiết lập thông tin về dấu đầu dòng ở thẻ **Bulleted** như sau:

- Dùng chuột ấn lên kiểu Bullet muốn thiết lập
- Để chọn một kiểu Bullet khác, nhấn vào nút **Customize**.
- Nhấn vào nút **Picture...** để chọn các hình khác. Sau đó nhấn **OK**



Hình 3.13b . Chọn một dạng Bullets khác

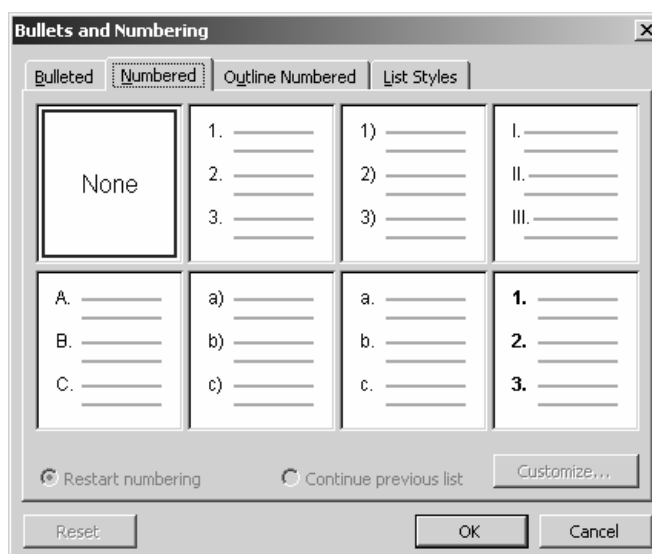
Bước 4: Chọn **OK**.

Lưu ý: Có thể dùng nút lệnh  trên thanh công cụ **Formating**

Thiết lập Numbering

Để đánh số chỉ mục cho các tiêu đề của tài liệu, thực hiện theo các bước sau:

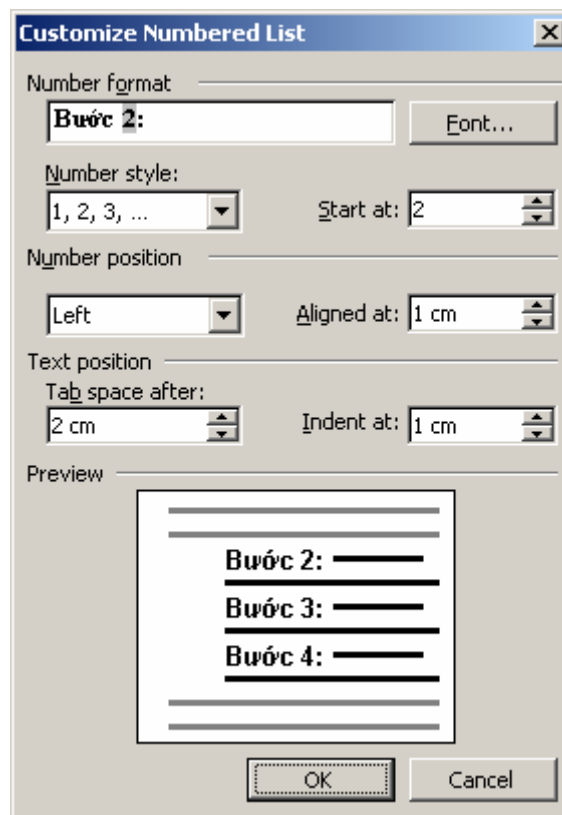
Bước 1: Chọn đoạn cần đánh số chỉ mục. Sau đó chọn **Format/ Bullets and Numbering** hộp thoại sau xuất hiện.



Hình 3.14a . Tạo số chỉ mục

Bước 2: Thiết lập thông tin về đánh số chỉ mục ở thẻ **Numbered** như sau:


- Dùng chuột chọn kiểu đánh số chỉ mục muốn thiết lập.
- Chọn nút Customize để định dạng cho số chỉ mục như hộp thoại sau:



Hình 3.14b . Thiết lập dạng số chỉ mục

Ý nghĩa của các mục tin trên hộp thoại này như sau

- Nút Font cho phép chọn phông chữ cho số chỉ mục
- Mục Start At: Để chọn chỉ mục đầu tiên cần đánh số: Nếu chọn 1, word sẽ thực hiện đánh số chỉ mục từ 1, 2, 3... tương tự nếu chọn là 5 thì sẽ đánh số chỉ mục là 5, 6, 7...
- Mục Number Position để thiết lập vị trí chỉ mục trên văn bản.
- Text position để thiết lập khoảng cách văn bản so với mép lề của trang tài liệu

Lưu ý: ta có thể chọn lệnh  trên thanh công cụ Formatting để đánh số chỉ mục cho đoạn văn bản.

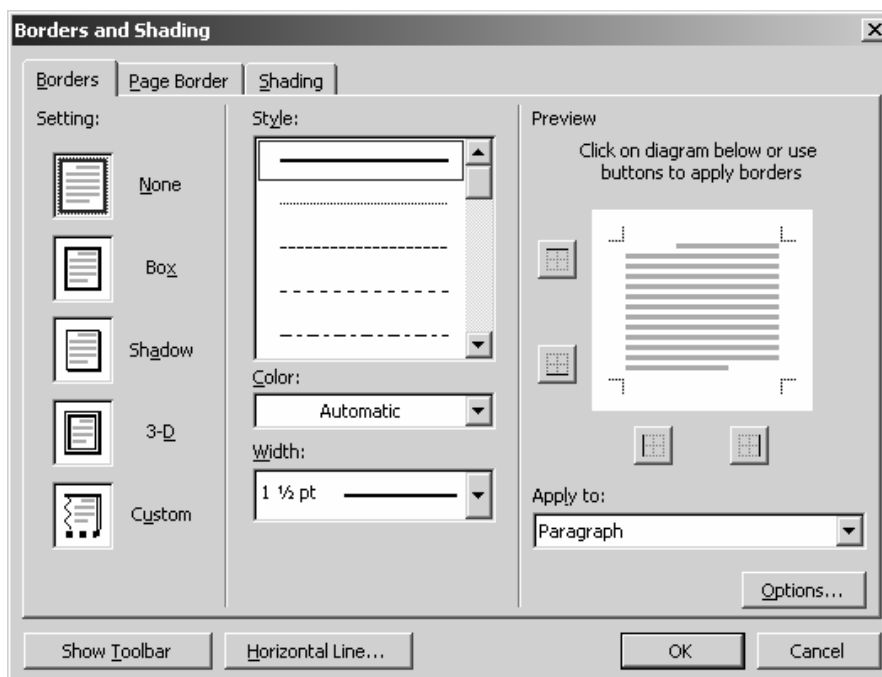
3.3.4. Kẻ khung và tô nền

Bước 1: Chọn các đoạn muốn kẻ khung hoặc tô nền.

Bước 2: Chọn lệnh **Format\Borders and Shading** để mở hội thoại **Borders and Shading**.

Bước 3: Kẻ khung:

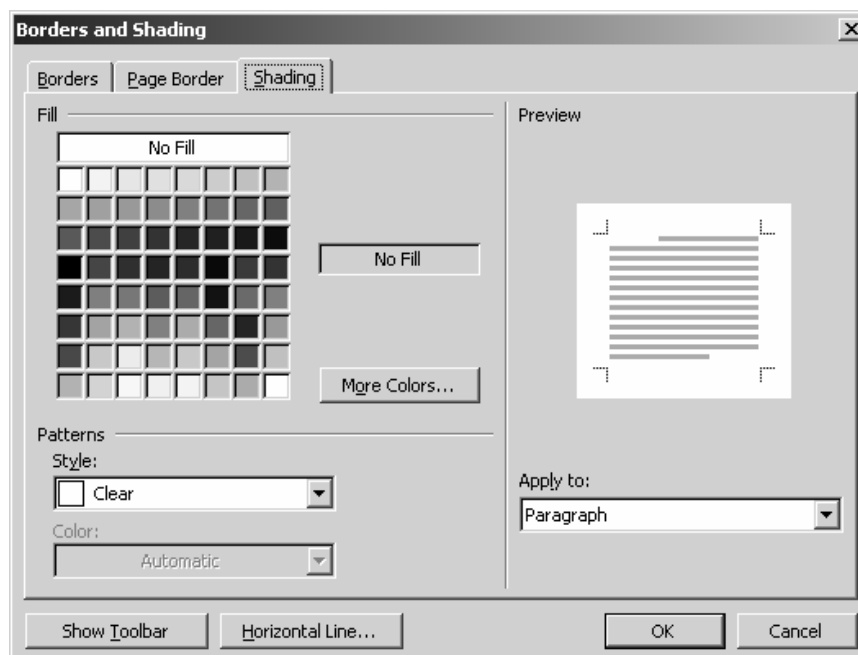
- ❖ Chọn tab **Borders**.
- ❖ Chọn kiểu đường kẻ (**Style**), màu đường kẻ (**Color**), độ đậm nhạt của đường kẻ (**Width**) và chọn các cạnh muốn kẻ khung trong phần Preview.



Hình 3.15 Hộp thoại Borders and Shading (lớp Borders)

Bước 4: Tô nền:

- ❖ Chọn tab **Shading**.
- ❖ Chọn màu nền trong khung **Fill**.



Hình 3.16 Hộp thoại Borders and Shading (lớp Shading)

Bước 5: Nhấp OK.

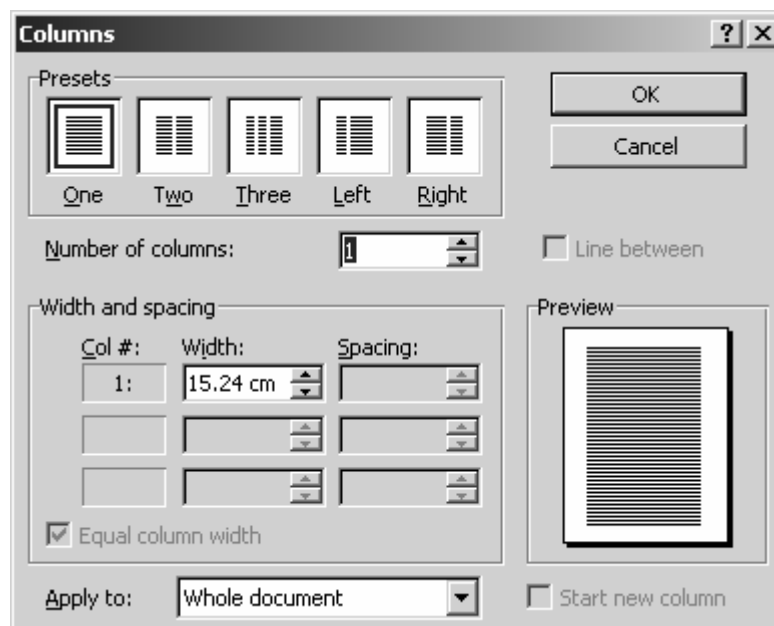
3.3.5. Định dạng văn bản thành nhiều cột

Word cung cấp tính năng **Columns** giúp người dùng dễ dàng chia văn bản thành nhiều cột (giống như các cột của trang báo và tạp chí). Mỗi đoạn văn bản có thể chia thành các cột có độ rộng và kích thước khác nhau. Trên mỗi cột có thể thực hiện các thông tin như bảng biểu, hình vẽ ... như thao tác trên các trang tài liệu bình thường. Lưu ý :*Việc chia cột nên thực hiện sau khi đã thực hiện hoàn chỉnh văn bản.*

Cách bước chia cột văn bản:

Bước 1: Xác định khối văn bản muốn chia cột.

Bước 2: Chọn lệnh **Format\Columns** để mở hộp thoại **Columns**.



Hình 3.17 Hộp thoại **Columns**

Bước 3: Xác định số cột muốn chia trong hộp **Number of columns**; click vào tùy chọn **Line Between** muốn có đường gạch giữa các cột; nếu muốn các cột có độ rộng bằng nhau thì chọn mục **Equal column width**.

Bước 4: Nhấp OK.

3.3.6. Định dạng tab (Format\Tabs)

Tab là công cụ được sử dụng rất nhiều trong việc nhập văn bản. Khi gõ phím Tab thì con trỏ sẽ nhảy và dừng ở những khoảng cách đều đặn (mặc nhiên là 0.5 inch). Đó là các điểm dừng của Tab. Tuy nhiên có thể đặt các điểm dừng của Tab một cách tùy ý.

Các loại tab:

- **Left:** dữ liệu nhập vào được canh về phía trái của tab rồi phát triển sang bên phải của tab (mặc nhiên).
- **Center:** dữ liệu nhập vào được canh giữa tab rồi phát triển sang hai bên.
- **Right:** dữ liệu nhập vào được canh về phía phải của tab rồi phát triển sang bên trái.

- **Decimal**: đóng thẳng hàng các số có dấu chấm thập phân.

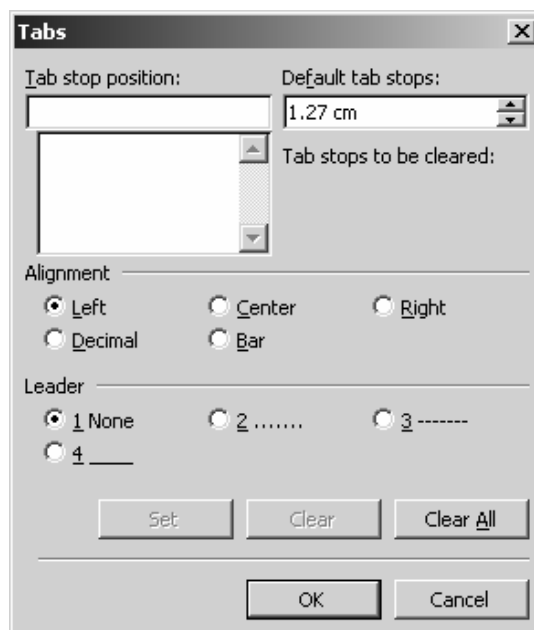
Cách đặt các điểm dừng của Tab:

Cách 1: Dùng lệnh

Bước 1: Chọn lệnh **Format\Tabs** để mở hội thoại **Tabs**.

Bước 2: Gõ vào hộp **Tab stop position** số chỉ chiều dài của tab tính từ lề trái.

Bước 3: Chọn loại tab trong khung **Alignment** và chọn các ký tự lấp đầy các khoảng trống phía trước tab trong khung **Leader** (nếu cần).




Hình 3.18 Hộp thoại **Tabs**

Bước 1: Nhấp trên nút **Set** để đặt điểm dừng của tab cho paragraph hiện hành.

Bước 2: Muốn đặt nhiều điểm tab thì thực hiện lại các bước 2, 3, 4

Bước 3: Nhấp **OK**.

Cách 2: Dùng chuột

Bước 1: Nhấp vào ký hiệu  (nằm ở phía bên trái của thước ngang) để chọn kiểu tab.

Bước 2: Nhấp vào thước ngang ngay vị trí muốn đặt tab. Nếu muốn tạo Leader thì vào **Format\Tab**, chọn tab muốn tạo **Leader**, chọn loại **Leader** và click **Set**.

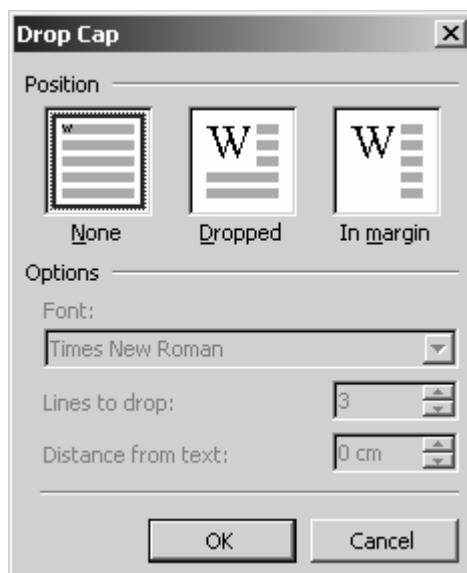
Thay đổi vị trí của tab:

Trỏ chuột vào tab muốn thay đổi vị trí rồi kéo đến vị trí mới.

3.3.7. Tạo Drop Cap

Tính năng **Drop Cap** của word giúp tạo kiểu chữ cái lớn đầu tiên cho một đoạn văn bản. Cách thực hiện định dạng này như sau:

Bước 1: Chọn ký tự muốn tạo **Drop Cap** rồi chọn lệnh **Format\Drop Cap**.



Hình 3.18 Hộp thoại Drop Cap

Bước 2: Chọn **Dropped** trong khung **Position**.

Bước 3: Xác định chiều cao (tính bằng dòng) của ký tự **Drop Cap** trong mục **Lines to drop**. Và click **OK**.

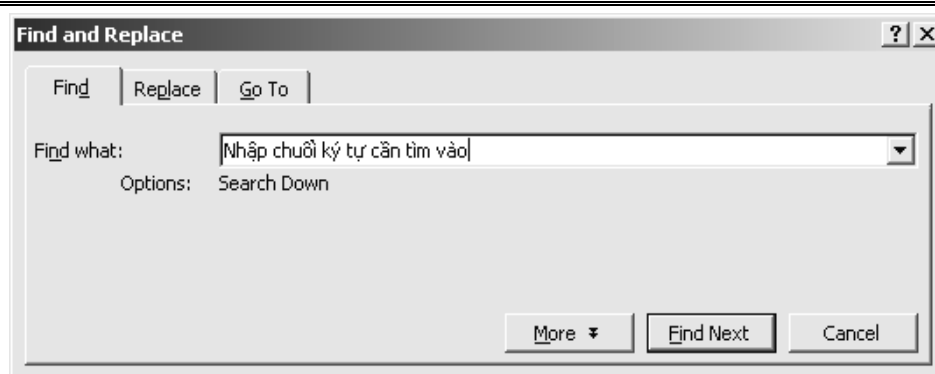
3.4. Các tiện ích của Word

Word cung cấp cho người dùng nhiều tiện ích cho phép chúng ta tìm kiếm, thay thế một đoạn văn bản cũ bởi đoạn văn bản mới hoặc chỉnh sửa lỗi chính tả Điều này rất giúp ích khi bạn làm việc trên một số lượng văn bản rất lớn.

3.4.1. Tìm kiếm văn bản (Edit\Find)

Đây là chức năng tìm một chuỗi ký tự bất kỳ trong văn bản hiện hành. Nếu tìm thấy sẽ định vị con trỏ đầu chuỗi. Ngược lại, Word sẽ đưa ra thông báo.

Bước 1: Chọn lệnh **Edit\Find** hoặc gõ tổ hợp phím **Ctrl+F** để mở hộp thoại **Find and Replace**.



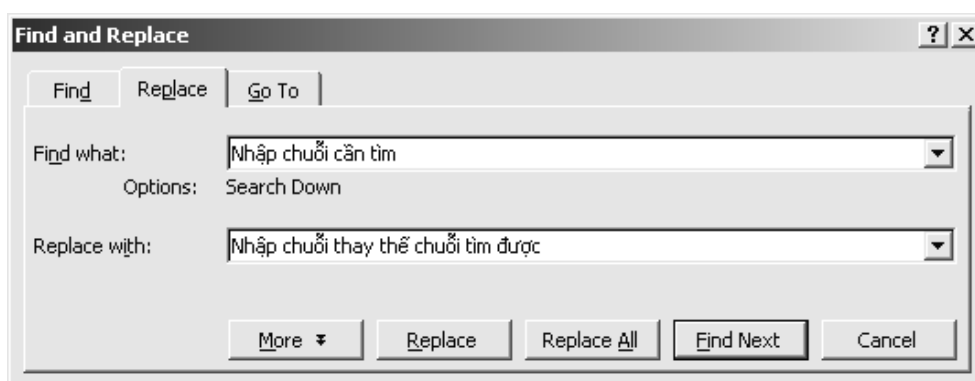
Hình 3.19 Hộp thoại Find and Replace (Chức năng Find)

Bước 2: Nhập chuỗi ký tự cần tìm vào hộp **Find what**.

Bước 3: Nhấp nút **Find Next** để bắt đầu tìm.

3.4.2. Tìm kiếm và thay thế (Edit\Replace)

Bước 1: Chọn lệnh **Edit\Replace** hoặc **Ctrl+H** để mở hộp thoại **Find and Replace**.



Hình 3.20 Hộp thoại Find and Replace (Chức năng Replace)

Bước 2: Nhập chuỗi ký tự cần tìm vào hộp **Find what**.

Bước 3: Nhập chuỗi ký tự để thay thế vào hộp **Replace with**.

Bước 4: Click nút **Find Next** để tìm đoạn văn bản trong ô **Find What**. Nếu tìm thấy con trỏ sẽ định vị tại đầu chuỗi, khi đó bạn có thể chọn **Replace** để thay thế hoặc click **Find Next** để tiếp tục tìm một chuỗi khác. Nút **Replace All** có công dụng tự động tìm và thay thế tất cả các chuỗi cần tìm bằng chuỗi thay thế.

Bước 5: Click nút **Cancel** hoặc nhấn phím **ESC** để đóng hộp thoại.

3.4.3 Tính năng AUTOCORRECT

Đây là tính năng tự động sửa lỗi chính tả rất mạnh của Word. Nó giúp tốc độ soạn thảo văn bản nhanh hơn nhờ những từ viết tắt và tránh được các lỗi chính tả không cần thiết bởi khả năng tự động sửa lỗi chính tả (chỉ chỉnh sửa lỗi chính tả đối với Anh ngữ).

✎ Thêm một từ viết tắt

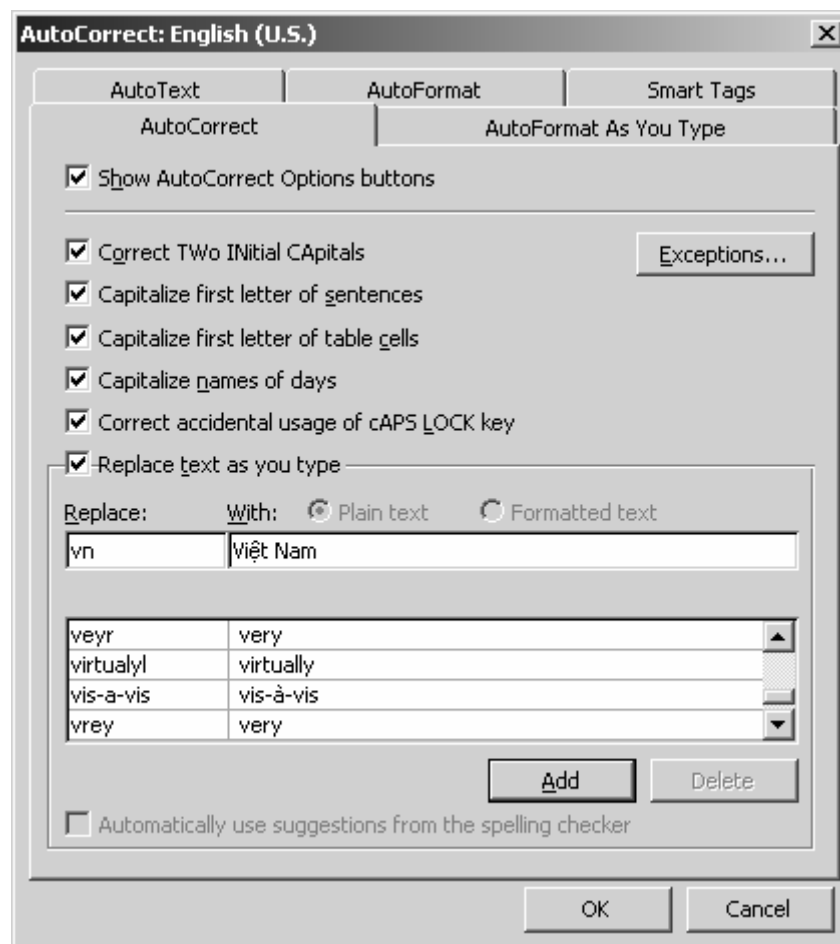
Để thêm một từ viết tắt, thực hiện theo các bước sau

Bước 1: Khởi động tính năng **AutoCorrect** bằng cách mở mục chọn lệnh: **Tool / AutoCorrect ...** Hộp thoại **Autocorrect** xuất hiện.

Ví dụ: Muốn viết tắt từ **Việt Nam** bởi **vn**. Thì gõ **vn** vào hộp thoại **Replace**, trong hộp thoại **With** gõ **Việt Nam**

Bước 2: Gõ cụm từ viết tắt vào mục **Replace**. Gõ cụm từ thay thế vào mục **With**.

Bước 3: Sau đó nhấn nút **Add** để ghi thêm từ này vào danh sách viết tắt của Word.



✎ Xóa đi một từ viết tắt

Thực hiện tương tự các bước như tạo một cụm từ viết tắt. Nhưng chọn cụm từ muốn xóa và nhấn **Delete**

Lưu ý: Khi ta soạn thảo văn bản trên Word thường gặp phải lỗi sau

Gõ chữ **Thi** máy sẽ tự động sửa thành chữ **THI**, hoặc gõ chữ **Thien** máy tự động sửa thành chữ **ThIên**.

Tức là chữ **i** (i thường) tự động chuyển thành chữ **I** (I hoa). Lỗi này do Autocorrect đặt sửa lỗi chính tả **i** thành **I**. Để sửa lỗi này chúng ta chỉ cần xóa đi kiểu viết tắt chữ **i** thành **I** của Autocorrect.

✎ Các tùy chọn sửa lỗi chính tả.

Autocorrect cho phép thiết lập một số tính chất tự động sửa các lỗi chính tả trong quá trình soạn thảo. Để thiết lập các tính chất tự động sửa lỗi chính tả, thực hiện các bước

Khởi động tính năng AutoCorrect

Thiết lập các thông số về sửa lỗi chính tả bằng các mục sau

Correct Two Initial Capitals

Tự động sửa nếu có 2 ký tự hoa đầu dòng sẽ chỉ để lại chữ hoa đầu tiên.

Capitalize first letter of sentences

Tự động sửa ký tự đầu tiên của một câu thành chữ hoa.

Capitalize names of days

Viết hoa ký tự đầu tiên của ngày.

CHƯƠNG 4.

TẠO BẢNG - TABLE

4.1. CẤU TẠO VÀ CÁCH TẠO BẢNG

4.1.1. Cấu tạo của bảng

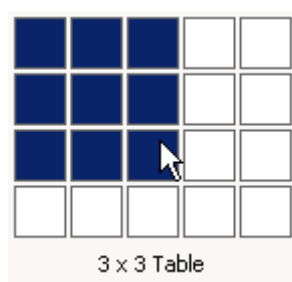
Một bảng được tạo thành từ các dòng (rows) và các cột (columns). Giao của dòng và cột tạo thành một ô (cell). Mỗi ô sẽ độc lập với các mục khác và nó có thể chứa mọi thứ ngoại trừ một bảng khác. Bảng thường được dùng để trình bày dữ liệu theo dòng và cột như danh sách, hay các dạng văn bản có yêu cầu trình bày dạng dòng và cột.

4.1.2. Cách tạo bảng

Cách 1: Dùng nút công cụ

Bước 1: Nhấp nút **Insert Table** , Word sẽ bung xuống các ô lưới.

Bước 2: Rà chuột để chọn số dòng và số cột.



Cách 2: Chọn lệnh **Table\Insert\Table** để mở hộp thoại **Insert Table**.



Hình 3.21 Hộp thoại Insert Table

Bước 1: Xác định số cột và số dòng của bảng trong mục *Number of columns* và *Number of rows*.

Bước 2: Click OK.

4.1.3. Di chuyển con trỏ trong bảng:

- **Tab:** di chuyển đến ô kế tiếp.


- **Shift+Tab**: di chuyển đến ô trước đó.
- **Alt+Home/End**: về ô đầu tiên/cuối cùng của dòng hiện hành.
- **Alt+PgUp/PgDn**: về ô đầu/cuối của cột hiện hành.
- Dùng chuột để click vào ô muốn chuyển đến.

Chú ý: Khi con trỏ đang ở ô cuối cùng của bảng, nếu gõ phím Tab thì Word sẽ tự động thêm một dòng.

4.2. CÁC THAO TÁC CƠ BẢN TRÊN BẢNG


4.2.1. Chỉnh độ rộng ô, cột

4.2.1.1. Thay đổi độ rộng chỉ một ô

Bước 1: Chọn ô muốn thay đổi độ rộng bằng cách trỏ chuột vào biên trái của ô, khi trỏ chuột có dạng , nhấp chuột trái, ô được tô đen.


Bước 2: Trỏ vào lưới dọc của ô (trái hoặc phải), bấm giữ chuột và kéo để co/giãn độ rộng.

4.2.1.2. Thay đổi độ rộng cột

Trỏ chuột vào lưới dọc của cột cần thay đổi độ rộng; khi con trỏ chuyển sang dạng  thì bấm giữ chuột trái rồi kéo di chuyển để thay đổi độ rộng.

4.2.2. Chỉnh độ cao của dòng

Các ô trên một dòng phải có chiều cao như nhau; nghĩa là chỉ có thể chỉnh chiều cao chung cho các ô của dòng chứ không định chiều cao riêng cho từng ô được.

Trỏ chuột vào thước dọc xuyên qua vị trí lưới phía dưới dòng muốn định chiều cao; khi con trỏ có dạng , bấm giữ chuột trái đồng thời kéo chuột lên, xuống để tăng/giảm chiều cao.

4.2.3. Xoá dòng, cột, ô

- Xoá nguyên dòng

Bước 1: Chọn các dòng muốn xoá.

Bước 2: Chọn lệnh **Table\Delete\Rows**.

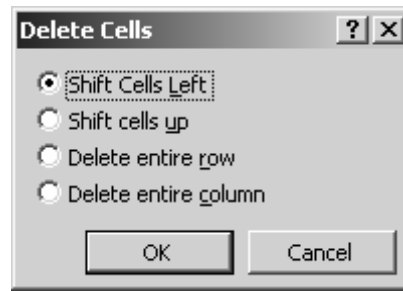
- Xoá nguyên cột

Bước 1: Chọn các cột muốn xoá (bôi đen tất cả các ô trong các cột muốn xoá).

Bước 2: Chọn lệnh **Table\Delete\Columns**.

- Xoá ô, xoá cột, xoá dòng

Bước 1: Chọn các ô, dòng hay cột cần xoá.



Hình 3.22 Hộp thoại Delete Cells

Bước 2: Chọn lệnh **Table\Delete\Cells** để mở hộp thoại **Delete Cells**.

Bước 3: Chọn một trong bốn tùy chọn sau đây:

- **Shift cells left:** xoá các ô đang chọn và dồn các ô ở bên phải về phía trái.
- **Shift cells up:** xoá các ô đang chọn và dồn các ô phía dưới lên trên.
- **Delete entire row:** xoá nguyên dòng và dời các dòng ở phía dưới lên.
- **Delete entire column:** xoá các cột đang chọn và dời các cột bên phải qua.

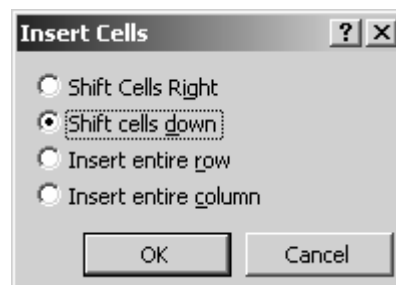
Bước 4: Nhấp **OK**.

4.2.4. Thêm dòng, thêm cột, thêm ô

- Dùng lệnh **Insert Cell:**

Bước 1: Chọn các ô, dòng hay cột cần thêm.

Bước 2: Chọn lệnh **Table\Insert\Cells** để mở hộp thoại **Insert Cells**.



Hình 3.23 Hộp thoại Insert Cells

Bước 3: Chọn một trong bốn tùy chọn sau đây:

- **Shift cells right:** ô mới xen vào đây ô hiện hành sang phải.
- **Shift cells down:** ô mới xen vào đây ô hiện hành xuống dưới.
- **Insert entire row:** xen trọn dòng, các dòng đang chọn sẽ bị đẩy xuống dưới.
- **Insert entire column:** xen trọn cột, các cột đang chọn sẽ bị đẩy sang phải.

Bước 4: Nhấp **OK**.

- Chèn một hoặc nhiều dòng phía trên/dưới dòng đang chọn

Bước 1: Chọn một hoặc nhiều dòng để xác định cho Word biết là muốn chèn vào một hay nhiều dòng.

Bước 2: Chọn lệnh **Table\Insert\Rows Above/Rows Below**.

- Chèn thêm dòng ở cuối bảng

Bước 1: Đặt con trỏ ở ô cuối cùng bên phải của bảng.

Bước 2: Gõ phím **Tab**.

- Chèn thêm cột phía bên trái/bên phải cột đang chọn

Bước 1: Chọn một hoặc nhiều cột để xác định cho Word biết là muốn chèn vào một hay nhiều cột.

Bước 2: Chọn lệnh **Table\Insert\Columns to the Left/Columns to the Right**.

4.2.5. Kẻ khung và tô nền cho bảng

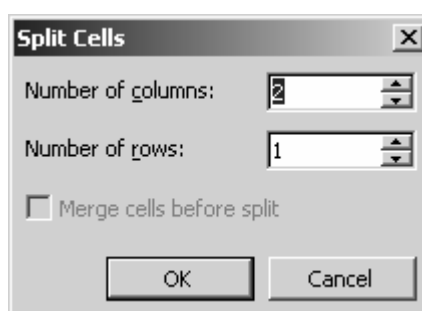
Dùng lệnh **Format\Borders and Shading**.

4.2.6. Tách và ghép các ô

4.2.6.1. Tách một ô thành nhiều ô (Split Cells)

Bước 1: Chọn ô/khối ô muốn tách.

Bước 2: Chọn lệnh **Table\Split Cells** để mở hộp thoại **Split Cells**.



Hình 3.24 Hộp thoại Split Cells

Bước 3: Xác định số dòng và số cột muốn có sau khi tách trong hộp **Number of columns** và **Number of rows**.

Bước 4: Nhấp **OK**.

4.2.6.2. Ghép nhiều ô thành một ô (Merge Cell)

Bước 1: Chọn các ô muốn ghép lại thành một ô.

Bước 2: Chọn lệnh **Table\Merge Cell**.

4.2.7 Tính toán trên bảng

Bạn có thể thực hiện một số phép tính đơn giản trên bảng của Word. Một khía cạnh nào đấy Word cũng có một số khả năng giống như bảng tính Excel tuy nhiên nó chỉ xử lý được các phép tính đơn giản.

- **Tính tổng**

Giả sử ta có một bảng số liệu sau

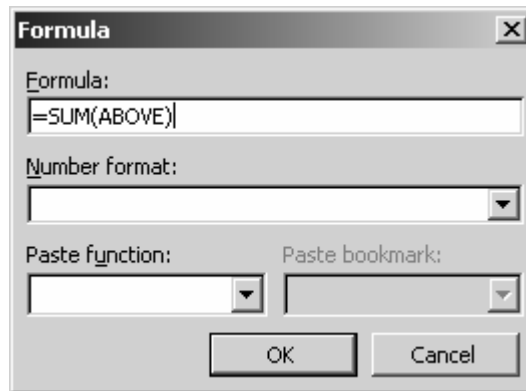
STT	Họ tên	Lương	Phụ cấp	Thực lĩnh
1.	Trác Bội Dao	1800000	150000	?

2.	Nguyễn Thị Hải Yến	2000000	120000	?
3.	Bùi Bích Vân	2500000	100000	?
	Tổng cộng	?	?	?

Tính các giá trị chưa có trong bảng biểu.

Bước 1: Đặt con trỏ lên ô cần chứa giá trị tổng.

Bước 2: Mở mục chọn **Table / Formula** hộp thoại Formula xuất hiện.



Ở mục **Formula** nhập vào công thức cần tính. Có 3 loại công thức tính tổng:

- **SUM(ABOVE)** Tính tổng các dòng từ trên dòng đặt con trỏ. Công thức này được áp dụng cho dòng **Tổng cộng**.
- **SUM(LEFT)** Tính tổng cho các dòng bên trái đặt con trỏ. Công thức này được tính **Cột thực lĩnh = Lương + Phụ cấp**.
- **SUM(RIGHT)** Tính tổng cho các dòng bên phải đặt con trỏ.

Thực hiện xong ta sẽ có kết quả sau.

STT	Họ tên	Lương	Phụ cấp	Thực lĩnh
1.	Trác Bội Dao	1800000	150000	1950000
2.	Nguyễn Thị Hải Yến	2000000	120000	1950000
3.	Bùi Bích Vân	2500000	100000	2600000
	Tổng cộng	6300000	370000	6500000

- **Tính trung bình cộng**

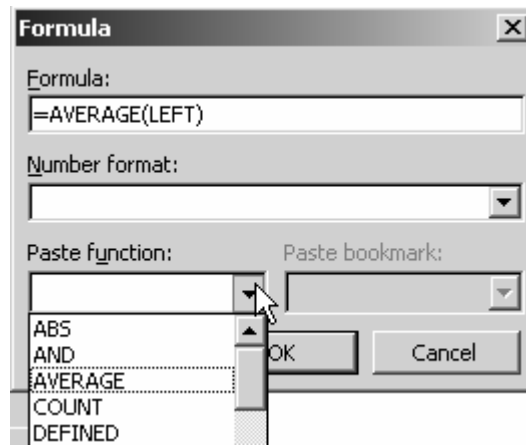
Giả sử ta có bảng số liệu như sau

STT	Họ tên	Điểm học kỳ 1	Điểm học kỳ 2	Điểm cả năm
1.	Trác Bội Dao	7.0	7.5	?
2.	Nguyễn Thị Hải Yến	7.4	7.0	?
3.	Bùi Bích Vân	8.2	8.5	?
	Điểm trung bình cả lớp	?	?	?

Dùng hàm tính trung bình cộng của Word để điền vào các ô còn trống

Bước 1: Đặt con trỏ lên ô cần chứa giá trị tổng.

Bước 2: Mở mục chọn **Table / Formula** hộp thoại **Formula** xuất hiện.



Ở mục **Formula** nhập vào công thức cần tính. Tương tự có 3 loại công thức tính trung bình:

- **AVERAGE(ABOVE)** Tính trung bình các dòng từ trên dòng đặt con trỏ. Công thức này được áp dụng cho dòng *Tính trung bình chung cả lớp*.
- **AVERAGE(LEFT)** Tính trung bình cho các dòng bên trái đặt con trỏ. Công thức này được tính *Điểm cả năm = (Điểm học kỳ 1 + Điểm học 2)/2*
- **AVERAGE(RIGHT)** Tính trung bình cho các dòng bên phải đặt con trỏ.

Thực hiện xong ta sẽ có kết quả sau.

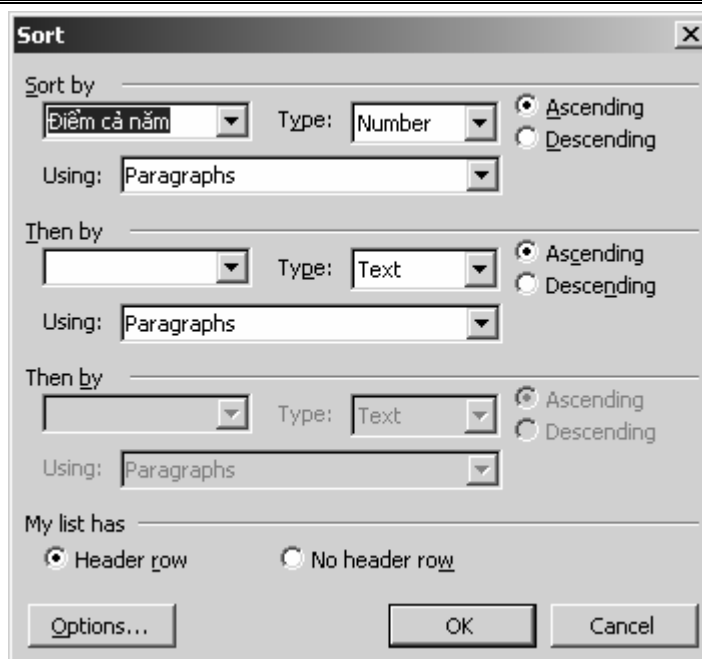
STT	Họ tên	Điểm học kỳ 1	Điểm học kỳ 2	Điểm cả năm
1.	Trác Bội Dao	7.0	7.5	7.25
2.	Nguyễn Thị Hải Yến	7.4	7.0	7.2
3.	Bùi Bích Vân	8.2	8.5	8.35
Điểm trung bình cả lớp		7.53	7.67	7.6

- **Sắp xếp dữ liệu trên bảng**

Sắp xếp dữ liệu theo điểm trung bình năm học

STT	Họ tên	Điểm học kỳ 1	Điểm học kỳ 2	Điểm cả năm
1.	Nguyễn Thị Hải Yến	7.4	7.0	7.2
2.	Trác Bội Dao	7.0	7.5	7.25
3.	Bùi Bích Vân	8.2	8.5	8.35
Điểm trung bình cả lớp		7.53	7.67	7.6

Bước 1: Chọn các trường dữ liệu cần sắp xếp rồi mở mục **Table/ Sort** hộp thoại sau xuất hiện.



Bước 2: Chọn cột cần sắp xếp trong hộp **Sort by**.

Bước 3: Nhấn OK

Kết quả thu được

STT	Họ tên	Điểm học kỳ 1	Điểm học kỳ 2	Điểm cả năm
4.	Nguyễn Thị Hải Yến	7.4	7.0	7.2
5.	Trác Bội Dao	7.0	7.5	7.25
6.	Bùi Bích Vân	8.2	8.5	8.35
Điểm trung bình cả lớp		7.53	7.67	7.6

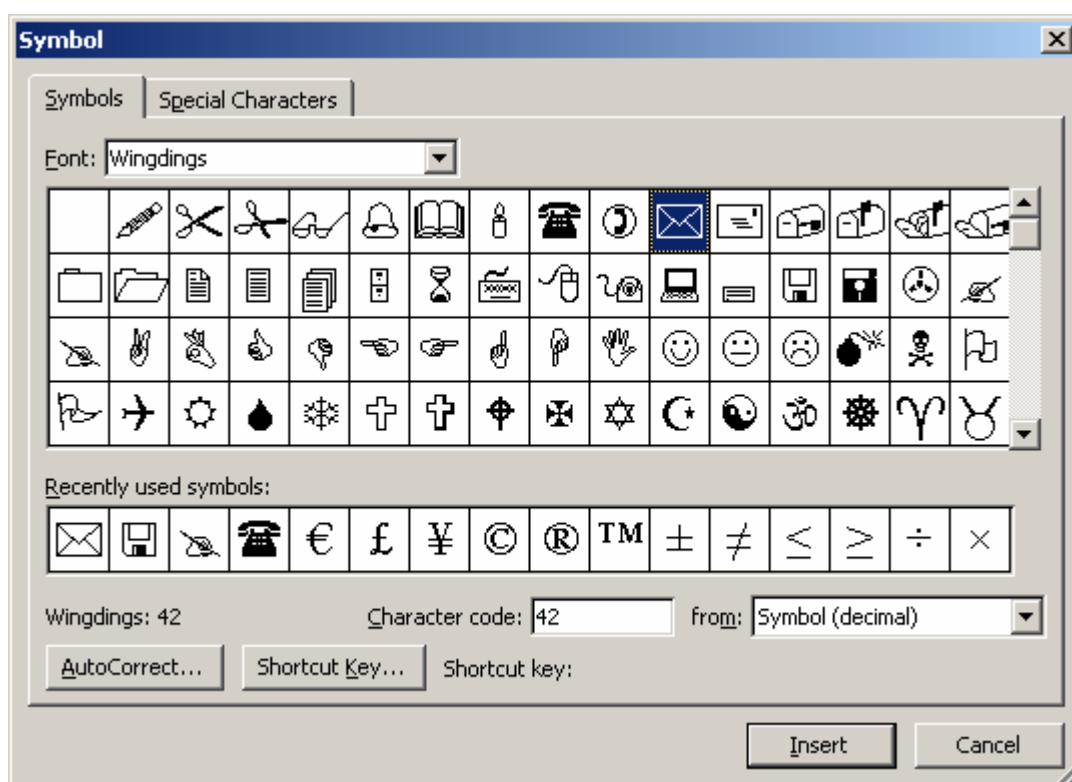
CHƯƠNG 5.

TRANG TRÍ VĂN BẢN

5.1. CHÈN CÁC KÝ HIỆU ĐẶC BIỆT.

Trong quá trình soạn thảo văn bản nhiều lúc chúng ta cần chèn những ký hiệu đặc biệt không có trên bàn phím, do đó không thể nhập trực tiếp từ bàn phím mà phải dùng lệnh **Insert\Symbol** để chèn vào.

Bước 1: Đặt con trỏ tại vị trí muốn chèn. Chọn lệnh **Insert\Symbol**.



Hình 3.25 Hộp thoại Symbol

Bước 2: Chọn lớp **Symbol** (hoặc **Special Characters**).

Bước 3: Chọn font thích hợp. Thường là các font **Symbol**, **Wingdings**, **Webdings**.

Bước 4: Nhấp đúp trên symbol cần chèn vào.

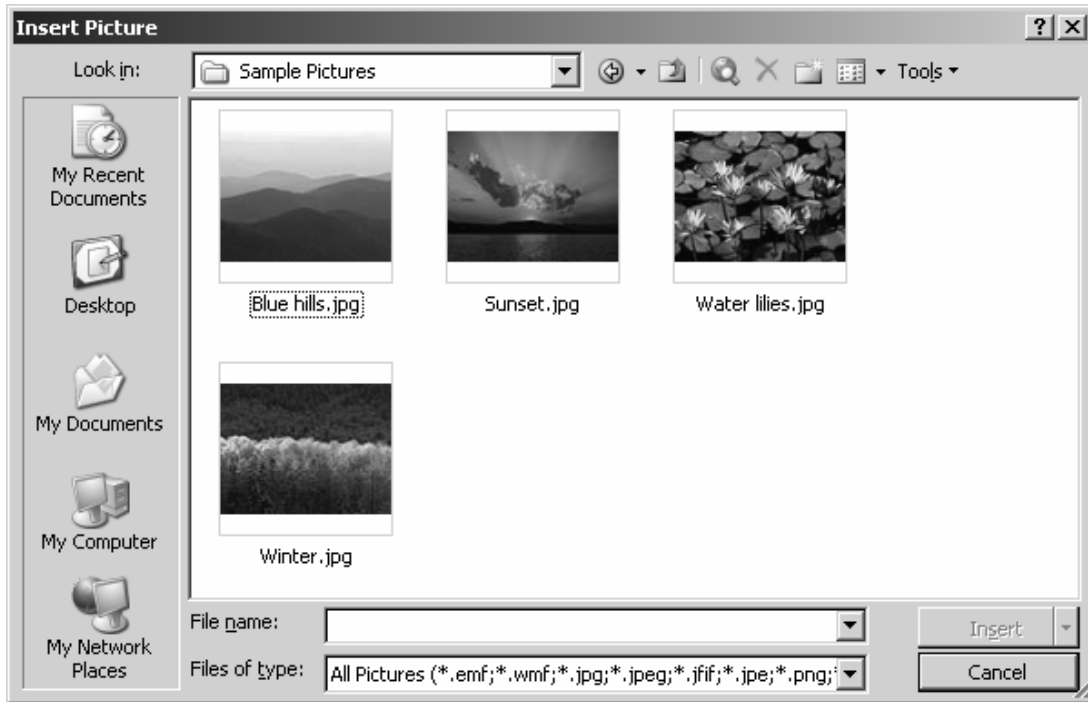
Bước 5: Muốn kết thúc thì nhấp nút Close.

5.2. CHÈN HÌNH ẢNH

5.2.1. Chèn ảnh từ File (Insert\Picture)

Các loại ảnh được tạo từ các phần mềm khác có thể đưa vào trong văn bản.


Bước 1: Dùng lệnh **Insert\Picture\From File** để mở hộp thoại **Insert Picture**.



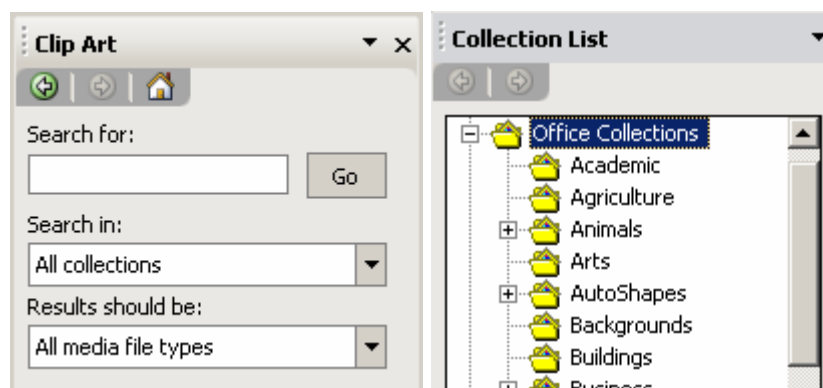
Hình 3.26 Hộp thoại Insert Picture

Bước 2: Xác định đường dẫn đến tập tin ảnh trong hộp thoại **Insert Picture** rồi click **Insert** để chèn ảnh vào văn bản.

5.2.2. Chèn ảnh từ thư viện ảnh (ClipArt)

Để chèn Clip Art vào văn bản ta dùng lệnh dùng lệnh **Insert\Picture\Clip Art** hoặc lệnh **Insert Clip Art**  trên thanh công cụ **Drawing**

Clip Art Task Pane:



Hình 3.27 Chèn Clip Art vào văn bản

Các bước thực hiện:

Bước 1: Đặt con nháy tại vị trí muốn chèn Clip Art.

Bước 2: Chọn lệnh **Insert > Picture > Clip Art** hoặc lệnh **Insert Clip Art**  trên thanh công cụ **Drawing**.

Bước 3: **Clip Art Task Pane** xuất hiện, click lệnh **Go** để tìm tất cả các Clip Art của Word. Nếu muốn tìm theo chủ đề thì chọn chủ đề trong mục **Search In** trước khi Click nút **Go**

Bước 4: Di chuyển thanh cuộn để chọn Clip Art mong muốn, click vào Clip Art để chèn nó vào văn bản tại vị trí con nháy.

5.2.3. Định dạng ảnh.

- Thay đổi kích thước của ảnh

Bước 1: Nhấp trên ảnh, khi đó trên 4 góc và 4 cạnh của ảnh xuất hiện nốt vuông màu trắng.

Bước 2: Trỏ chuột vào các nốt này, khi chuột có dạng mũi tên hai đầu thì kéo để thay đổi kích thước.

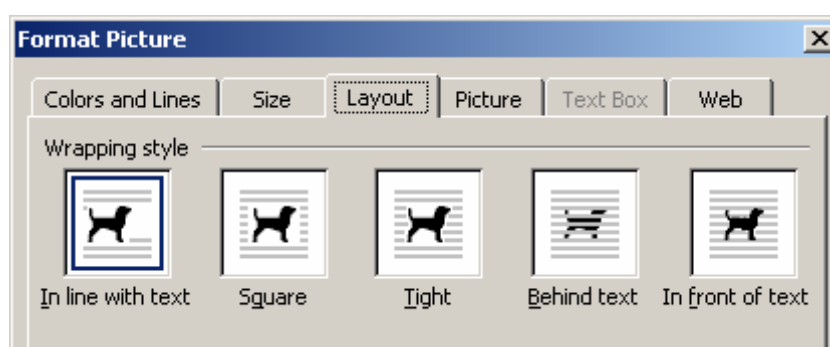
- Chọn kiểu bố trí văn bản bao quanh ảnh

Bước 1: Click phải vào ảnh và chọn **Format Picture** để mở hộp thoại **Format Picture**.

Bước 2: Chọn tab **Layout**.

Bước 3: Chọn một trong các dạng bố trí ảnh sau:

- + **In line with text:** Ảnh được xem như một ký tự đặc biệt trong văn bản.
- + **Square:** Văn bản bao quanh khối vuông của ảnh.
- + **Tight:** Tương tự như Square nhưng văn bản sẽ được đặt trên các phần biên trống của ảnh.
- + **Behind text:** Ảnh đặt sau văn bản.
- + **In front of text:** Ảnh đặt trước văn bản.



Hình 3.28 Hộp thoại **Format Picture** – Thẻ **Layout**

- Di chuyển ảnh

Trỏ chuột vào ảnh và bấm giữ nút chuột trái và kéo ảnh đến vị trí mới.

- **Xóa ảnh:** Click chọn ảnh rồi nhấn phím **Delete**.

5.3. TẠO CHỮ NGHỆ THUẬT (WORDART)

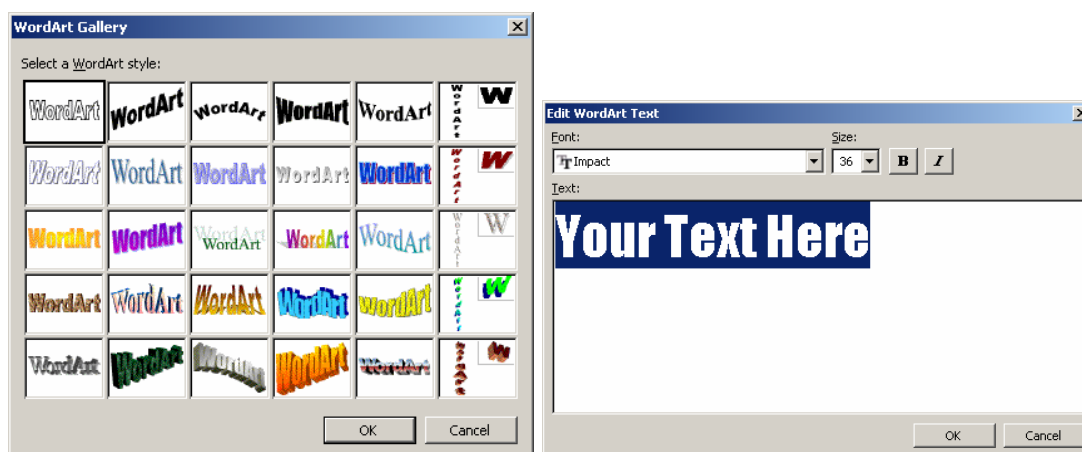
WordArt là dạng chữ dùng để trang trí tiêu đề của văn bản, bìa sách, trang trí thiệp mời, quảng cáo...

Cách tạo:

Bước 1: Đặt con trỏ tại vị trí muốn chèn WordArt.

Bước 2: Chọn lệnh **Insert\Picture\WordArt** hoặc nhấp nút lệnh  trên thanh công cụ Drawing.

Bước 3: Chọn kiểu trình bày trong **WordArt Gallery**.





Hình 3.29 . Chèn Word Art vào văn bản

Bước 4: Nhập nội dung văn bản trang trí vào hộp **Edit WordArt Text**. Có thể chọn Font, cỡ chữ, kiểu chữ cho văn bản.

Bước 5: Nhấp **OK**.


Các thao tác định dạng WordArt cũng thực hiện tương tự như định dạng ảnh. Ngoài ra, bạn có thể dùng các nút lệnh sau trên thanh công cụ WordArt (xuất hiện khi bạn click chọn mẫu Wordart).

-  *WordArt Gallery*: chọn lại kiểu WordArt.
-  *WordArt Shape*: chọn dạng bố trí chữ.

5.4. CHÈN TEXTBOX

Text box là một hộp hình chữ nhật mà trong đó có thể chứa các đối tượng khác như văn bản, ảnh, biểu bảng,... ngoại trừ một Text box khác. Có thể di chuyển và đặt Text box ở vị trí mong muốn.

Cách vẽ Text box:

Bước 1: Nhấp chuột trên nút công cụ Text Box  trên thanh công cụ vẽ (Drawing) ở cuối màn hình, khi đó con trỏ có dạng dấu cộng.

Bước 2: Đặt con trỏ tại vị trí muốn vẽ, bấm giữ chuột trái và kéo để vẽ.

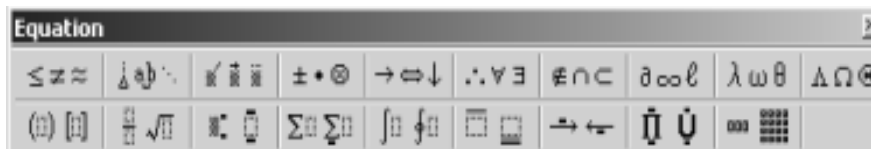
Bước 3: Nhập và định dạng văn bản trong textbox.

Định dạng Text Box: Thực hiện tương tự như định dạng ảnh.

5.5. CHÈN KÝ HIỆU VÀ CÔNG THỨC TOÁN HỌC.

Bước 1: Đặt con trỏ tại vị trí muốn nhập công thức.

Bước 2: Chọn lệnh **Insert\Object\Microsoft Equation 3.0**, thanh công cụ **Equation** xuất hiện.



Hình 3.30. Hộp thoại Equation

Bước 3: Nhấp chọn các ký hiệu không có trên bàn phím như dấu căn, lũy thừa, ma trận,... sau đó nhập các ký tự có trên bàn phím vào các textbox trống trong công thức.

Bước 4: Nhập xong nhấp tại vị trí bất kỳ trong văn bản; thanh công cụ Equation tự động biến mất và công thức được đưa vào trong văn bản.

Ví dụ: 1) $x_1 = x_2 = \sqrt{\frac{-b + \sqrt{b^2 - 4ac}}{2a}}$ 2) $x_3 = x_4 = \sqrt{\frac{-b - \sqrt{b^2 - 4ac}}{2a}}$

3) $\sin \alpha = \pm \sqrt{1 - \cos^2 \alpha} = \frac{\pm \operatorname{tg} \alpha}{\sqrt{1 + \operatorname{tg}^2 \alpha}} = \frac{\pm 1}{\sqrt{1 + \cot^2 \alpha}}$

Cách điều chỉnh công thức toán học:

Bước 1: Nhấp đúp trên công thức, công thức được kích hoạt và thanh công cụ Equation xuất hiện.

Bước 2: Tiến hành điều chỉnh công thức giống như lúc nhập.

Bước 3: Điều chỉnh xong nhấp ngoài văn bản.

5.6. VẼ HÌNH.









Word sử dụng thanh công cụ Drawing phục vụ cho việc vẽ hình. Với các công cụ trên thanh công cụ Drawing, Word cho phép vẽ các đối tượng như: đường thẳng (line), hình chữ nhật (rectangle), oval, textbox,... và định dạng các đối tượng này.



Hình 3.31. Thanh công cụ Drawing

5.6.1. Các nút công cụ thường dùng trên thanh Drawing

AutoShapes ▾	AutoShape: chọn mẫu vẽ		Fill Color: định màu tô đặc (lấp đầy)
--------------	------------------------	---	---------------------------------------

	<i>Line</i> : vẽ đường thẳng		<i>Line Color</i> : định màu đường viền
	<i>Arrow</i> : vẽ đường thẳng có mũi tên		<i>Arrow Style</i> : các kiểu đầu mũi tên
	<i>Rectangle</i> : vẽ hình chữ nhật		<i>Line Style</i> : kiểu đường liền nét
	<i>Oval</i> : vẽ hình tròn, elip		<i>Dash Style</i> : kiểu đường đứt nét

5.6.2. Các thao tác cơ bản trên hình ảnh



- Cách vẽ đối tượng

Bước 1: Nhấp vào **AutoShapes** hoặc các nút vẽ đường thẳng, mũi tên, hình chữ nhật, oval,... để chọn loại hình vẽ.

Bước 2: Kéo chuột để vẽ hình tại vị trí mong muốn.


- Thay đổi nét vẽ cho đối tượng hình

Bước 1: Chọn hình vẽ muốn thay đổi nét vẽ.

Bước 2: Chọn nút lệnh **Line style** , hoặc **Dash style**  để chọn kiểu nét vẽ.


- Thay đổi màu viền (line color) cho đối tượng hình

Bước 1: Chọn hình vẽ muốn thay đổi màu viền.

Bước 2: Chọn mũi tên bên phải nút lệnh **Line color**  để chọn màu cho đường viền của hình vẽ.

- Thay đổi màu tô (fill color) cho hình vẽ

Bước 1: Chọn hình vẽ muốn thay đổi màu tô.

Bước 2: Chọn mũi tên bên phải nút lệnh **Fill color**  để chọn màu tô đặc cho hình vẽ.

- **Nhập văn bản vào hình vẽ:** Click phải vào hình vẽ và chọn Add Text từ menu tắt, sau đó nhập và định dạng giống như văn bản thông thường.

- **Di chuyển, thay đổi kích thước và chọn dạng bố trí văn bản bao quanh hình vẽ:** Thực hiện tương tự như đối với ảnh.

BÀI TẬP

Bài 1: Sử dụng Font chữ (Format Font).

Yêu cầu:

- Mở file mới và trình bày đoạn văn bản bên dưới. Lưu ý không đánh số dòng kèm theo bài làm của các bạn.
- Dòng 1: dùng Font **VNI-Present**, cỡ chữ 28.
- Dòng 2: chữ “Tác giả” dùng Font **VNI-Duff**, cỡ chữ 12. Chữ “Thanh Tịnh” dùng Font **VNI-Commerce**, cỡ chữ 18.
- Đoạn từ dòng 3 đến dòng 7 dùng Font Unicode **Arial**, cỡ chữ 12.
- Các đoạn từ dòng 8 đến dòng 14 dùng font Unicode **Times New Roman**, cỡ chữ 13.
- Xong lưu lại (Save) với tên **Bai tap 1.doc**.

1

Tôi Đi Học

2

Tác giả: *Thanh Tịnh*

3

Hằng năm, cứ vào cuối thu, lá ngoài đường rụng nhiều, và trên không có những đám mây bàng bạc, lòng tôi lại nao nức những kỷ niệm mơn man của buổi tựu trường.

5

Tôi quên thế nào được những cảm giác trong sáng ấy nảy nở trong lòng tôi như mấy cành hoa tươi mỉm cười giữa bầu trời quang đãng.

7




Những ý tưởng ấy, tôi chưa lần nào ghi lên giấy, vì hồi ấy tôi không biết ghi và ngày nay tôi không nhớ hết. Nhưng mỗi lần thấy mấy em nhỏ rụt rè núp dưới nón mẹ lần đầu tiên đi đến trường, lòng tôi lại tưng bừng rộn rã.

10

Buổi mai hôm ấy, một buổi mai đầy sương thu và đầy gió lạnh. Mẹ tôi âu yếm nắm tay tôi dẫn đi trên con đường làng dài và hẹp. Con đường này, tôi đã quen đi lại lắm lần, nhưng lần này tôi tự nhiên thấy lạ. Cảnh vật chung quanh tôi đều thay đổi, vì chính lòng tôi đang có sự thay đổi lớn: hôm nay tôi đi học

14

Bài 2: Định dạng Font chữ, định dạng đoạn (paragraph) cơ bản.**Yêu cầu:**

- Mở file Bai tap 1.doc và lưu lại (Save As...) với tên mới là Bai tap 2.doc.
- Nhập thêm các đoạn từ dòng 15 → dòng 22 bằng Font unicode Times New Roman, cỡ chữ 13.
- Dòng 1: chọn in đậm (Bold **B**) và canh giữa (Center .
- Dòng 2: chọn canh phải (Right .
- Các đoạn từ dòng 3 → dòng 22 chọn canh đều (Justify .
- Đoạn từ dòng 3 → dòng 4: in nghiêng (*I*).
- Dòng 13: đậm, nghiêng, gạch dưới (***U***).
- Chọn màu sắc cho chữ (tùy ý).

Tôi Đi Học

Tác giả: *Thanh Tịnh*

Hàng năm, cứ vào cuối thu, lá ngoài đường rụng nhiều, và trên không có những đám mây bàng bạc, lòng tôi lại nao nức những kỷ niệm mơn man của buổi tựu trường.

Tôi quên thế nào được những cảm giác trong sáng ấy nảy nở trong lòng tôi như mấy cành hoa tươi mỉm cười giữa bầu trời quang đãng.

Những ý tưởng ấy, tôi chưa lần nào ghi lên giấy, vì hồi ấy tôi không biết ghi và ngày nay tôi không nhớ hết. Nhưng mỗi lần thấy mấy em nhỏ rụt rè núp dưới nón mẹ lần đầu tiên đi đến trường, lòng tôi lại tưng bừng rộn rã.

Buổi mai hôm ấy, một buổi mai đầy sương thu và đầy gió lạnh. Mẹ tôi âu yếm nắm tay tôi dẫn đi trên con đường làng dài và hẹp. Con đường này, tôi đã quen đi lại lắm lần, nhưng lần này tôi tự nhiên thấy lạ. Cảnh vật chung quanh tôi đều thay đổi, vì chính lòng tôi đang có sự thay đổi lớn: **hôm nay tôi đi học**.



Tôi không lội qua sông thả diều như thằng Quý và không đi ra đồng nô đùa như thằng Sơn nữa.

Trong chiếc áo vải dù đen dài, tôi cảm thấy mình trang trọng và đứng đắn.

Đọc đường thấy mấy cậu nhỏ trạc bằng tôi, áo quần tươm tất, nhí nhảnh gọi tên nhau hay trao sách vở cho nhau xem mà tôi thèm. Hai quyển vở mới đang ở trên tay tôi đã bắt đầu thấy nặng. Tôi bặm tay ghi thật chặt, nhưng một quyển vở cũng xệch ra và chênh đầu chúi xuống đất. Tôi xóc lên và nắm lại cẩn thận. Mấy cậu đi trước ôm sách vở nhiều, lại kèm cả bút thước nữa. Nhưng mấy cậu không để lộ vẻ khó khăn gì hết

Bài 3: Định dạng đoạn (Format paragraph): biên trái (left), biên phải (right). Khoảng cách giữa các đoạn (before, after).

Yêu cầu:

- Tạo file mới và lưu với tên là Bai tap 3.doc.
- Tựa đề bài thơ dòng 1 (chữ Quê Hương) dùng Font VNI-Ariston, cỡ chữ 26.
- Dòng 2 dùng Font VNI-Commerce, cỡ chữ 18.
- Dòng 3 → dòng 13 dùng Font VNI-Zap, cỡ chữ 14.
- Dòng 1 canh giữa (Center .
- Dòng 2 canh phải (Right , biên phải 3cm.
- Đoạn từ dòng 3 → dòng 6 canh trái, biên trái (left) 3.5cm.
- Đoạn từ dòng 7 → dòng 12 canh trái, biên trái (left) 5cm.
- Các đoạn (paragraph) cách nhau 9pt (giữa dòng 2 và 3; giữa dòng 6 và 7).

1 QUÊ HƯƠNG

2 Thơ: *Giang Nam*

3 *Thuở còn thơ ngày hai buổi đến trường*

4 *Yêu quê hương qua từng trang sách nhỏ*

5 *Ai bảo chăn trâu là khổ?*

6 *Tôi mơ màng nghe chim hót trên cao.*

7 *Những ngày trốn học*

8 *Đuổi bướm cạnh cầu ao*

9 *Mẹ bắt được*

10 *Chưa đánh roi nào đã khóc*



11 *Cô bé nhà bên nhìn tôi cười khúc khích*


12 *Mắt đen tròn thương thương quá đi thôi.*


Bài 4: Định dạng đoạn (Format paragraph), 3 kiểu trình bày đoạn văn bản: bình thường (normal), thụt vô đầu dòng (first line) và thụt ra đầu dòng (hanging).

Yêu cầu:

- Tạo file mới và lưu với tên là Bai tap 4.doc.
- Đoạn từ dòng 1 → dòng 6 trình bày kiểu bình thường.
- Đoạn từ dòng 7 → dòng 11 trình bày kiểu First line, thụt đầu dòng 1cm.
- Đoạn từ dòng 12 → dòng 16 trình bày kiểu Hanging, các dòng sau thụt vào 1.5cm.
- Các đoạn cách nhau 9pt.

1 Đây là đoạn văn bản trình bày theo kiểu bình thường. Dòng đầu và các dòng sau cách
2 biên trái một khoảng bằng nhau. Chúng ta có thể dùng hộp thoại **Format Paragraph**
3 (**Format → Paragraph**), chỉnh giá trị **Left** trong phần **Indentation** để tăng, giảm biên trái
4 của đoạn. Tương tự đối với biên phải, chỉnh giá trị **Right**. Còn muốn làm nhanh, bạn có
5 thể dùng chuột drag **nút vuông** trên thước kẻ () bên trái để tăng, giảm biên trái; hoặc
6 **drag nút tam giác** bên phải () để tăng giảm biên phải.

7 Đây là đoạn văn bản trình bày theo kiểu thụt đầu dòng (**First line**). Dòng đầu thụt
8 vào 1 khoảng 1cm, các dòng sau bình thường. Chúng ta có thể dùng hộp thoại **Format**
9 **Paragraph (Format → Paragraph)** chọn giá trị trong ô **Special** là **First line**, rồi chỉnh
10 khoảng cách thụt trong ô **By**. Còn muốn làm nhanh, bạn có thể dùng chuột **drag nút**
11 **tam giác trên** bên trái () để tăng, giảm **First line**.

12 Đây là đoạn văn bản trình bày theo kiểu **hanging**. Dòng đầu bình thường còn các dòng
13 sau thụt vào 1 khoảng 1.5cm. Chúng ta có thể dùng hộp thoại **Format**
14 **Paragraph (Format → Paragraph)** chọn giá trị trong ô **Special** là **Hanging**, rồi
15 chỉnh khoảng cách thụt trong ô **By**. Còn muốn làm nhanh, bạn có thể dùng
16 chuột **drag nút tam giác dưới** bên trái () để tăng, giảm **Hanging**.

Bài 5: Trình bày Font chữ (Format Font), chèn các ký hiệu đặc biệt (Insert symbol).

Yêu cầu:

- Mở file mới và lưu vào Z:\ với tên là Bai tap 5.doc.
- Gõ và trình bày đoạn văn bản bên dưới. Dòng 1 có thể dùng Font VNI-Top, size=16.
- Các dòng còn lại dùng Font unicode Times New Roman, size=13.
- Dùng lệnh Insert → Symbol để chèn các ký hiệu đặc biệt. Các font có chứa các ký hiệu đặc biệt như: Symbol, Webdings, Wingdings, Wingdings 2,3...

TRÌNH BÀY FONT CHỮ

Winword là một phần mềm xử lí văn bản mạnh, ngoài các chức năng giúp người sử dụng gõ nhanh và đúng, nó còn cho phép chúng ta dễ dàng chèn các kí tự đặc biệt vào văn bản như ©, Ω, ≠, α, β, ♣, ♦, ♥, ♠, 📖, ☎, 📧, ✂ ... làm cho hình thức văn bản phong phú hơn.

Văn bản có thể được trình bày nhờ các biểu tượng lệnh như **Bold** để tạo chữ **in đậm**. *Italic* để tạo chữ *in nghiêng*. Underline để tạo chữ gạch dưới, hoặc **kết hợp cả ba**.

Nếu cần kì hơn thì phải dùng lệnh trong menu. Để tạo các kiểu trình bày đặc biệt hơn cho đoạn văn bản đã chọn bằng cách dùng các lệnh **Effects**: Strikethrough tạo ra chữ ~~gạch giữa~~, các lệnh Superscript và Subscript giúp ta tạo được một biểu thức đơn giản có dạng như $a_1x^2 + b_1y^2 = 0$. Các lệnh trong lớp **Character Spacing** có tác dụng điều chỉnh chi tiết cho đoạn văn bản về mặt khoảng cách và vị trí: bằng cách thay đổi giá trị trong mục **Spacing** ta có thể điều chỉnh được **khoảng cách giữa các kí tự**, và khi muốn làm cho một đoạn văn bản được *đưa lên cao* hoặc *chuyển xuống thấp* ta thay đổi giá trị trong mục **Position**.

Bài 6: Kẻ khung, tô nền cho đoạn (paragraph) và cho text. Sử dụng hộp thoại *Border and Shading (Format → Border and Shading)*

Yêu cầu:

- Mở file mới và lưu với tên là Bai tap 6.doc.
- Dòng 1 định dạng paragraph: Left = 3cm, Right = 3cm. Kẻ khung nét đôi, tô nền xám 30%.
- Các chữ đầu ở các dòng 3, 6, 8, 11: tô nền đen.
- Khoảng cách giữa các đoạn là 6pt.

BỐN KIỂU THỂ HIỆN CỦA VĂN BẢN

1 **Một văn bản trong Word có thể được thể hiện bằng một trong bốn kiểu sau**
2 **đây:**





3 **Kiểu Normal:** là kiểu thể hiện không phân trang, văn bản trải dài từ trên xuống
4 dưới. Một số đối tượng như tiêu đề - hạ mục (headers and footers), các hình ảnh không
5 có thuộc tính **In line with text...** sẽ không được hiển thị.

6 **Kiểu Web:** văn bản sẽ thể hiện như một trang web, màu nền hoặc ảnh nền (nếu
7 có) sẽ được thể hiện.

8 **Kiểu Print Layout:** trình bày văn bản theo kiểu phân trang, mọi đối tượng có thể in
9 ra giấy được đều được thể hiện đầy đủ. Đây là kiểu hiển thị thường dùng nhất. Những gì
10 bạn thấy là những gì bạn nhận được (**WYSIWYG**).

11 **Kiểu Outline:** trình bày văn bản theo đề mục (heading 1,2,3...). Bạn có thể che
12 các phần văn bản bên dưới các đề mục, có thể di chuyển cả đoạn 1 cách dễ dàng.

13 Nào, bây giờ hãy thử chuyển đổi qua lại giữa các kiểu view xem sao: vào menu
14 **View** chọn lần lượt **Normal**, **Web Layout**, **Print Layout** và **Outline**.

Bài 8: Sử dụng các công cụ soạn thảo: copy , cut , paste  và format painter .

Yêu cầu:

- Tạo file mới và lưu với tên là **Bai tap 8.doc**. Sau đó dùng các công cụ copy, cut, paste, format painter định dạng như mẫu bên dưới và lưu vào **Z:** với tên **Bai tap 8.doc**. Lưu ý không nhập liệu thêm mà chỉ sử dụng dữ liệu cho sẵn.

THỰC TẬP VỀ COPY, CUT, PASTE

1
2 *Quê hương là chùm khế ngọt*

3 *Cho con trèo hái mỗi ngày*

4 *Quê hương là đường đi học*

5 *Con về rợp bướm vàng bay*

6 *Quê hương là con diều biếc*

7 *Tuổi thơ con thả trên đồng*

8 *Quê hương là con đò nhỏ*

9 *Em đêm khua nước ven sông*

10 *Quê hương là cầu tre nhỏ*

11 *Mẹ về nón lá nghiêng che*

12 *Quê hương là đêm trăng tỏ*

13 *Hoa cau rụng trắng ngoài thềm*

14 *Quê hương mỗi người chỉ một*

15 *Như là chỉ một mẹ thôi*

16 *Quê hương nếu ai không nhớ*

17 *Để không lớn nổi thành người*

THỰC TẬP VỀ FORMAT PAINTER

Trước khi xây dựng cấu trúc định hướng, ta tìm hiểu qua các khái niệm về thứ bậc của **FrontPage**:

- Trang chủ (home page): mỗi web site có một trang đặc biệt gọi là **trang chủ**. **Trang chủ** thường chứa thông tin tổng quát về web site và có các link chỉ tới các trang khác trong site. **Trang chủ** thường có các tên như: [default.htm](#), [index.htm](#), [home.htm](#)..., trong **FrontPage**, **trang chủ** có ký hiệu ngôi nhà kèm theo. Như ví dụ trên, trang [Welcome](#) là **trang chủ**.

- Top-level pages (các trang ở mức cao nhất): là các trang cùng mức với **trang chủ** nhưng không liên kết với **trang chủ**. Trong ví dụ trên là các trang [Description](#) và [Contact Us](#).

- Parent-level Pages (các trang cha): gồm các trang cha trực tiếp và các trang cùng mức với trang cha. Trong ví dụ trên: các trang cha của [What's new](#) là [Welcome](#), [Description](#) và [Contact Us](#). Tương tự, các trang cha của [Downloads](#) là [Products](#), [What's New](#) và [Services](#).

- Child-level Pages (các trang con): là các con trực tiếp của một trang cha. Như ví dụ trên, [Welcome](#) có 3 trang con, [Contact Us](#) có 1 trang con và [Products](#) có 2 trang con.

- Same-level Pages (các trang cùng mức): là các trang cùng mức trong cấu trúc và có cùng cha. Như ví dụ trên: [What's New](#), [Products](#) và [Services](#) là các trang cùng mức, [Downloads](#) và [Ordering](#) là các trang cùng mức.

Bài 9: Chia cột văn bản (Format → Column). Chèn hình ảnh vào văn bản (Insert → Picture).

Yêu cầu:

- Tạo file mới và lưu với tên là Bai tap 9.doc.
- Hình lấy trong bộ Clip Art (Insert → Picture → Clip Art), hoặc lấy từ dữ liệu hình trên máy tính (dùng Insert → Picture → From File).



VUI CƯỜI



Con trai yêu quý,

Viết mấy chữ để con biết rằng mẹ vẫn sống. Mẹ viết rất chậm vì mẹ biết con đọc không được nhanh. Con sẽ không nhận ra nhà của chúng ta khi quay về - vì đã chuyển sang nhà mới. Về bố, bố đã có công việc mới rất khá. Dưới bố có 500 người - ông là thợ sửa mái ngói nhà mồ. Nhà mình mới mua một chiếc máy giặt, xem chừng không được tốt lắm. Tuần trước mẹ cho vào đó 14 chiếc sơ mi, sau khi ấn nút chúng cũng tiêu luôn. Sáng nay chị Mary đã sinh cháu. Mẹ không biết là trai hay là gái vì vậy không hiểu con bây giờ được thăng chức là chú hay là cô. Lễ giáng sinh vừa qua bố không có dịp để uống rượu như mọi năm. Mẹ đã cho một



ít dầu BP vào cốc bia vì thế bố cứ đi đi lại lại mãi suốt một tuần. Hôm rồi mẹ bị cúm và bố đưa mẹ đến viện. Ông bác sỹ cho cặp nhiệt độ vào miệng và yêu cầu mẹ không được há ra trong mười phút. Bố đã kỳ kèo và mua lại chiếc ống đó của ông bác sỹ. Tuần trước chỉ mưa hai lần, một lần ba ngày và một lần bốn ngày liền. Hôm thứ Hai trời gió mạnh quá nên con gà nhà mình phải đẻ bốn lần mới được một quả trứng.

Mẹ của con, Ethel

Tái bút: Mẹ định gửi kèm cho con 10\$ nhưng lúc nhớ ra thì đã trót dán phong bì rồi.

Bài 10: Chia cột văn bản (Format → Column), sử dụng dấu ngắt cột (Insert → Break → Column break). Chèn WordArt vào văn bản.

Yêu cầu:

- Mở file mới, nhập liệu và định dạng như mẫu bên dưới và lưu vào Z:\ với tên Bai tap 10.doc
- WordArt sử dụng font VNI-Univer.
- Định khoảng cách giữa các cột là 0.8cm. Giữa các cột có đường kẻ (line between).
- Các chữ tựa đề dùng font Times New Roman, định dạng paragraph cách đoạn sau (after) là 12pt.
- Phần nội dung dùng font Arial, size=12, định dạng paragraph giữa các đoạn cách nhau 6pt.

1 nụ cười = 10 thang thuốc bổ?

Đãng trí

Bác tôi rất đãng trí.

Ngày nọ ông về nhà sớm và biết có điều gì đấy ông định làm, nhưng ông lại không tài nào nhớ ra được. Ông ngồi đến 12 giờ cố nghĩ cho ra. Cuối cùng, hoá ra, ông muốn đi ngủ sớm!

Kẻ tám lạng

người nửa cân

Hai vợ chồng giận nhau.

Anh chồng đi ngủ trước. Chị vợ đi ngủ sau thấy có mảnh giấy trên bàn ghi:

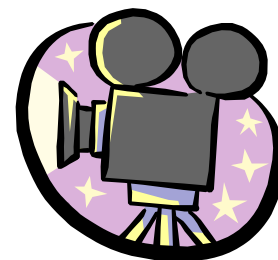
- Đề nghị cô gọi tôi dậy lúc 7 giờ!

Sáng hôm sau, mãi 8 giờ anh mới tỉnh giấc. Lại thấy trên bàn có mảnh giấy ghi:

- Dậy đi thôi, đã 7 giờ rồi!

May quá...

Một anh đến nhà người yêu chơi, khi về bỏ quên bật lửa. Về đến nhà nhớ ra vội vàng quay lại. Bố người yêu đưa chiếc bật lửa cho, anh ta cảm lầy nói: "Dạ, cháu xin. May quá, tí mất!".



Bài 11: Chia cột văn bản. Tạo Drop cap (Format → Drop Cap). Sử dụng Word Art (Insert → Picture → Word Art).

Yêu cầu:

- Tạo file mới và lưu với tên là Bai tap 11.doc.
- Chữ DropCap thứ 1 (chữ H) chọn font VNI-Slogan, drop 3 dòng.
- Chữ DropCap thứ 2 (chữ B) drop 2 dòng.
- Khoảng cách giữa các đoạn là 9pt (point). Căn chỉnh tùy ý sao cho đẹp mắt.

TÔI ĐI HỌC



Tác giả: *Thanh Tịnh*

Hàng năm, cứ vào cuối thu, lá ngoài đường rụng nhiều, và trên không có những đám mây bàng bạc, lòng tôi lại nao nức những kỷ niệm mơn man của buổi tựu trường.

Tôi quên thế nào được những cảm giác trong sáng ấy nảy nở trong lòng tôi như mấy cành hoa tươi mỉm cười giữa bầu trời quang đãng.

Những ý tưởng ấy, tôi chưa lần nào ghi lên giấy, vì hồi ấy tôi không biết ghi và ngày nay tôi không nhớ hết. Nhưng mỗi lần thấy mấy em nhỏ rụt rè núp dưới nón mẹ lần đầu tiên đi đến trường, lòng tôi lại tưng bừng rộn rã.

Buổi mai hôm ấy, một buổi mai đầy sương thu và đầy gió lạnh. Mẹ tôi âu yếm nắm tay tôi dẫn đi trên con đường làng dài và hẹp. Con đường này, tôi đã quen đi lại lắm lần, nhưng lần này tôi tự nhiên thấy lạ. Cảnh vật chung quanh tôi đều thay đổi, vì chính lòng tôi đang có sự thay đổi lớn: ***hôm nay tôi đi học***.

Tôi không lội qua sông thả diều như thằng Quý và không đi ra đồng nô đùa như thằng Sơn nữa.

Trong chiếc áo vải dù đen dài, tôi cảm thấy mình trang trọng và đứng đắn.

Đọc đường thấy mấy cậu nhỏ trạc bằng tôi, áo quần tươm tất, nhí nhảnh gọi tên nhau hay trao sách vở cho nhau xem mà tôi thèm. Hai quyển vở mới đang ở trên tay tôi đã bắt đầu thấy nặng. Tôi bặm tay ghi thật chặt, nhưng một quyển vở cũng xệch ra và chênh đầu chúi xuống đất. Tôi xóc lên và nắm lại cẩn thận. Mấy cậu đi trước ôm sách vở nhiều, lại kèm cả bút thước nữa. Nhưng mấy cậu không để lộ vẻ khó khăn gì hết.

...



Bài 12: Tạo hoa thị và đánh số thứ tự cho các đoạn văn bản (Format → Bullets and Numbering).

Yêu cầu:

- Mở file mới, nhập liệu, định dạng như mẫu bên dưới và lưu vào Z:\ với tên **Bai tap 12.doc**

Phiếu đăng ký đại biểu

1/ Công Ty Quý Khách thuộc lĩnh vực nào?

- Viễn thông
- Phần cứng / linh kiện máy tính
- Giáo dục
- Ngân hàng & tài chính
- Cơ quan nhà nước
- Lĩnh vực khác

2/ Công Ty bạn có bao nhiêu nhân viên?

- Dưới 10
- Từ 11 – 50
- Từ 51 – 100
- Trên 100

MỤC LỤC

A. PHẦN I

I/ CHƯƠNG I.1: Thông tin và cách biểu diễn thông tin trong máy tính

- 1) Thông tin là gì?
- 2) Đơn vị đo thông tin

II/ CHƯƠNG I.2: Cấu thành một hệ thống máy tính

- 1) Thiết bị nhập
- 2) Thiết bị xuất
- 3) Bộ xử lý trung tâm
- 4) Thiết bị lưu trữ

III/ CHƯƠNG I.3: Các thiết bị lưu trữ

- 1) Đĩa mềm
- 2) Đĩa cứng
- 3) Đĩa CD-ROM

B. PHẦN II

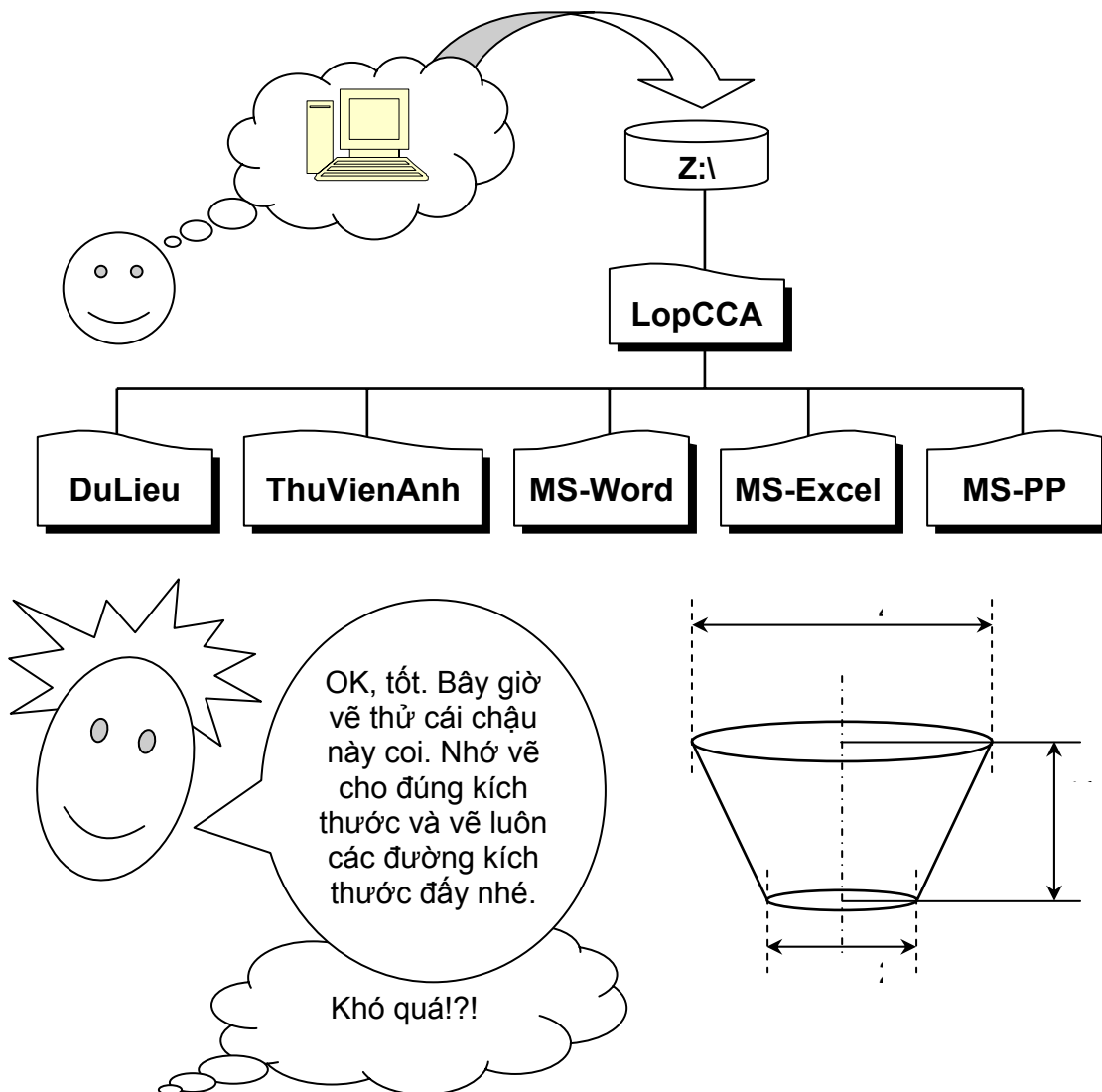
- I/ **CHƯƠNG II.1: Chương trình máy tính là gì?**
- II/ **CHƯƠNG II.2: Cấu trúc cơ bản của một chương trình Pascal**
- III/ **CHƯƠNG II.3: Các kiểu dữ liệu cơ bản**
- IV/ **CHƯƠNG II.4: Dữ liệu có cấu trúc**

C. PHẦN III

- I/ **CHƯƠNG III.1: Chương trình con là gì?**
- II/ **CHƯƠNG III.2: Các loại tham số hình thức**
- III/ **CHƯƠNG III.3: Kỹ thuật đệ quy**

Bài 13: Vẽ hình trong Word.**Yêu cầu:**

- Tạo file mới và lưu vào thư mục Z:\ với tên Bai tap 13.doc, nội dung trình bày như mẫu bên dưới.
- Sử dụng các tính năng xoay, lấy đối xứng (Rotate or Flip), gom nhóm các đối tượng (Group), thay đổi trật tự giữa các đối tượng (Order), sắp xếp và phân bố các đối tượng (Align or Distribute).



Bài 14: Sử dụng các công cụ hỗ trợ: Autocorrect, Find and Replace, đếm số từ (word count), bảo vệ nội dung văn bản (Security options).

Yêu cầu:

- Tạo file mới và lưu với tên là **Bai tap 14.doc**, với nội dung lấy từ một bài báo trên mạng .

Bài 15: Sử dụng Tab để trình bày văn bản (Format → Tab)

Yêu cầu:

- Tạo file mới và trình bày như mẫu bên dưới rồi lưu vào thư mục **Z:** với tên **Bai tap 15.doc**.

Điểm dừng Tab canh trái:

Lê Thành	Nam	HS phổ thông
Nguyễn Thị Kiều	Trang	Giáo viên
Hồ Trung	Hiếu	Sinh viên
Lê	Nguyễn	Sinh viên

Điểm dừng Tab canh phải:

Lê Thành	Nam	75.000 đ
Nguyễn Thị Kiều	Trang	120.000 đ
Hồ Trung	Hiếu	250.000 đ
Lê	Nguyễn	85.000 đ

Điểm dừng Tab canh giữa:

UBND Tỉnh An Giang	Cộng Hòa Xã Hội Chủ Nghĩa Việt Nam
Trường Đại học An Giang	Độc lập – Tự do – Hạnh phúc
----oOo ---	---- □ ----

Điểm dừng Tab thập phân:

Nhãn	70.5 kg
Táo	125.8 kg
Chôm chôm	18.3 tấn
Cà chua	30.4 tấn

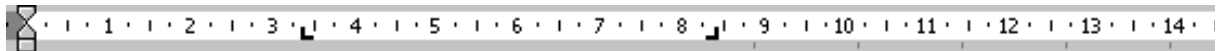
Thiết lập Leader cho khoảng Tab

Nhãn.....70.5 kg

Táo125.8 kg

Chôm chôm..... 18.3 tấn

Cà chua.....30.4 tấn



Lê Thành Nam ----- 75.000 đ

Nguyễn Thị Kiều Trang-----120.000 đ

Hồ Trung Hiếu-----250.000 đ

Lê Nguyễn ----- 85.000 đ



Chỉ mục	Trang
Lời nói đầu.....	2
Chương I.....	3
Chương II.....	7
Chương III.....	12

Tổng hợp:

UBND Tỉnh An Giang
Trường Đại học An Giang

---- oOo ----

Cộng Hòa Xã Hội Chủ Nghĩa Việt Nam
Độc lập – Tự do – Hạnh phúc

---- □ ----

GIẤY ĐỀ NGHỊ

Đề nghị cấp cho:

Đồng chí:..... Chức vụ:.....

Đồng chí:..... Chức vụ:.....

Đồng chí:..... Chức vụ:.....

Các tài khoản mail với các đặc trưng sau:

User name	Password ban đầu	Thời hạn
.....
.....
.....

Long xuyên, ngày..... tháng..... năm.....

Người đề nghị

Bài 16: Sử dụng bảng (Table) để trình bày văn bản.

Yêu cầu:

Trình bày theo mẫu của bảng điểm. Sau đó tính Điểm trung bình và sắp xếp lại danh sách học sinh tăng dần theo điểm trung bình.

BẢNG ĐIỂM

STT	Họ và tên	Toán	Lý	Hóa	Sinh	Văn	Ngoại ngữ	Điểm TB
1.	Trương Thị Mỹ	5	10	5	5	4	4	
2.	Nguyễn Thị Thu	9	6	7	4	9	5	
3.	Lý Thị Thu Hồng	5	8	7	5	9	4.5	
4.	Lê Thị Ngọc	9	8.5	10	9.5	8	9.5	
5.	Quách Anh Ngọc	4	7	6	4.5	9	7.5	
6.	Trần Thị Thanh	5	9	6	3	3	9	
7.	Bùi Thị Yến	8	8	9	5	10	10	
8.	Nguyễn Thanh Thảo	9	7	5	8	10	8	
9.	Nguyễn Phú Trung	3	5	10	9	5	6	
10.	Phạm Thị Thanh	7	9	7	7	8	4	
11.	Nguyễn Ngọc Ngạn	5	5	5	8	6	10	
12.	Lưu Thị Vân	5	3	3	4	6	5	





Yêu cầu: Trình bày theo mẫu văn bản sau

Thời khóa biểu

	Tiết	Thứ 2	Thứ 3	Thứ 4	Thứ 5	Thứ 6	Thứ 7	Chủ nhật
Sáng	1	Sinh hoạt	CNXH	Toán cao cấp A3	Anh văn 3	Vật lý đại cương	Cấu trúc dữ liệu 1	Không có lịch học chính thức
	2	Toán cao cấp A3				Tin học đại cương 3		
	3	Vật lý đại cương		Anh văn CN	Tài nguyên Internet			
	4							
	5							
Chiều	1	Thực hành (MT3)	Thực hành (MT3)	Giáo dục thể chất	Thực hành (MT3)	Thực hành (MT3)	Thực hành (MT3)	
	2							
	3							
	4							

Yêu cầu: Trình bày theo mẫu văn bản sau

**TRUNG TÂM TIN HỌC & TRUYỀN THÔNG
ĐẠI HỌC AN GIANG**

 Tin học căn bản	 Sử dụng Web Mail AGU	 Sử dụng FTP AGU để tải dữ liệu
 Sử dụng hệ thống tổng đài điện thoại nội bộ	 Soạn thảo văn bản với MS-WORD	 Tạo bảng tính với MS-EXCEL
 Sử dụng MS-POWER POINT để tạo file trình diễn, bài giảng	 Mã hóa thông tin - Cryptography	 Sử dụng trình thiết kế Web MS-FRONTPAGE

Yêu cầu: Trình bày theo mẫu phiếu trắc nghiệm sau

PHIẾU TRẢ LỜI TRẮC NGHIỆM

Họ và tên:

Mã số sinh viên:.....

Lớp:

Nếu chọn a	a	b	c	d
Nếu chọn c, bỏ a	a	b	c	d
Nếu chọn lại a bỏ c, bôi đen a	a	b	c	d

Câu	Trả lời			
1.				
2.				
3.				
4.				
5.				

Câu	Trả lời			
6.				
7.				
8.				
9.				
10.				

Yêu cầu: Trình bày văn bản theo mẫu sau

TRƯỜNG ĐẠI HỌC AN GIANG

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

TRUNG TÂM TIN HỌC

Độc lập - Tự do - Hạnh phúc

---*_*_*---



PHIẾU THÔNG BÁO KẾT QUẢ

Kỳ thi cấp Chứng chỉ Tin học ứng dụng Trình độ A Quốc gia

TRUNG TÂM TIN HỌC TRƯỜNG ĐẠI HỌC AN GIANG

TRÂN TRỌNG THÔNG BÁO ĐẾN

Thí sinh:

Số báo danh: Số CMND:

Sinh ngày:

Quê quán:

Địa chỉ hiện nay:

Đã tham dự **Kỳ thi cấp Chứng chỉ Tin học ứng dụng Trình độ A Quốc gia** với kết quả như sau:

<i>MS. Windows</i>	<i>MS. Word</i>	<i>MS. PowerPoint</i>	<i>MS. Excel</i>	<i>Tổng điểm</i>	<i>Kết quả</i>	<i>Ghi chú</i>

Trung tâm Tin học sẽ giải quyết các khiếu nại và nhận đơn chấm phúc khảo kể từ ngày tháng năm 2007 đến hết ngày tháng năm 2007. Điểm phúc khảo sẽ được công bố sau đó 1 tuần kể từ thời gian công bố chấm phúc khảo nêu trên.

TPLX, ngày tháng năm 2007

Giám đốc

(*Đã ký*)

Bài 17: Những tính năng mở rộng: công thức toán học Equation, vẽ biểu đồ MS-Graph.

Yêu cầu:

- Tạo file mới và trình bày như mẫu bên dưới rồi lưu vào thư mục **Z:** với tên **Bai tap 18.doc**.

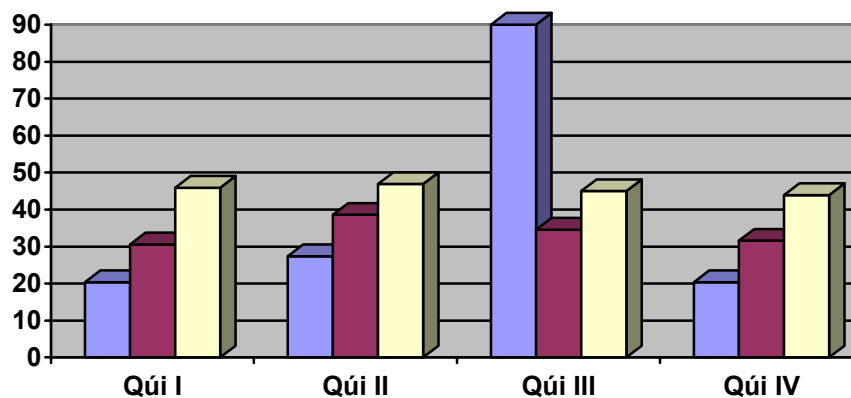
$$a) y = \frac{a^2 + b^2}{c^2}$$

$$b) \frac{x^m}{x^n} = x^{m-n}$$

$$c) y_1x_1 = y_2x_2 \Rightarrow \frac{y_1}{y_2} = \frac{x_2}{x_1}$$

$$d) \frac{a}{b} = \frac{ma}{mb}$$

$$e) \left(\left(-\frac{2}{3} \right)^2 \right)^3 = \left(-\frac{2}{3} \right)^6 \quad f) y = \sqrt[3]{ax^2 + bx^2 + cx + d}$$



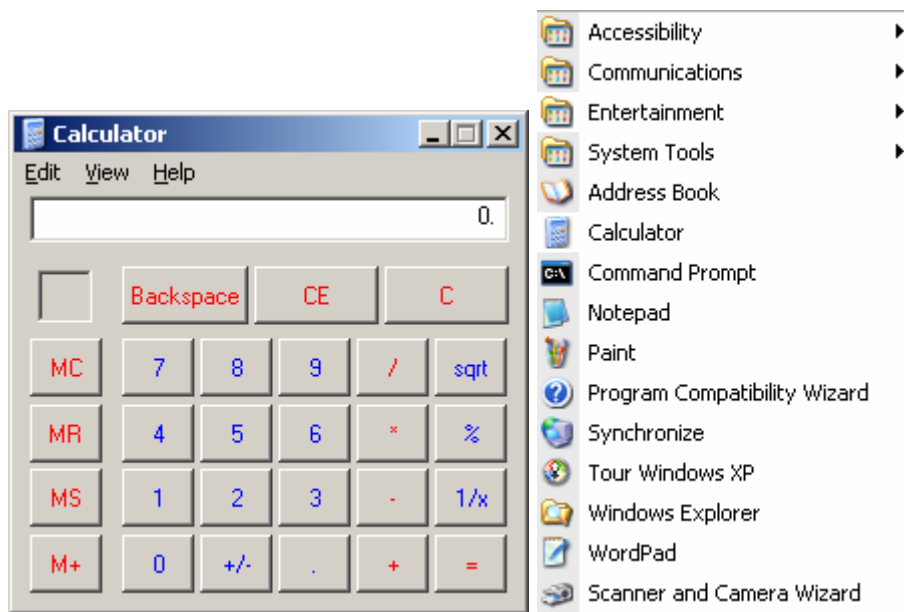
Bài 18: Tích hợp các ứng dụng trong windows.

Yêu cầu:

- Tạo file mới và trình bày như mẫu bên dưới rồi lưu vào thư mục **Z:** với tên **Bai tap 18.doc**.

Hướng dẫn: Sử dụng phím **Print Screen** để chụp hình ảnh trên Windows. Sau đó mở Microsoft Paint xử lý, và copy vào Word.





Soạn thư cảm ơn theo mẫu:

UBND TỈNH AN GIANG TRƯỜNG ĐẠI HỌC AN GIANG <hr/> Số: 517/TB-ĐHAG	CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM Độc lập - Tự do - Hạnh phúc <hr/> Long Xuyên, ngày 15 tháng 6 năm 2007
THƯ CẢM ƠN	
Kính gửi: Ban Giám đốc «Tên_cơ_quan»	
<p>Thực hiện nội dung chương trình đào tạo Cử nhân Tin học, vừa qua trường Đại học An Giang đã giới thiệu sinh viên «Họ_tên_SV» đến «Tên_cơ_quan» để thực tập tốt nghiệp.</p> <p>Được sự giúp đỡ tận tình và tạo điều kiện thuận lợi của lãnh đạo cùng với tập thể cán bộ, công chức «Tên_cơ_quan», đặc biệt là đồng chí «Họ_tên_CB_hướng_dẫn», người hướng dẫn trực tiếp giúp sinh viên «Họ_tên_SV» hoàn thành nhiệm vụ thực tập.</p> <p>Thay mặt nhà trường, tôi chân thành cảm ơn và ghi nhận sự giúp đỡ cùng với sự cộng tác có hiệu quả của «Tên_cơ_quan» trong công tác đào tạo của nhà trường. Mong rằng mối quan hệ giữa «Tên_cơ_quan» và trường Đại học An Giang ngày càng gắn bó và phát triển.</p> <p>Trân trọng kính chào.</p>	
HIỆU TRƯỞNG	

Tên cơ quan	Họ tên CB hướng dẫn	Họ tên SV
Sở Bru chính Viễn thông AG	Huỳnh Thiện Hiếu	Nguyễn Quốc Việt
Công ty TNHH Vĩnh Lợi	Nguyễn Văn Phúc	Võ Thị Phương Thảo
Công ty Xuất Nhập Khẩu AG	Lê Văn Tân	Huỳnh Thanh Tùng

2. Tạo thẻ khách hàng theo mẫu sau (chọn document type là Directory):

 KHÁCH HÀNG THÂN THIẾT 2007
Họ tên: Nguyễn Phước Đại Năm sinh: 1986 Địa chỉ: Mỹ Long, TP Long Xuyên
<i>Co.op Mart - Nơi mua sắm đáng tin cậy, bạn của mọi nhà</i>
 KHÁCH HÀNG THÂN THIẾT 2007
Họ tên: Nguyễn Phước Thuận Năm sinh: 1984 Địa chỉ: Mỹ Bình, TP Long Xuyên
<i>Co.op Mart - Nơi mua sắm đáng tin cậy, bạn của mọi nhà</i>

Bài tập tổng hợp

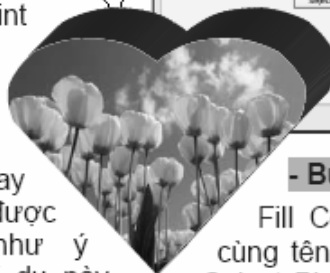
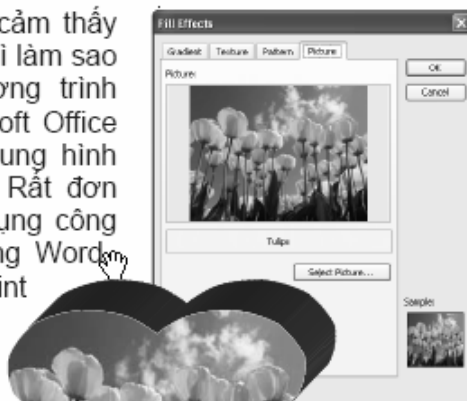
1. Yêu cầu: Trình bày văn bản như mẫu (Trích đề thi chứng chỉ Tin học A – Trung tâm tin học đại học An Giang)

- Font Arial Size 13pt
- Định dạng trang giấy: Top, Left, Right, Bottom: 2 cm
- Định dạng khoảng cách giữa các đoạn: Before, After: 6pt

Làm khung hình bằng...Microsoft Office.


Chắc bạn cảm thấy bất ngờ vì làm sao các chương trình trong bộ Microsoft Office có thể đóng khung hình cho ảnh được. Rất đơn giản, bạn tận dụng công cụ Drawing trong Word, Excel, PowerPoint để chèn các hình ảnh vào trong Autoshapes là bạn sẽ có ngay các hình ảnh được đóng khung như ý muốn. Trong ví dụ này, tôi sử dụng Microsoft Word để thực hiện:



- Bước 1: Mở thanh công cụ Drawing ra bằng cách nhấp chuột phải



vào thanh công cụ, chọn Drawing hoặc vào menu View > Toolbars, chọn Drawing. Lúc này, bạn sẽ thấy thanh công cụ Drawing xuất hiện phía dưới màn hình.

- Bước 2: Nhấp chuột chọn Autoshapes trên thanh công cụ Drawing. Trong danh sách các mẫu hình xuất hiện bạn hãy chọn một chủ đề tùy thích, tôi chọn Basic Shapes và chọn hình trái tim.

- Bước 3: Nhấp chuột vào hình mũi tên kế bên nút Fill Colors , chọn Fill Effects để mở hộp thoại cùng tên. Bạn chọn thẻ Picture rồi nhấp chuột vào nút Select Picture, duyệt đến thư mục chứa hình ảnh muốn làm khung ảnh. Nhấn OK để thấy kết quả.

- Bước 4: Bạn có thể trang điểm cho bức ảnh bằng hiệu ứng đổ bóng (Shadow Style ) hay bằng hiệu ứng 3 chiều (3-D Style )

VĂN VỮNG (Nha Trang)

2. Yêu cầu: Trình bày như mẫu văn bản(Trích đề thi chứng chỉ Tin học A – Trung tâm tin học Đại học An Giang)

- Font Arial Size 13pt
- Định dạng trang giấy: Top, Left, Right, Bottom: 2 cm
- Định dạng khoảng cách giữa các đoạn: Before, After: 6pt

TRÒ LỪA BÁO TIN VIRUS PHÁ Ổ CỨNG TRÊN YAHOO MESSENGER

Hãng bảo mật Sophos (Anh) cho biết một kiểu bịp mới đang xuất hiện phổ biến trong cộng đồng người dùng phần mềm nhắn tin nhanh (IM) của Yahoo.


Theo Sophos, người dùng Yahoo IM nên cảnh giác với những ai cố tình liên hệ với họ qua công cụ này. Thông điệp bịp bợm nói rằng máy tính của người nhận tin sẽ bị nhiễm một loại virus “rất khủng khiếp”, xóa sạch ổ cứng (Hard drive killer) nếu như họ cho phép một người có địa chỉ `dvorak@yahoo.com` đưa (add) nickname của bạn vào danh sách của hắn ta. Thông báo này yêu cầu người nhận hãy loãn tin, cảnh báo tất cả những ai có trong danh sách liên hệ ở Yahoo Messenger của họ đừng mở bất cứ cái gì xuất xứ từ một số cái tên như “angell11”, “tewwtuler” và “sassybitch”. Kết thúc thông báo là lời xin lỗi của `tanwir2001`, tự xưng là đại diện dịch vụ của Yahoo.

Đến thời điểm này, số thông báo về trò lừa nói trên mà những khách hàng lâm tương là có virus và gửi về cho Sophos đang tăng nhanh. Thông tin này còn được phát tán qua cả e-mail và một số mạng tin trực tuyến.



“Đây đơn giản là một tin nhảm”, chuyên gia Graham Cluley của Sophos khẳng định. “Tuy nhiên, những trò lừa kiểu chuyên tiếp thông điệp như thế này không hẳn là vô hại vì chúng làm tiêu tốn thời gian và băng thông mạng, gây đau đầu cho các bộ phận hỗ trợ kỹ thuật. Người sử dụng cần để ý kỹ một điều: có nên tin vào bất kỳ cái gì mà mình được thông báo hay không”.

Một số tin liên quan đến Virus.

	TÊN BÀI VIẾT	NGÀY
	Virus giả dạng e-mail có video về Bin Laden	05/11
	Hacker bắt trọng tài phải xin lỗi	23/10
	Sâu Zafi.C nhắm vào thủ tướng Hungary	29/10
	Từ chức vì để máy tính bị xâm nhập	14/10
	Virus tấn công dịch vụ IM của Microsoft	12/10

3. Yêu cầu: Trình bày như mẫu văn bản (Trích đề thi chứng chỉ Tin học A – Trung tâm tin học Đại học An Giang)

- Font Arial Size 13pt
- Định dạng trang giấy: Top, Left, Right, Bottom: 2 cm
- Định dạng khoảng cách giữa các đoạn: Before, After: 6pt

Bão hình thành như thế nào ?

Bão chỉ hình thành trên vùng đại dương nhiệt đới, nơi có vùng nước ấm, tối thiểu là 26 độ C, không khí ẩm ướt và gió hội tụ. Một trận bão bắt đầu từ trước tiên là những đám mây tạo mưa dông. Trên đất liền, khi có mưa dông ta thấy bao giờ cũng có những trận gió lốc rất mạnh, do sự chênh lệch áp suất không khí giữa áp cao do lạnh giữa đám mây mưa và bầu không khí nóng chung quanh.

Trên mặt biển, nếu có hai cơn mưa dông gặp nhau, những luồng gió khi gặp nhau sẽ bốc lên cao theo luồng hơi nước bốc lên mặt biển ấm. Và ở tầng trên của lớp đối lưu, luồng



khí ẩm ướt này toả ra và bắt đầu xoay theo quán tính hình thành từ chiều quay của trái đất. Nếu hiện tượng này tiếp tục thì vận tốc quay sẽ ngày càng tăng dần, những đám mây đầy hơi nước cũng lớn dần, chúng cần phải tăng tốc xoay tròn do sự tản ra khi gặp tầng bình lưu ở độ cao 16km. Và một cơn bão hình thành.

Khi dòng không khí xoay tròn đạt đến tốc độ trên 118km/giờ thì một vùng khí áp cực thấp hình thành ở trung tâm dòng xoáy, ta hay gọi là mắt bão, và hút toàn bộ bầu không khí ẩm ướt trong khu vực lên cao, bổ

sung lượng hơi nước, hình thành nên những đám mây ngày càng to mà mưa rơi xuống bao nhiêu cũng không rơi cạn. Như vậy, mắt bão chỉ hình thành khi tốc độ gió đạt đến trên cấp 10, 11 như cách phân loại các tốc độ gió bão ở ta .

Mặt biển càng ấm, lượng nước bốc hơi càng nhiều thì lượng không khí ẩm bổ sung cho dòng xoáy càng nhiều và vì thế gió cũng tăng tốc. Vì vậy có thể nói, lượng nước bốc hơi chính là năng lượng của các cơn bão. Năng lượng này đôi khi đạt đến mức tương đương 5 quả bom hạt nhân/giây. Nhưng ngay khi gặp một dòng nước lạnh hơn hoặc gặp đất liền, bão giảm cường độ vì thiếu khí nóng ẩm bốc hơi ☐.

Phần 4.

MICROSOFT EXCEL 2003

Chương 1.


GIỚI THIỆU

1.1. GIỚI THIỆU

Microsoft Excel 2003 là một chương trình xử lý bảng tính điện tử trong bộ Microsoft Office 2003. Với Microsoft Excel ta có thể thực hiện các phép tính số học, sắp xếp dữ liệu, tổng hợp và thống kê dữ liệu, lọc trích các danh sách, tạo biểu đồ (Chart)...

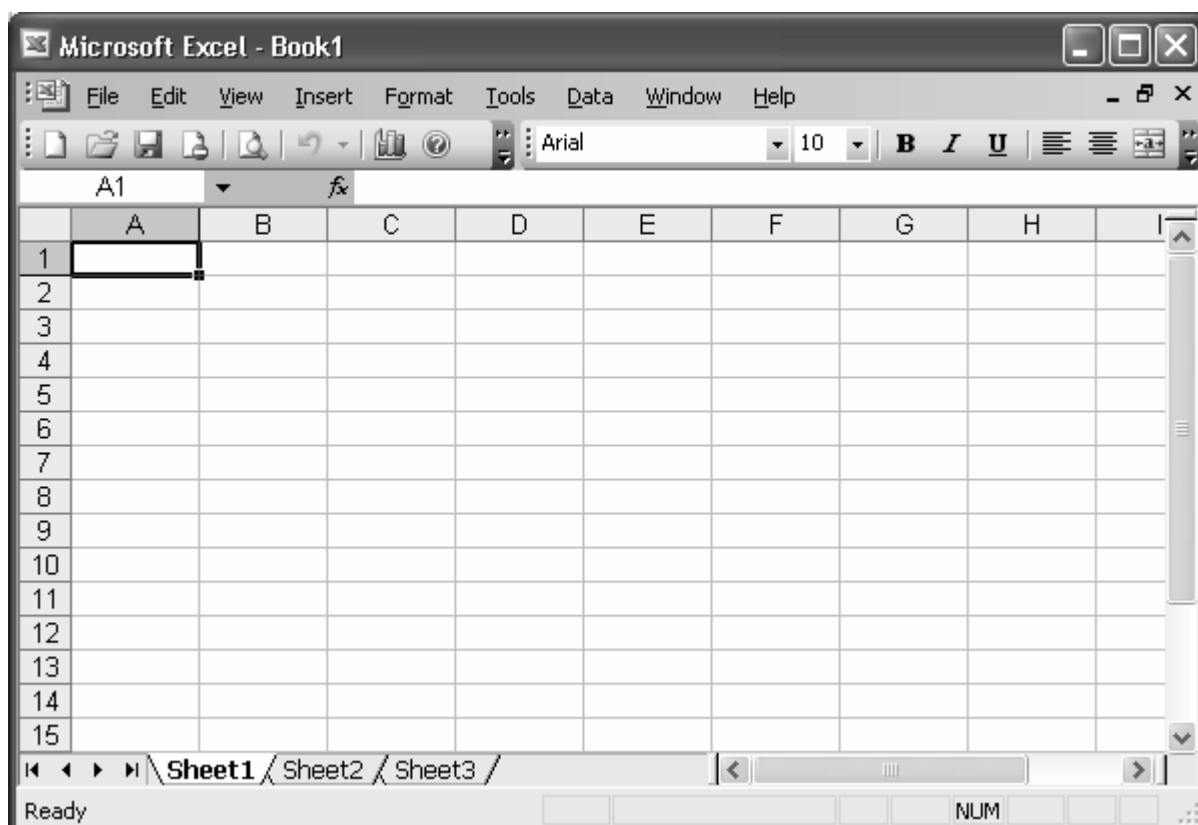
1.2. Khởi động và Thoát

1.2.1. Khởi động

↪ **Cách 1:** Click chuột vào biểu tượng  trên desktop.

↪ **Cách 2:** Chọn menu *Start/Programs/Microsoft Office/Microsoft Office Excel*.

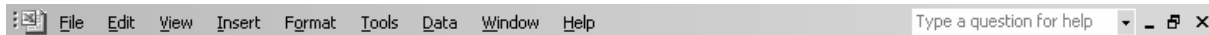
Cửa sổ giao diện của Excel có dạng như sau:



Hình 4.1 Màn hình Excel



- **Thanh tiêu đề (Title bar):** Chứa tên chương trình Microsoft Excel, tên workbook hiện hành. Bên trái là hộp chứa menu điều khiển (*control menu box*), bên phải có các nút **phóng to/phục hồi** (*maximize/restore*), **thu nhỏ** (*minimize*) và **đóng cửa sổ Excel** (*close*).



- **Thanh thực đơn (Menu bar):** Chứa các lệnh chính của Excel trong các menu *File, Edit, ..., Windows*. Các menu được kích hoạt bằng cách click chuột, nhấn phím *Alt* hoặc phím *F10* kết hợp với các ký tự gạch chân trên các menu.

- **Các thanh công cụ (Toolbars):** Chứa các lệnh thông dụng của Excel dùng để thao tác nhanh bằng cách click chuột. Có tất cả 13 thanh công cụ mặc định trong Excel, mỗi thanh công cụ tương ứng với một chức năng của Excel. Hai thanh công cụ thường dùng là thanh công cụ chuẩn (*standard*) và định dạng (*formatting*). Có thể bật/tắt các thanh công cụ bằng lệnh *View\Toolbars* hoặc click phải chuột vào vùng chứa các thanh công cụ, sau đó chọn bật/tắt từng thanh.

- **Dòng nhập (Formula bar):** Chứa địa chỉ và nội dung của ô hiện hành.

- **Thanh trượt dọc (Vertical scroll bar) và trượt ngang (Horizontal scroll bar):** Dùng để cuộn bảng tính theo chiều dọc và chiều ngang.


1.2.2. Thoát khỏi Excel

Giống như Word, để thoát khỏi Excel ta có thể thực hiện theo các cách: chọn menu **File\Exit** hoặc nhấn **ALT+F4**, ...


1.3. QUẢN LÝ WORKBOOK

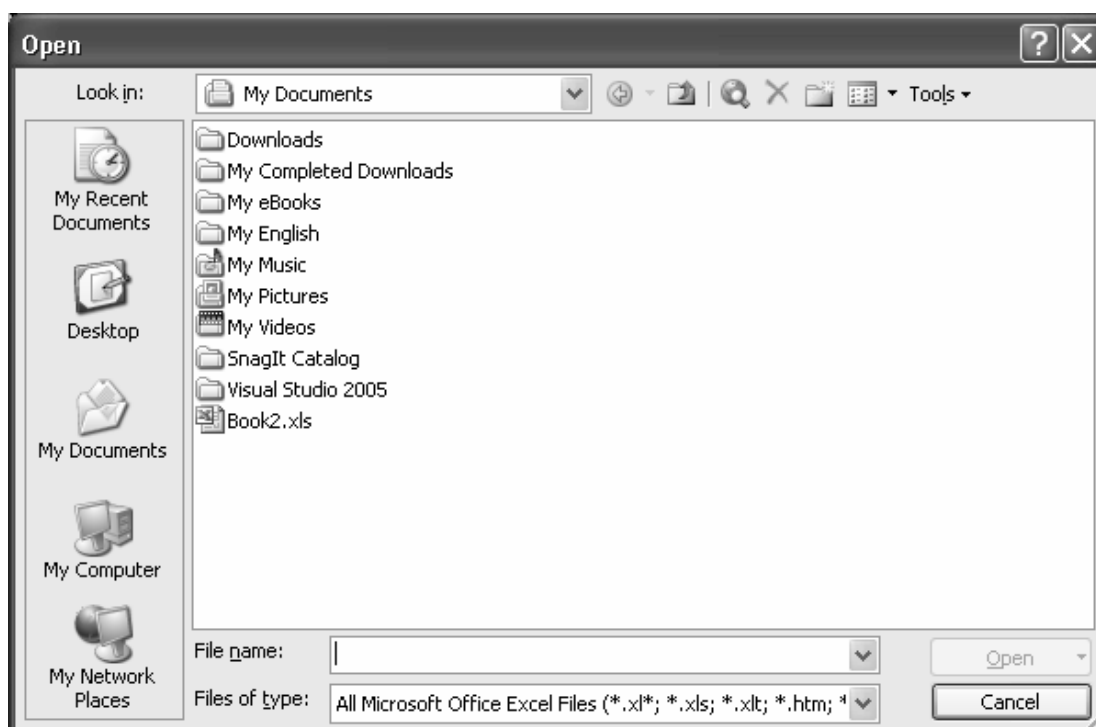
Một tập tin của Excel được gọi là **workbook** và có phần mở rộng mặc nhiên là **.XLS**. Một workbook gồm nhiều **sheet**. Một workbook có thể chứa tối đa 255 sheets.

1.3.1. Mở một bảng tính mới: Có các cách chính sau:

- Chọn biểu tượng .
- Gõ **Ctrl+N** hoặc chọn menu **File\New**.


1.3.2. Mở một tập tin đã có trên đĩa (Open).

Click nút lệnh  hoặc gõ **Ctrl+O** hoặc chọn menu **File\Open** để mở hộp thoại **Open**. Xác định đường dẫn đến tập tin cần mở và click **Open**.

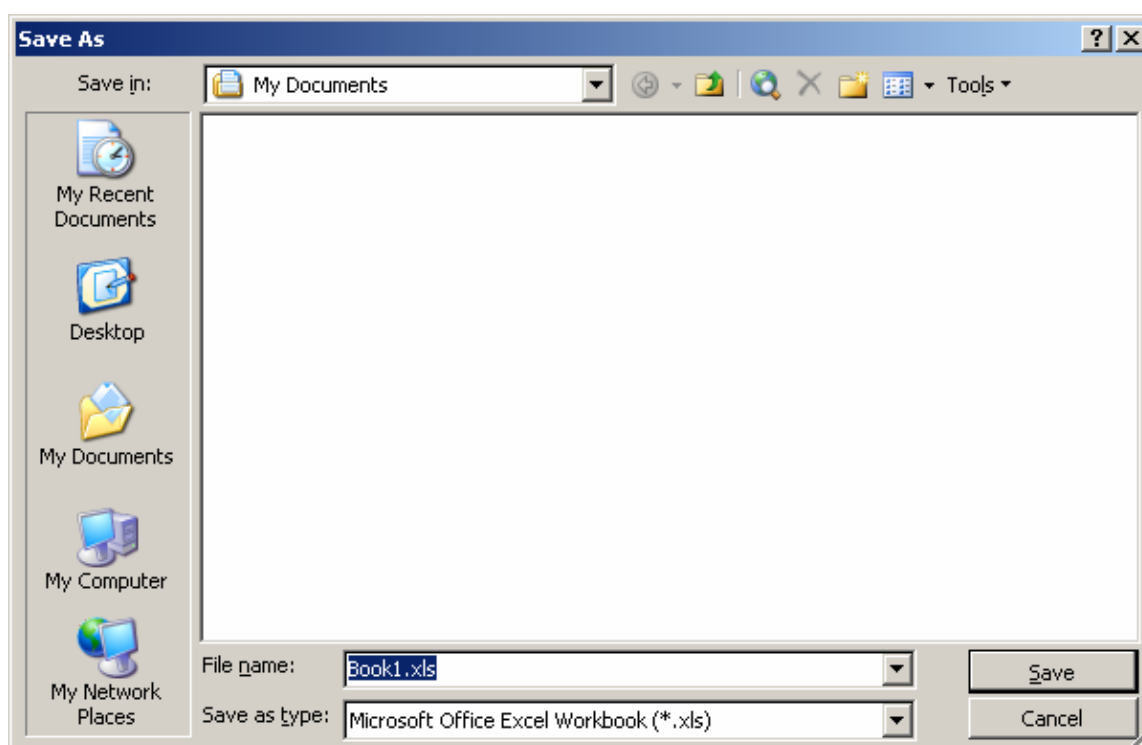


Hình 4.2. Mở một tập tin Excel

1.3.3. Lưu tập tin (Save).

Bước 1: Nhấp vào biểu tượng , hoặc nhấn **Ctrl+S** hoặc chọn menu **File\Save**.

Bước 2: Nếu lưu tập tin lần đầu, Excel sẽ mở hộp thoại *Save as*. Bạn cần xác định thư mục chứa tập tin và tên tập tin rồi click **Save**.



Hình 4.3. Lưu một tập tin Excel

Nếu tập tin đã có trên đĩa, Excel tự động lưu các thay đổi.

1.3.4. Lưu tập tin với tên khác (Save as)

Muốn lưu tập tin đã có thành một tập tin khác, chọn menu **File/Save As** để mở hộp thoại **Save as** rồi thực hiện tương tự như trên.

1.4. BẢNG TÍNH (WORKSHEET)

1.4.1. Cấu tạo của worksheet

- *Sheet* là một trang bảng tính, cấu trúc của một sheet là một bảng gồm nhiều *cột (column)* và nhiều *dòng (row)*. Mỗi sheet có 65536 dòng được đánh số từ 1, 2, 3, ..., 65536 và 256 cột được đánh số từ A, B, C, ..., Z, AA, AB, AC, ..., IV. Độ rộng của cột và dòng có thể thay đổi được.

- *Ô (Cell)*: Giao giữa cột và dòng tạo thành ô. Địa chỉ của một ô được xác định bởi tên cột trước và chỉ số dòng sau, đây cũng chính là *tên của ô*.

Ví dụ: Ô tại giao điểm giữa *cột B* và *dòng 9* sẽ có địa chỉ là **B9**, còn gọi là *ô B9*.

- *Vùng (Range)* là các ô liền kề nhau có dạng hình chữ nhật tạo thành một vùng. Địa chỉ của một vùng được xác định bởi địa chỉ của ô *góc trên bên trái* và địa chỉ của ô *góc dưới bên phải*, giữa địa chỉ của 2 ô có dấu : (dấu hai chấm).

Ví dụ: Vùng A2:B5 bao gồm các ô: A2, A3, A4, A5, B2, B3, B4, B5.

- *Khối (Block)*: là các ô được chọn để xử lý, có thể liên tiếp hoặc không liên tiếp, một khối ô có thể gồm nhiều vùng.

↳ **Ô hiện hành.**

Là ô đang chứa con trỏ. Ô hiện hành còn được gọi là ô hoạt động (active cell) hoặc là ô được chọn.

1.4.2. Địa chỉ tham chiếu

Khi muốn sử dụng giá trị của một ô hoặc một vùng vào biểu thức, ta đưa tên ô hoặc tên vùng vào trong biểu thức, các tên này được gọi là địa chỉ tham chiếu. Trong Excel có 3 loại địa chỉ tham chiếu:

- **Địa chỉ tham chiếu tương đối**: là loại địa chỉ sẽ bị thay đổi khi ta thực hiện các thao tác di chuyển, chép, ...

- **Địa chỉ tham chiếu tuyệt đối**: là loại địa chỉ không bị thay đổi khi ta thực hiện các thao tác di chuyển, chép, ... Muốn sử dụng địa chỉ tuyệt đối ta thêm ký hiệu \$ vào trước chỉ số cột và chỉ số dòng.

- **Địa chỉ tham chiếu hỗn hợp**: là loại địa chỉ được kết hợp giữa địa chỉ tham chiếu tương đối và địa chỉ tham chiếu tuyệt đối.

Ví dụ: Ta có các tham chiếu sau:

A4 + 2 tham chiếu tương đối đến ô A4.

\$A\$4 + 2 tham chiếu tuyệt đối đến ô A4.

\$A4 + 2 tham chiếu tuyệt đối cột và tương đối dòng đến ô A4.

A\$4 + 2 tham chiếu tương đối cột và tuyệt đối dòng đến ô A4.

1.4.3. Các thao tác trên worksheet

① **Chọn một worksheet:** Click chuột vào worksheet cần chuyển đến. hoặc nhấn tổ hợp phím *Ctrl+PgUp* (hoặc *Ctrl+PgDn*) để bật chuyển đến worksheet trước (hoặc sau) worksheet hiện hành.

② **Đổi tên worksheet:** Theo mặc định, các worksheet được đặt tên là sheet1, sheet2,... Để đổi tên worksheet, click đúp chuột vào tên worksheet cũ rồi nhập tên mới vào.

③ **Di chuyển và sao chép worksheet:** Nhấn chuột vào worksheet muốn di chuyển rồi rê chuột đến vị trí mới. Nếu muốn sao chép thì nhấn thêm phím **Ctrl** trong khi rê chuột.

④ **Chèn thêm worksheet:** Thực hiện lệnh **Insert\Worksheet**.

⑤ **Xóa worksheet:** Chọn worksheet cần xóa và thực hiện lệnh **Edit>Delete Sheet**.

⑥ **Di chuyển ô hiện hành.**

+ Dùng phím:

Phím	Công dụng
→, ←, ↑, ↓	Qua phải, qua trái, lên, xuống.
PageUp	Cuộn lên một trang.
PageDown	Cuộn xuống một trang.
Ctrl+Home	Về thẳng ô A1.

+ Dùng chuột: Click vào ô cần chuyển đến.

1.4.4. Chọn ô và vùng trong worksheet.

– *Chọn một vùng:* có 2 cách dùng bàn phím và dùng chuột:

↳ *Cách 1:*

- Đặt con trỏ ở ô đầu tiên hoặc ô cuối cùng của vùng định chọn.
- Nhấn giữ phím Shift và dùng các phím →, ←, ↑, ↓ để xác định vùng chọn.

↳ *Cách 2:*

- Chuyển trỏ chuột đến ô đầu tiên hoặc ô cuối cùng của vùng định chọn.
- Rê chuột để xác định vùng định chọn.

– *Chọn đồng thời nhiều ô cách quãng* (hoặc chọn nhiều vùng khác nhau): nhấn giữ phím *Ctrl* trong khi chọn.

– *Chọn nguyên cột:* click chuột lên thanh ghi tiêu đề của cột cần chọn.

– *Chọn nguyên dòng:* click chuột lên thanh ghi tiêu đề của dòng cần chọn.

– *Chọn toàn bộ sheet:* có 2 cách:

- Nhấn tổ hợp phím **Ctrl+ A**.
- Nhấp vào điểm giao giữa thanh tiêu đề cột và chỉ số dòng.

Chương 2.

NHẬP VÀ XỬ LÝ DỮ LIỆU

2.1. CÁC KIỂU DỮ LIỆU VÀ TOÁN TỬ

Kiểu dữ liệu là sự qui định về hình dáng, cách hiển thị dữ liệu cũng như cách xử lý dữ liệu.

2.1.1. Các kiểu dữ liệu

↳ Dữ liệu kiểu số (Numeric).

Là kiểu dữ liệu chỉ chứa các số thuần túy gồm 0, 1, 2, ..., 9, ngoài ra còn có các dấu dương (+), dấu âm (-), dấu đôla (\$), dấu chấm thập phân.

↳ Dữ liệu kiểu ngày (Date)

Kiểu ngày là kiểu dữ liệu có chứa ba tham số ngày, tháng và năm. Ba tham số được phân cách với nhau bởi một ký hiệu phân cách được qui định bởi mục Date separator trong Control Panel/Regional and language options/Customize/Date. Theo mặc định ký hiệu / là ký hiệu phân cách kiểu ngày.

Trong Excel xem các dữ liệu kiểu ngày và kiểu giờ như là một trị số, do đó chúng có thể tham gia vào các biểu thức như là một trị số. Ngày 01/01/1900 sẽ có giá trị 1, ngày 02/01/1900 có giá trị 2, ..., ngày 21/03/1979 có giá trị 28935 (nghĩa là tính từ đầu thế kỷ đến ngày 21/03/1979 thì được 28935 ngày).

↳ Dữ liệu kiểu giờ (Time)

Là kiểu dữ liệu có chứa các tham số: giờ, phút, giây, phần trăm giây. Theo mặc định ký hiệu : (dấu hai chấm) là ký hiệu phân cách kiểu giờ.

↳ Dữ liệu kiểu logic

Là kiểu dữ liệu chỉ có thể có một trong hai giá trị hoặc là TRUE hoặc là FALSE. Đối với trị là TRUE được hiểu là đúng, và FALSE được hiểu là sai.

↳ Dữ liệu kiểu chuỗi (Text)

Nếu dữ liệu được nhập vào không phải là kiểu số, kiểu ngày, kiểu giờ, kiểu logic thường thì Excel sẽ hiểu đó là dữ liệu kiểu chuỗi. Theo mặc định dữ liệu kiểu chuỗi sẽ được canh trái trong ô.

Ta có thể báo cho Excel biết dữ liệu nhập vào là kiểu chuỗi bằng cách nhập thêm ký tự “ (dấu nháy đôi) ở đầu ô.

2.1.2 Các toán tử.

Đối với mỗi kiểu dữ liệu ta sẽ có các phép toán tương ứng thông qua các toán tử. Các toán tử dùng để kết hợp các toán hạng trong biểu thức. Khi tính trị của một biểu thức trong Excel, cần tuân theo quy định của Excel về độ ưu tiên của các toán tử.

ĐỘ ƯU TIÊN	TOÁN TỬ	Ý NGHĨA
1	()	Dấu ngoặc đơn
2	^	Lũy thừa
3	-	Dấu âm
4	*, /	Nhân, chia
5	+, -	Cộng trừ
6	=, <>	Bằng nhau, khác nhau
	>, >=	Lớn hơn, lớn hơn hoặc bằng
	<, <=	Nhỏ hơn, nhỏ hơn hoặc bằng
7	NOT	Phủ định
8	AND	Và (điều kiện đồng thời)
9	OR	Hoặc (điều kiện không đồng thời)
10	&	Toán tử ghép chuỗi

Ví dụ: $3^2 * 5 + 12 * 5 + (25-10)/3 = 9*5 + 60 + 5/3 = 45 + 60 + 5 = 110$

Trong trường hợp, biểu thức có nhiều cặp ngoặc đơn lồng vào nhau thì sẽ ưu tiên tính từ trong ra ngoài; nếu có nhiều toán tử cùng độ ưu tiên, sẽ được tính từ trái qua phải.

2.2. NHẬP DỮ LIỆU

Dữ liệu khi được đưa vào trong ô dưới 2 dạng: Hằng số (constant) và công thức (formula).

☞ Hằng số là đại lượng có giá trị không đổi, có kiểu dữ liệu thuộc 5 kiểu dữ liệu cơ bản (số, ngày, giờ, chuỗi và logic).

- Khi nhập dữ liệu dạng hằng số, ta nhập trực tiếp vào ô.

- Khi muốn nhập một biểu thức vào ô, ta phải nhập thêm dấu bằng (=) vào trước biểu thức. Khi đó dữ liệu trong ô này được hiểu là dạng công thức.

Đối với các ô dạng công thức, thì giá trị xuất hiện trong ô là kết quả của công thức. Nếu trong công thức có các toán hạng là địa chỉ tham chiếu thì giá trị của công thức có thể thay đổi tùy thuộc vào giá trị của ô được tham chiếu.

Ví dụ: Giả sử ô B5 có công thức là =A5 + 20

- Nếu A5 bằng **10** thì giá trị của ô B5 là **30**

- Nếu đổi giá trị ô A5 bằng **15** thì giá trị của ô B5 là **35**

☞ **Lưu ý:**

- Khi nhập dữ liệu kiểu ngày phải theo qui ước đã định trong **Control Panel/Regional and language options/Customize/Date**.

- Mặc nhiên, dữ liệu chuỗi trong ô sẽ được canh trái, các kiểu còn lại sẽ được canh phải. Tuy nhiên, chúng ta có thể định dạng lại cách canh lề tùy theo ý muốn.

- Thông thường, Excel có thể nhận định được kiểu của các dữ liệu nhập vào. Việc xác định chính xác kiểu của các dữ liệu trong ô là rất cần thiết cho việc xử lý số liệu và áp dụng các hàm trong Excel.

2.3. XỬ LÝ DỮ LIỆU

2.3.1. Xóa dữ liệu trong khối

Bước 1: Chọn khối muốn xóa.

Bước 2: Gõ phím **Delete** hoặc chọn menu **Edit/Clear/Contents**.

2.3.2. Sao chép (copy) khối dữ liệu

Bước 1: Chọn khối dữ liệu cần chép (khối nguồn).

Bước 2: Gõ tổ hợp phím **Ctrl+C** hoặc chọn menu **Edit/Copy**.

Bước 3: Đưa con trỏ tới ô đích gõ **Ctrl+V** hoặc chọn menu **Edit/Paste**.

2.3.3. Di chuyển khối dữ liệu

Bước 1: Chọn khối dữ liệu cần di chuyển.

Bước 2: Gõ tổ hợp phím **Ctrl+X** hoặc chọn menu **Edit/Cut**.

Bước 3: Đưa con trỏ tới ô đích gõ **Ctrl+V** hoặc chọn menu **Edit/Paste**.

2.4. SỬ DỤNG CÔNG THỨC

2.4.1. Nhập công thức

Thành phần của một công thức được bắt đầu bằng dấu =, tiếp theo là một toán hạng hoặc nhiều toán hạng phân cách nhau bởi các toán tử (nếu có). Mỗi toán hạng có thể là:

- 1) Một hằng số (số hoặc chuỗi ký tự).
- 2) Một tham chiếu tới địa chỉ ô/khối ô trong bảng tính.
- 3) Một hàm của Excel.

*** Lưu ý:**

- Không được có khoảng trắng trong công thức trừ khi nó nằm trong một hằng chuỗi ký tự.

- Sau khi nhập công thức, nội dung thực sự của ô là công thức mà bạn đã nhập nhưng giá trị thể hiện trong ô lại là kết quả mà công thức trả về. Muốn xem công thức vừa nhập, bạn phải vào chế độ chỉnh sửa nội dung của ô (phím F2) hoặc quan sát trên thanh công thức.

*** Ví dụ:**

① =IF(LEFT(B3)="K";1.25;1*VLOOKUP(MID(B3;2;1);\$B\$15:\$D\$17;3;0).

② =E4*D4.

③ E5*500000+F5*90000.

④ =210000*F4/22.

⑤ =A3&"An Giang".

2.4.2. Sao chép công thức

Bạn có thể sao chép công thức tương tự như sao chép dữ liệu. Ngoài ra, bạn có thể click chuột vào Fill handle của ô chứa công thức rồi rê nó xuống dưới hoặc sang phải theo hướng muốn sao chép.

2.5. HÀM (FUNCTION) TRONG EXCEL

2.5.1. Định nghĩa

Hàm là các công thức phức tạp được định nghĩa sẵn dùng để thực hiện các tính toán cho một vùng giá trị xác định, được cài đặt chung với phần mềm và được coi là tiện ích của phần mềm đó. Mỗi hàm là một công cụ nhằm giải quyết một công việc nhất định. Mỗi hàm cần cung cấp các đối số theo một trật tự cú pháp nhất định. Hàm có thể trả về một giá trị, một chuỗi hoặc một thông báo lỗi.

* **Thí dụ:** Hàm SUM dùng để cộng các giá trị hoặc một dãy các ô trong bảng tính.

2.5.2. Cú pháp chung của các hàm

Cú pháp của các hàm bắt đầu bằng tên hàm, các dấu mở ngoặc đơn, các đối số của hàm được phân cách nhau bằng các dấu phẩy và cuối cùng là dấu đóng ngoặc đơn.

Tuy nhiên, có một số hàm không có các đối số như hàm PI(), hàm NOW(), hàm DATE(), ...

Các đối số có thể là các số, chuỗi ký tự, các giá trị logic TRUE hoặc FALSE, các mảng dữ liệu, các thông báo lỗi hoặc các tham chiếu đến các ô khác trong bảng tính. Ngoài ra, các đối số cũng có thể là các hằng số, các công thức, hoặc các hàm khác.

* **Thí dụ:**

- Các trị số: SUM(10;20;50).
- Địa chỉ các ô: SUM(A1;A2;B1;B2).
- Địa chỉ một vùng: MAX(B1:B10).
- Một chuỗi ký tự: RIGHT("Trung Tâm Tin Học";7).
- Một biểu thức logic: IF(B1>B2; B1;B2).
- Một hàm khác: IF(A1>SQRT(B1);A1;A2).

2.6. CÁC HÀM THÔNG DỤNG

2.6.1. Các hàm toán học

① Hàm **ABS**: Cho trị tuyệt đối của trị số X.

- Cú pháp: ABS(X)
- Ví dụ: =ABS(5-15) = 10

- ② Hàm **INT**: Cho trị là phần nguyên của trị số X.
- Cú pháp: INT(X)
 - Ví dụ: =INT(8,9) → 8 hoặc =INT(-8,9) → -9
- ③ Hàm **ODD**: Cho trị là số nguyên lẻ lớn hơn hoặc bằng X.
- Cú pháp: ODD (X)
 - Ví dụ: =ODD(1,5) → 3 hoặc ODD(3) → 3
- ④ Hàm **MOD**: Cho trị là phần dư của phép chia: trị số X chia trị số Y.
- Cú pháp: MOD (X; Y): X là số bị chia, Y là số chia.
 - Ví dụ: =MOD(30; 4) → 2
- ⑤ Hàm **ROUND**: Làm tròn trị số X đến n vị trí chỉ định tính từ cột hàng đơn vị.
- Cú pháp: ROUND (X; n): X là số cần làm tròn, n là số lượng số muốn làm tròn.
 - + Nếu $n > 0$ làm tròn về bên phải tính từ cột hàng đơn vị.
 - + Nếu $n < 0$ làm tròn về bên trái tính từ cột hàng đơn vị.
 - + Nếu $n = 0$ làm tròn đến cột hàng đơn vị (không lấy số lẻ).
 - Ví dụ:
 - + ROUND(2,15; 1) → 2,2
 - + ROUND(2,149; 1) → 2,1
 - + ROUND(-1,475; 2) → 1,48
 - + ROUND(21,5; -1) → 20

2.6.2. Các hàm logic

① Hàm IF

- Cú pháp: IF(**biểu thức luận lý; biểu thức 1; biểu thức 2**)

- Công dụng: Biểu thức luận lý chỉ cho 2 trị: TRUE hoặc FALSE. Hàm sẽ cho kết quả là biểu thức 1 nếu biểu thức luận lý có giá trị TRUE, ngược lại sẽ cho kết quả là biểu thức 2.

☞ Hàm IF có thể là đối số cho một hàm IF khác để áp dụng vào các điều kiện phức tạp.

Ví dụ: Giả sử giá trị trong ô D1 = 6. Ta có công thức:

= IF(D1>=5; "Đậu"; "Rót") → "Đậu"

= IF (C1>0; "số dương"; "số âm") giá trị trả về phụ thuộc vào C1.

② Hàm **OR**: Trả về giá trị TRUE nếu có ít nhất một biểu thức luận lý cho trị TRUE.

- Cú pháp: OR(**biểu thức luận lý 1; biểu thức luận lý 2;...**)

- Ví dụ:

= OR (3>4; 4<5; 8 =10) → TRUE

= OR (3>4; 4>5; 8=10) → FALSE

③ **Hàm AND:** Trả về giá trị TRUE nếu tất cả các đối số đều có giá trị TRUE.

- **Cú pháp: AND (biểu thức luận lý 1; biểu thức luận lý 2;...)**

- **Ví dụ:**

= AND (3<4;4<5;8=8) → TRUE

= AND (3<4;4>5;8=8) → FALSE

④ **Hàm NOT:** Cho trị logic ngược lại với trị của biểu thức luận lý.

- **Cú pháp: NOT (biểu thức luận lý)**

- **Ví dụ:**

= NOT(TRUE) → FALSE

= NOT(2*2=3) → TRUE

2.6.3. Các hàm xử lý chuỗi

① **Hàm LEFT:** Trả về n ký tự bên trái của chuỗi text.

- **Cú pháp: LEFT(text; n)**

- **Ví dụ:** = LEFT("Nguyen Thanh Hoang"; 6) → "Nguyen"

② **Hàm RIGHT:** Trả về n ký tự bên phải của chuỗi text.

- **Cú pháp: RIGHT(text; n)**

- **Ví dụ:** = RIGHT("Bo mon Tin hoc"; 7) → "Tin hoc"

③ **Hàm MID:** Trả về n ký tự bắt đầu tại vị trí m trong chuỗi text.

- **Cú pháp: MID(text; m ; n)**

- **Ví dụ:** = MID("Dai hoc An Giang"; 9; 2) → "An"

④ **Hàm LEN:** Hàm cho giá trị số là số lượng ký tự của chuỗi text.

- **Cú pháp: LEN(text)**

- **Ví dụ:** = LEN("Dai hoc An Giang") → 16

⑤ **Hàm LOWER:** Đổi tất cả các ký tự trong chuỗi text thành ký tự thường

- **Cú pháp: LOWER(text)**

- **Ví dụ:** = LOWER("KY NIEM") → "ky niem"

⑥ **Hàm UPPER:** Đổi tất cả các ký tự thường trong chuỗi text thành ký tự hoa.

- **Cú pháp: UPPER(text)**

- **Ví dụ:** =UPPER("bo mon tin hoc") → "BO MON TIN HOC"

⑦ **Hàm PROPER:** Đổi các ký tự đầu của mỗi từ trong chuỗi text thành dạng in hoa.

- **Cú pháp: PROPER(text)**

- Ví dụ: =PROPER("Tin hOc") → "Tin Hoc"

⑧ **Hàm VALUE:** Chuyển một chuỗi có dạng số thành một trị số.

- **Cú pháp:** VALUE(text)

- Ví dụ: =VALUE("\$1.000") → 1.000

⑨ **Hàm TRIM:** Cắt bỏ các khoảng trống vô ích của text.

- **Cú pháp:** TRIM(text)

- Ví dụ: =TRIM(" Step by step") → "Step by step"

2.6.4. Các hàm thống kê.

(Các ví dụ trong phần này sử dụng số liệu ở bảng bên dưới)

	A	B	C
1	C1	10	2
2	C2	2	4
3	C3	ABC	6
4	C1		8
5	C2	21	10
6	C3	3	12
7	C1	CDE	14

① **Hàm MAX:** Trả về trị số lớn nhất trong danh sách (list)

- **Cú pháp:** MAX(List)

- Ví dụ: =MAX(B1: B7) → 21

② **Hàm MIN:** Trả về trị số nhỏ nhất trong danh sách (list)

- **Cú pháp:** MIN(List)

- Ví dụ: =MIN(B1: B7) → 2

③ **Hàm AVERAGE:** Tính trung bình cộng các trị số trong danh sách (List).

- **Cú pháp:** AVERAGE(List)

- Ví dụ: =AVERAGE(B1: B7) → 9

④ **Hàm SUM:** Tính tổng các trị số trong danh sách (List)

- **Cú pháp:** SUM(List)

- Ví dụ: =SUM(B1:B7; 4) → 40

⑤ **Hàm RANK**

- **Cú pháp:** RANK(X; vùng xếp hạng; cách xếp hạng)

- **Công dụng:** Trả về thứ hạng của trị số X trong vùng xếp hạng căn cứ vào cách xếp hạng. Nếu cách xếp hạng bằng 0 thì X sẽ được đếm theo thứ tự hạng giảm dần; nếu cách xếp hạng bằng 1 thì X sẽ được đếm theo thứ tự hạng tăng dần.

- Ví dụ:

=RANK(C1;C1:C7;1) → 1

=RANK(C1;C1:C7;0) → 7

⑥ Hàm COUNT

- **Cú pháp:** COUNT(List)

- **Công dụng:** Đếm số phần tử có giá trị số, ngày, giờ (kể cả phần tử là biểu thức hoặc hàm cho kết quả là một trị số, ngày, giờ) trong danh sách (List). Nếu đối mục là một vùng bảng tính (Cell, Range) thì hàm sẽ đếm số ô có giá trị là trị số, ngày, giờ trong vùng. Với các đối mục không phải là vùng bảng tính thì hàm sẽ đếm số lượng đối mục có giá trị số, ngày, giờ.

- **Ví dụ:**

=COUNT(B1:B7) → 4

=COUNT(B1:B7; 6; "A") → 5

⑦ Hàm COUNTA

- **Cú pháp:** COUNTA(List)

- **Công dụng:** Nếu đối mục là một vùng bảng tính (Cell, Range) thì hàm sẽ đếm số ô có chứa số liệu trong vùng (không đếm các ô trống). Với các đối mục không phải là vùng bảng tính thì hàm sẽ đếm số lượng đối mục.

- **Ví dụ:**

=COUNTA(B1:B7) → 6

=COUNTA(B1:B7; 6; "A"; ; " ") → 10

⑧ Hàm COUNTBLANK

- **Cú pháp:** COUNTBLANK(Vùng)

- **Công dụng:** Đếm số ô rỗng trong vùng bảng tính, kể cả ô chứa biểu thức hoặc hàm cho trị rỗng.

- **Ví dụ:** (Số liệu bảng 1)

=COUNTBLANK(B1:B7) → 1

=COUNTBLANK(C1:C7) → 0

=COUNTBLANK(B1:B7; 6;"A") → Báo lỗi

⑨ Hàm COUNTIF

- **Cú pháp:** COUNTIF(Vùng; "Điều kiện")

- **Công dụng:** Đếm số ô trong Vùng thỏa điều kiện quy định bởi Điều kiện (kể cả những ô chứa biểu thức hoặc hàm cho trị kết quả thỏa điều kiện).

Điều kiện là chuỗi ký tự được bao bởi cặp dấu ngoặc kép "...", trong cặp dấu nháy kép này một điều kiện nào đó được bắt đầu bởi 1 trong các toán tử : > < >= <= = <> (riêng nếu là điều kiện "bằng" thì có thể không cần bắt đầu bởi toán tử =).

- Ví dụ: (Số liệu bảng 1)

=COUNTIF(A1:A7;"C2") → 2

=COUNTIF(B1:B7; ">=5") → 2

⑩ Hàm SUMIF

- **Cú pháp:** SUMIF(Vùng dò điều kiện; "Điều kiện"; Vùng lấy tổng số)

- **Công dụng:** Tính tổng các ô có giá trị số thuộc Vùng lấy tổng số mà có ô tương ứng cùng hàng thuộc Vùng dò điều kiện thỏa được Điều kiện.

- **Điều kiện:** Tương tự như hàm COUNTIF()

- Ví dụ: (Số liệu bảng 1)

=SUMIF(A1:A7;"C2"; B1:B7) → 24

2.6.5. Các hàm ngày giờ

① **Hàm NOW:** Cho trị là ngày giờ hệ thống của máy.

- **Cú pháp:** NOW()

② **Hàm TODAY:** Cho trị là ngày hệ thống hiện hành.

- **Cú pháp:** TODAY()

③ **Hàm DAY:** Cho trị là ngày trong tháng của một biểu thức dạng ngày tháng.

- **Cú pháp:** DAY(serial_number)

- Ví dụ: =DAY("21/3/1979") → 21

④ **Hàm MONTH:** Cho trị là tháng trong biểu thức dạng ngày tháng.

- **Cú pháp:** MONTH(serial_number)

- Ví dụ: =MONTH("20/8/1977") → 8

⑤ **Hàm YEAR:** Cho trị là năm trong biểu thức dạng ngày tháng.

- **Cú pháp:** YEAR(serial_number)

- Ví dụ: =YEAR("30/4/1975") → 1975

⑥ **Hàm DATEVALUE:** Đổi chuỗi dạng ngày thành trị số (number).

- **Cú pháp:** DATEVALUE(Date_text)

- Ví dụ: =DATEVALUE("2/9/1945") → 16477

⑦ **Hàm DATE:** Đổi các giá trị của các đối số năm, tháng, ngày của hàm thành dữ liệu ngày tháng tương ứng.

- **Cú pháp:** DATE(năm,tháng,ngày)

- Ví dụ: DATE(2005,02,18) → 18/02/2005

2.6.6. Các hàm dò tìm và tham chiếu.

(Các ví dụ trong phần này sử dụng bảng số liệu sau)

	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>	<i>G</i>
<i>1</i>		<i>MUA</i>	<i>BÁN</i>		<i>ĐIỂM</i>	<i>LOAI</i>	
<i>2</i>	<i>GẠO</i>	<i>3000</i>	<i>3050</i>		<i>0</i>	<i>YEU</i>	
<i>3</i>	<i>NẾP</i>	<i>3500</i>	<i>3600</i>		<i>5</i>	<i>TB</i>	
<i>4</i>	<i>ĐẬU</i>	<i>2800</i>	<i>2840</i>		<i>7</i>	<i>KHA</i>	
<i>5</i>	<i>BẮP</i>	<i>2500</i>	<i>2540</i>		<i>9</i>	<i>GIOI</i>	

① Hàm VLOOKUP

- **Cú pháp:** VLOOKUP(X; Vùng dò tìm; n; Cách dò)

- **Công dụng:** Dò tìm giá trị X trong cột đầu tiên bên trái của Vùng dò tìm và tùy theo Cách dò.

+ Nếu cách dò bằng 0 hoặc FALSE thì cột đầu tiên bên trái của Vùng dò tìm không cần xếp theo thứ tự tăng dần từ trên xuống. Trường hợp này hàm chỉ dò tìm chính xác giá trị X. Nếu không thấy thì xem như không tìm được.

+ Nếu cách dò bằng 1 hoặc TRUE hoặc “không có đối số” thì cột đầu tiên bên trái của Vùng dò tìm phải được xếp theo thứ tự tăng dần từ trên xuống dưới. Trường hợp này nếu không tìm được chính xác giá trị X thì hàm sẽ dò tìm trị lớn nhất nhỏ hơn X trong cột đầu tiên bên trái của Vùng dò tìm.

- **Kết quả:**

+ Nếu dò tìm được thì hàm sẽ cho kết quả là giá trị của ô tương ứng cùng hàng với ô chứa giá trị tìm được nhưng ở cột thứ n trong Vùng dò tìm.

+ Nếu không dò tìm được thì hàm sẽ cho trị là lỗi #N/A.

☞ Lưu ý :

- X có thể là chuỗi, số, địa chỉ ô chứa giá trị là chuỗi, số hoặc biểu thức có kết quả là chuỗi, số.

- n là số thứ tự của cột trong Vùng dò tìm được tính từ trái sang phải. Cột đầu tiên bên trái là 1.

- Vùng dò tìm phải có số cột $\geq n$, nếu không hàm sẽ trả về lỗi là #REF!.

Ví dụ: (Số liệu bảng trên)

=VLOOKUP(“NẾP”; A2:C5; 3;0)	→	3600
=VLOOKUP(5;E2:F5;2; 1)	→	“TB”
=VLOOKUP(6;E2:F5;2; 1)	→	“TB”
=VLOOKUP(6;E2:F5;2; 0)	→	#N/A
=VLOOKUP(5;E2:F5;3; 1)	→	#REF!

② Hàm HLOOKUP

- Cú pháp: HLOOKUP(X; Vùng dò tìm; n; Cách dò)

- **Công dụng:** Dò tìm giá trị X trong hàng đầu tiên bên trên của Vùng dò tìm và tùy theo Cách dò.

+ Nếu cách dò bằng 0 hoặc FALSE thì hàng đầu tiên bên trên của Vùng dò tìm không cần xếp theo thứ tự tăng dần từ trái sang phải. Trường hợp này hàm chỉ dò tìm chính xác giá trị X. Nếu không thấy thì xem như không tìm được.

+ Nếu cách dò bằng 1 hoặc TRUE hoặc “không có đối số” thì hàng đầu tiên bên trên của Vùng dò tìm phải được xếp theo thứ tự tăng dần từ trái qua phải. Trường hợp này nếu không tìm được chính xác giá trị X thì hàm sẽ dò tìm trị lớn nhất nhỏ hơn X trong hàng đầu tiên bên trên của Vùng dò tìm.

- Kết quả:

+ Nếu dò tìm được thì hàm sẽ cho kết quả là giá trị của ô tương ứng cùng cột với ô chứa giá trị tìm được nhưng ở hàng thứ n trong Vùng dò tìm.

+ Nếu không dò tìm được thì hàm sẽ cho trị là lỗi #N/A.

🔍 Lưu ý :

- X có thể là chuỗi, địa chỉ ô chứa giá trị là chuỗi số hoặc biểu thức có kết quả là chuỗi, số.

- N là số thứ tự của hàng trong Vùng dò tìm được tính từ trên xuống dưới. Hàng đầu tiên bên trên là 1.

- Vùng dò tìm phải có số hàng $\geq n$, nếu không hàm sẽ trả về lỗi là #REF!.

③ Hàm INDEX**- Cú pháp: INDEX(Vùng lấy kết quả; X; Y)**

- **Công dụng:** Cho trị là giá trị của ô ở hàng thứ X và cột thứ Y của Vùng lấy kết quả.

+ Hàng X tính từ trên xuống dưới, hàng đầu tiên phía trên là hàng 1

+ Cột Y tính từ trái sang phải, cột đầu tiên bên trái là cột 1.

Ví dụ: (Số liệu bảng trên)

=INDEX(B2:C5; 3; 2) → 2840

=INDEX(B2:C5; 3; 3) → #REF!

=INDEX(B2:C5; 5; 2) → #REF!

④ Hàm MATCH**- Cú pháp: MATCH(X; Vùng dò tìm; Cách dò)**

- **Công dụng:** Dò tìm giá trị X trong Vùng dò tìm và tùy theo Cách dò

+ Nếu cách dò bằng 0 thì Vùng dò tìm không cần xếp theo thứ tự từ trên xuống dưới hoặc từ trái qua phải. Trường hợp này hàm chỉ dò tìm chính xác giá trị X. Nếu không thấy thì xem như không tìm được.

+ Nếu cách dò bằng 1 thì Vùng dò tìm phải được xếp theo thứ tự tăng dần từ trên xuống dưới hoặc từ trái qua phải. Trường hợp này nếu không tìm được chính xác giá trị X thì hàm sẽ dò tìm trị lớn nhất nhỏ hơn X trong Vùng dò tìm.

+ Nếu cách dò bằng -1 thì Vùng dò tìm phải được xếp theo thứ tự giảm dần từ trên xuống dưới hoặc từ trái qua phải. Trường hợp này nếu không tìm được chính xác giá trị X thì hàm sẽ dò tìm trị nhỏ nhất lớn hơn X trong Vùng dò tìm.

-Kết quả:

+ Nếu dò tìm được thì hàm sẽ cho kết quả là số thứ tự của ô chứa trị tìm được trong Vùng dò tìm.

+ Nếu không dò tìm được thì hàm sẽ cho trị là lỗi #N/A (Value not available).

Lưu ý :

- X có thể là chuỗi, số, tọa độ ô chứa giá trị là chuỗi, số hoặc biểu thức có kết quả là chuỗi, số.

- Nếu cách dò là 0 và X là chuỗi thì có thể dùng các ký tự đại diện.

Ví dụ: (Số liệu bảng 2)

=MATCH(7; E2:E5; 1)	→	3
=MATCH(8; E2:E5; 1)	→	3
=MATCH(8; E2:E5; 0)	→	#N/A

2.7. XỬ LÝ DỮ LIỆU DẠNG DANH SÁCH

2.7.1. Khái niệm.

Danh sách là một tập hợp các thông tin có liên quan với nhau được tổ chức theo một cấu trúc: Gồm một dòng tiêu đề ở phía trên, bên dưới là các dòng dữ liệu. Danh sách được tạo ra trên một phần của bảng tính, và được sử dụng để quản lý và bảo trì một khối lượng dữ liệu tương đối lớn và phức tạp.

Chẳng hạn nếu muốn lưu trữ tên, địa chỉ, số điện thoại,... của khách hàng, có thể tạo ra một danh sách để quản lý các thông tin sau cho mỗi khách hàng: Họ và tên, địa chỉ, số điện thoại, số fax,... Mỗi thông tin trên được đưa vào một cột dữ liệu riêng biệt. Tất cả các cột có liên quan đến một khách hàng tạo thành một dòng dữ liệu.

Bảng sau đây là một ví dụ về danh sách, với các tiêu đề nằm ở dòng thứ hai và các dòng tiếp theo là các dòng dữ liệu. Các cột trong danh sách có các kiểu dữ liệu khác nhau như trị số, ký tự, tiền tệ và có một số cột được tính ra từ các cột khác.

	A	B	C	D	E	F	G
1		BẢNG LƯƠNG NHÂN VIÊN THÁNG 01/2002					
2	STT	HỌ VÀ TÊN	HSL	NC	LƯƠNG	PHỤ CẤP	TỔNG CỘNG
3	1	Nguyễn Minh Hải	4.98	22	1,195,200	597,600	1,792,800
4	2	Lý Văn Hùng	3.56	23	893,236	446,618	1,339,855
5	3	Võ Duy Tâm	2.81	22	674,400	337,200	1,011,600
6	4	Châu Ngọc Giao	2.5	22	600,000	300,000	900,000
7	5	Trương Minh Cường	2.26	22.5	554,727	277,364	832,091
8	6	Võ Văn Cường	2.02	21	462,764	231,382	694,145
9	7	Nguyễn Chánh	1.78	23	446,618	223,309	669,927
10	8	Đương Thị Châu	1.31	24	342,982	171,491	514,473
11		TỔNG CỘNG			5,169,927	2,584,964	7,754,891

* **Lưu ý:**

- Không dùng các dòng trống để chia nhóm các dòng dữ liệu trong cùng một danh sách. Excel diễn dịch một dòng trống là dấu hiệu kết thúc danh sách.
- Các cột nên có cùng một kiểu dữ liệu ngoại trừ dòng tiêu đề.
- Các công thức nên được tính toán dựa trên các ô trên cùng một dòng. Nếu phải sử dụng một công thức dựa trên dữ liệu nằm bên ngoài danh sách, bạn hãy sử dụng tham chiếu tuyệt đối.
- Giữa dòng tiêu đề và các dòng dữ liệu không được có một dòng trống.
- Các ô ở dòng tiêu đề phải có dữ liệu loại chuỗi, cần phải ngắn gọn, không trùng lặp.

2.7.2. Sắp xếp dữ liệu trong danh sách (Sort).

Excel cho phép sắp xếp dữ liệu trong danh sách theo một cột nào đó trong một phạm vi đã chọn. Cần lưu ý khi chọn vùng dữ liệu sẽ sắp xếp vì nếu chọn không hết các cột trong danh sách có liên hệ với nhau thì có thể làm sai lệch nội dung trên các dòng.

Trình tự sắp xếp như sau:

Bước 1: Quét chọn vùng dữ liệu cần sắp xếp.

Bước 2: Chọn lệnh **Data\Sort** để mở hộp thoại **Sort**.



Hình 4.2

Bước 3: Xác định các thông số sắp xếp trong hộp thoại, bao gồm:

- **Sort by (Then by, Then by):** Chọn các cột làm khóa thứ nhất (thứ hai, thứ ba) để thực hiện sắp xếp. Dữ liệu sẽ được sắp xếp theo khóa thứ nhất. Chỉ khi nào có nhiều ô ở cột dùng làm khóa thứ nhất trùng nhau thì khóa thứ hai mới được xét đến,....

- ♦ **Ascending:** Sắp xếp theo thứ tự tăng dần.

- ♦ **Descending:** Sắp xếp theo thứ tự giảm dần.

- **Header row:** Lấy dòng đầu tiên trong vùng chọn làm tiêu đề. Nếu chọn mục này, Excel chỉ sắp xếp từ dòng thứ hai của danh sách trở đi.

- ♦ **No header row:** Sắp xếp tất cả các dòng trong vùng đã chọn.

Bước 4: Click *OK*.

2.7.3. Lọc dữ liệu trong danh sách (Autofilter).

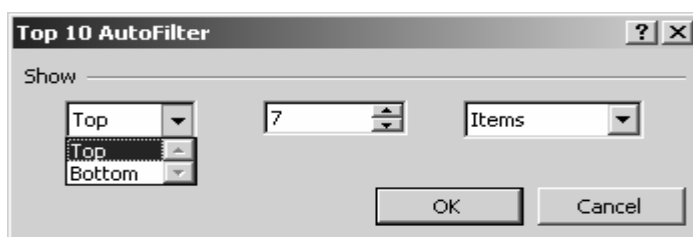
Lệnh **Data\Filter\AutoFilter** dùng để lọc các mẫu tin thỏa mãn một hoặc nhiều điều kiện được chỉ định. Khi lọc thì chỉ những mẫu tin thỏa mãn điều kiện chỉ định mới được hiển thị, các mẫu tin còn lại sẽ bị che.

* **Cách thực hiện:**

Bước 1: Chọn vùng dữ liệu để lọc hoặc chỉ cần di chuyển ô hiện hành vào vùng dữ liệu.

Bước 2: Chọn lệnh **Data\Filter\Autofilter**, Excel sẽ thêm các nút thả bên phải mỗi ô ở dòng tiêu đề. Click vào mỗi nút thả và chọn một trong các điều kiện lọc sau:

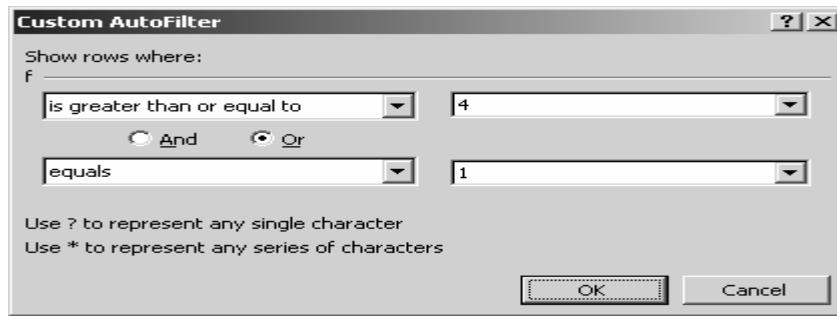
- **All:** Hiện thị tất cả các mẫu tin.
- **Top 10:** Mở hộp thoại *Top 10 Autofilter* để chọn 10 (hay một giá trị tùy ý) mẫu tin (hoặc số phần trăm mẫu tin) có giá trị cao nhất hay thấp nhất để hiển thị.



Hình 4. 3

* **Lưu ý:** tùy chọn *Top 10* không có tác dụng đối với các cột có dữ liệu là ký tự. Nếu cột đó có cả giá trị số và chữ, chỉ có các trị số trong cột được hiển thị.

- **Custom:** Mở hộp thoại *Custom Autofilter* để đặt các điều kiện lọc bằng cách dùng các toán tử so sánh. Bạn có thể chọn một toán tử so sánh ở hộp thả bên trái và chọn hoặc nhập vào giá trị so sánh ở hộp thả bên phải. Bạn cũng có thể dùng 2 toán tử so sánh cho mỗi cột (các toán tử này có thể thỏa mãn đồng thời (and) hoặc không cần phải thỏa mãn đồng thời (or). Ngoài ra, Excel còn cho phép dùng các ký tự đại diện ? và * đối với dữ liệu có kiểu ký tự.



Hình 4. 4

• **Các toán tử so sánh dùng để thiết lập điều kiện lọc.**

Toán tử	Ý nghĩa	Toán tử	Ý nghĩa
Equals	Bằng	Begins with	Bắt đầu bằng
Does not equal	Khác	Does not begin with	Không bắt đầu bằng
Is greater than	Lớn hơn	Ends with	Kết thúc bằng
Is greater than or equal to	Lớn hơn hoặc bằng	Does not end with	Không kết thúc bằng
Is less than	Nhỏ hơn	Contains	Bao gồm
Is less than or equal to	Nhỏ hơn hoặc bằng	Does not contain	Không bao gồm

- ♦ *Các trị xác định:* Hiện thị tất cả các giá trị *khác nhau* trong cột, nếu bạn chọn trị nào thì chỉ những mẫu tin có trị đó mới được hiển thị.
- ♦ *Blanks:* Hiện thị các mẫu tin trống ở cột đang xét.
- ♦ *Non blanks:* Hiện thị các mẫu tin có nội dung ở cột đang xét.

Lưu ý:

- Nếu đặt điều kiện ở các cột khác nhau thì chỉ những mẫu tin thỏa mãn tất cả các điều kiện đã đặt mới được hiển thị.
- Muốn hiện lại tất cả các mẫu tin, bạn thực hiện lệnh **Data\Filter>Show all** hoặc chọn *All* ở tất cả các cột.
- Nếu muốn gỡ bỏ các nút thả thì chọn lại lệnh **Data\Filter\Autofilter**.

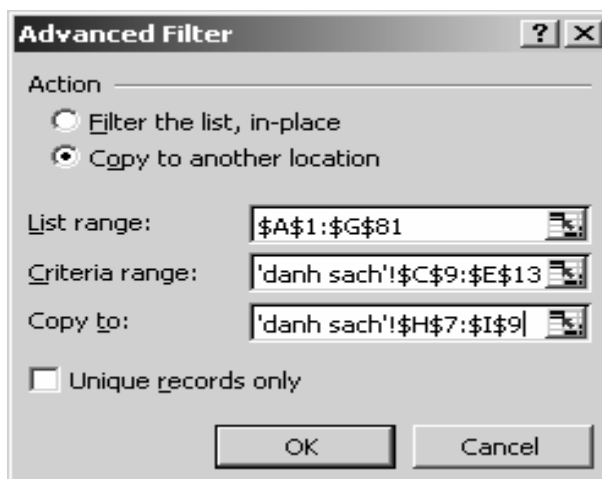
2.7.4. Lọc và rút trích các mẫu tin trong danh sách.

Khi điều kiện lọc phức tạp, hoặc bạn cần tính toán điều kiện thay vì chọn dữ liệu đang có,... bạn có thể dùng tính năng *lọc và rút trích (Advanced Filter)* của Excel.

*** Trình tự thực hiện.**

Bước 1: Chuẩn bị vùng điều kiện: Sao chép các ô trên dòng tiêu đề muốn đặt điều kiện lọc và dán vào một vị trí khác trên bảng tính, sau đó nhập các điều kiện lọc vào.

Bước 2: Quét chọn vùng dữ liệu cần trích, thực hiện lệnh **Data\Filter\Advanced Filter** để mở hộp thoại **Advanced Filter**.



Hình 4. 5

- Trong khung *Action*, chọn *Filter the list, in-place* để hiển thị kết quả lọc tại vị trí hiện tại cũ hoặc chọn *Copy to another location* để lọc và hiển thị kết quả ở một vị trí khác được xác định trong hộp *Copy to*.
- Xác định tham chiếu đến vùng chứa dữ kiện lọc trong khung *List range* và vùng chứa điều kiện lọc trong khung *Criteria range*.
- Xác định tham chiếu đến vùng chứa dữ kiện kết quả trong hộp *Copy to* (hộp này chỉ được kích hoạt khi bạn chọn *Copy to another location* trong khung *Action*).
- Chọn *Unique Records Only* để chỉ hiển thị một mẫu tin nếu có nhiều mẫu tin hoàn toàn giống nhau thỏa mãn điều kiện lọc.

Bước 3: Click *OK*.

* Lưu ý

- Các điều kiện ghi trên cùng một dòng là những điều kiện thỏa mãn đồng thời, các điều kiện ghi trên các dòng khác nhau là những điều kiện thỏa mãn không đồng thời.
- Khi điều kiện lọc có chứa các hàm như *Left, Right, Mid, Month, Day, Year,...* thì cần lưu ý:
 - Lấy tên trường ở vùng dữ liệu cần rút trích làm tham số của các hàm điều kiện lọc.
 - Thay đổi tên trường ở vùng điều kiện lọc cho khác với tên trường ban đầu.
 - Có thể sử dụng các ký tự đại diện * và ? trong các điều kiện có kiểu dữ liệu là chuỗi ký tự.
 - Bạn không thể chép dữ liệu đã được trích lọc đến sheet khác với *Advanced Filter*.
 - Nếu bạn chọn mục *Copy to another location*, tất cả dữ liệu (nếu có) ở phía dưới tham chiếu được chỉ ra trong hộp *Copy to* sẽ bị xóa.

* **Thí dụ:** Có một bảng tính lương sau:

BẢNG LƯƠNG NHÂN VIÊN								
STT	HỌ VÀ TÊN	CVỤ	LCB	NGÀY	PCCV	LƯƠNG	T.ÚNG	CLẠI
1	Lư Thanh Tuấn	GD	25.000	25	200.000	625.000	250.000	575.000
2	Võ Ngọc Hùng	PGD	22.000	28	150.000	682.000	250.000	582.000
3	Cao Ngọc Trang	PGD	22.000	26	150.000	594.000	250.000	494.000
4	Nguyễn Trí Dũng	TL	22.000	26	100.000	594.000	250.000	444.000
5	Huỳnh Công	KT	20.000	20	100.000	400.000	250.000	250.000
6	Lê Anh Tuấn	TQ	20.000	23	100.000	460.000	250.000	310.000
7	Võ Đông Điền	NV	15.000	28	50.000	465.000	250.000	265.000
8	Trần Đình Cảnh	NV	18.000	27	50.000	522.000	250.000	322.000
9	Lương Hùng Minh	NV	14.000	14	0	196.000	130.000	66.000
10	Trần Nhật Tân	NV	15.000	17	0	255.000	170.000	85.000

* Sau đây là một số cách thiết lập điều kiện lọc và rút trích.

① Có một yêu cầu duy nhất

- Yêu cầu: Lọc ra những người có họ bắt đầu bằng ký tự L.

+ Điều kiện:

HỌ VÀ TÊN	Hoặc	HỌ VÀ TÊN (*)
L*		=LEFT(B2;1)="L"

+ Kết quả

STT	HỌ VÀ TÊN	C.VỤ	LCB	NGÀY	PCCV	LƯƠNG	T.ÚNG	CLẠI
1	Lư Thanh Tuấn	GD	25.000	25	200.000	625.000	250.000	575.000
6	Lê Anh Tuấn	TQ	20.000	23	100.000	460.000	250.000	310.000
9	Lương Hùng	NV	14.000	14	0	196.000	130.000	66.000

- Yêu cầu: Lọc ra những người có LCB từ 20.000 trở xuống

+ Điều kiện:

LCB
<=20.000

+ Kết quả

STT	HỌ VÀ TÊN	C.VỤ	LCB	NGÀY	PCCV	LƯƠNG	T.ÚNG	CÒN
5	Huỳnh Công	KT	20.000	20	100.000	400.000	250.000	250.000
6	Lê Anh Tuấn	TQ	20.000	23	100.000	460.000	250.000	310.000
7	Võ Đông Điền	NV	15.000	28	50.000	465.000	250.000	265.000
8	Trần Đình Cảnh	NV	18.000	27	50.000	522.000	250.000	322.000
9	Lương Hùng	NV	14.000	14	0	196.000	130.000	66.000
10	Trần Nhật Tân	NV	15.000	17	0	255.000	170.000	85.000

② Có hai yêu cầu trở lên trong cùng một cột.

- Yêu cầu: Lọc ra những người có chức vụ là KT hoặc TQ.

+ Điều kiện lọc:

C.VỤ
KT
TQ

+ Kết quả

STT	HỌ VÀ TÊN	C.VỤ	LCB	NGÀY	PCCV	LƯƠNG	T.ÚNG	CÒN LẠI
5	Huỳnh Công	KT	20.000	20	100.000	400.000	250.000	250.000
6	Lê Anh Tuấn	TQ	20.000	23	100.000	460.000	250.000	310.000

③ Có yêu cầu ở hai cột trở lên

- Yêu cầu: Lọc ra những người có chức vụ là NV và lương lớn hơn 400.000.

+ Điều kiện lọc:

C.VỤ	LƯƠNG
NV	>400000

+ Kết quả

STT	HỌ VÀ TÊN	C.VỤ	LCB	NGÀY	PCCV	LƯƠNG	T.ÚNG	CÒN
7	Võ Đông Điền	NV	15.000	28	50.000	465.000	250.000	265.000
8	Trần Đình Cảnh	NV	18.000	27	50.000	522.000	250.000	322.000

- Yêu cầu: Lọc ra những người có chức vụ là NV hoặc lương nhỏ hơn 450.000.

+ Điều kiện lọc:

C.VỤ	LƯƠNG
NV	
	<450000

+ Kết quả

STT	HỌ VÀ TÊN	C.VỤ	LCB	NGÀY	PCC	LƯƠNG	T.ÚNG	CÒN
5	Huỳnh Công	KT	20.000	20	100.0	400.000	250.000	250.000
7	Võ Đông Điền	NV	15.000	28	50.00	465.000	250.000	265.000
8	Trần Đình Cảnh	NV	18.000	27	50.00	522.000	250.000	322.000
9	Lương Hùng	NV	14.000	14	0	196.000	130.000	66.000
10	Trần Nhật Tân	NV	15.000	17	0	255.000	170.000	85.000

2.7.5. Tổng hợp dữ liệu theo nhóm.

Trong các danh sách lớn gồm nhiều mẫu tin, bạn có thể dùng chức năng tổng hợp dữ liệu theo nhóm (*subtotal*) để tổng hợp số liệu thành các nhóm theo yêu cầu.

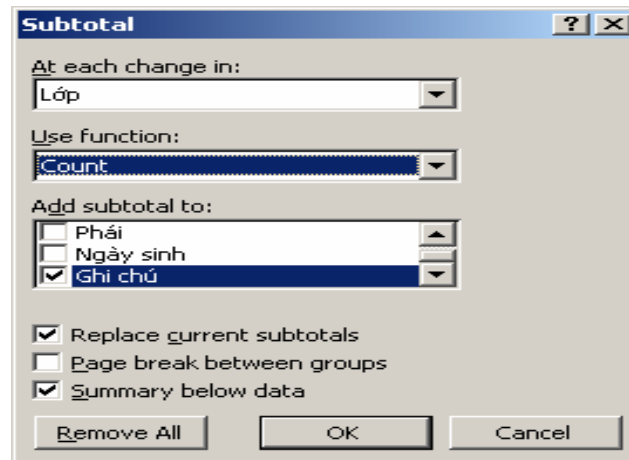
Ví dụ: Tổng lương của mỗi phòng ban, điểm trung bình của một nhóm học sinh,...

* Các bước thực hiện

Bước 1: Sắp xếp (*sort*) danh sách theo các cột muốn tổng hợp dữ liệu.

Bước 2: Click vào một ô trong danh sách.

Bước 3: Chọn lệnh *Data\Subtotals* để mở hộp thoại *Subtotal*.



Hình 4. 6

Bước 4: Trong hộp *At each change in*, click vào cột chứa các nhóm muốn tổng hợp. Cột này chính là cột đã sắp xếp ở bước 1.

Bước 5: Trong hộp *Use function*, chọn hàm sử dụng để thống kê hay tính toán. Bao gồm: *Sum, Count, Average, Max, Min, Product, ...*

Bước 6: Đánh dấu vào các hộp kiểm trong hộp *Add subtotal to* để chọn các cột chứa các giá trị muốn thống kê hay tổng hợp.

Bước 7: Click *OK* để thực hiện.

2.8. Vẽ biểu đồ trên bảng tính Excel

Trong một số lĩnh vực việc trình bày dữ liệu dưới dạng biểu đồ rất cần thiết. Biểu đồ giúp bạn biểu diễn sự tương quan của dữ liệu trên bảng tính trên phương diện đồ họa và so sánh các số liệu một cách trực quan hơn, điều này sẽ làm cho bảng tính trở nên sinh động và thuyết phục hơn.

2.8.1. Các loại biểu đồ

Có 3 loại biểu đồ thường gặp là biểu đồ dạng cột (*Columns*), dạng đường thẳng (*Line*) và dạng bánh (*Pie*). Từ 3 dạng này Excel triển khai thành 14 loại biểu đồ chuẩn (*Standard Type*) và 20 kiểu biểu đồ tùy chọn (*Customize types*) có thể dùng để biểu diễn số liệu trong bảng tính tùy theo nhu cầu thực tế của người dùng.

Các loại biểu đồ trong Excel và công dụng cơ bản của mỗi loại:

Loại biểu đồ	Chức năng
<i>Columns</i>	So sánh các loại dữ liệu với nhau theo chiều dọc
<i>Bar</i>	So sánh các loại dữ liệu với nhau theo chiều ngang.
<i>Line</i>	Cho xem sự thay đổi dữ liệu trong một giai đoạn.

<i>Pie</i>	So sánh tỷ lệ của các thành phần trong một tổng thể.
<i>XY (Scatter)</i>	Mô tả dữ liệu giữa hai loại dữ liệu liên quan.
<i>Area</i>	Nhấn mạnh tầm quan trọng tương đối của các giá trị qua một giai đoạn.
<i>Doughnut</i>	So sánh các phần với tổng thể trong một hoặc nhiều phạm trù dữ liệu (Biểu đồ <i>Pie</i> có một lỗ ở giữa).
<i>Rada</i>	Chỉ ra các thay đổi dữ liệu hoặc tần số dữ liệu tương đối với tâm điểm.
<i>Surface</i>	Tạo vết các thay đổi trong hai biến số với biến số thứ ba thay đổi, là một đồ họa 3 chiều.
<i>Buble</i>	Hiện sáng các chùm giá trị, tương tự như biểu đồ <i>Scatter</i>
<i>Stock</i>	Kết hợp giữa biểu đồ kiểu <i>Line</i> và <i>Column</i> . Được thiết kế đặc biệt để tạo vết giá cổ phiếu.
<i>Cylinder</i>	Sử dụng một hình trụ để trình bày các dữ liệu của biểu đồ kiểu <i>Bar</i> hay biểu đồ kiểu <i>Column</i>
<i>Cone</i>	Nhấn mạnh các đỉnh của dữ liệu.
<i>Pyramid</i>	Tương tự như biểu đồ <i>Cone</i>

2.8.2. Các bước dựng biểu đồ

1) Chuẩn bị dữ liệu cho biểu đồ.

Một biểu đồ được tạo từ dữ liệu của bảng tính hiện hành vì vậy khi xây dựng biểu đồ bạn cần tạo bảng tính có chứa các dữ liệu cần thiết.

Ví dụ như số liệu trong bảng chỉ tiêu xét tuyển nguyện vọng 1 của các ngành ngoài sư phạm của trường Đại học An Giang dưới đây.


CHỈ TIÊU XÉT TUYỂN NV1 TUYỂN SINH 2007 ĐHAG

(Các ngành ngoài sư phạm)

STT	Mã ngành	Tên ngành	Khối thi	Chỉ tiêu
1	103	Tin học	A,D1	100
2	302	Công nghệ Sinh học	B	90
3	202	Công nghệ Thực phẩm	B	60
4	310	Kỹ thuật Môi trường	A	50
5	401	Tài chính Doanh nghiệp	A,D1	50
6	402	Kế toán Doanh nghiệp	A,D1	100
7	405	Kinh tế đối ngoại	A,D1	100
8	407	Quản trị Kinh doanh	A,D1	100
9	408	Tài chính ngân hàng	A,D1	100
10	404	Phát triển Nông thôn	A,B	50
11	304	Nuôi trồng thủy sản	B	50
12	305	Chăn nuôi	B	50
13	306	Trồng trọt	B	50

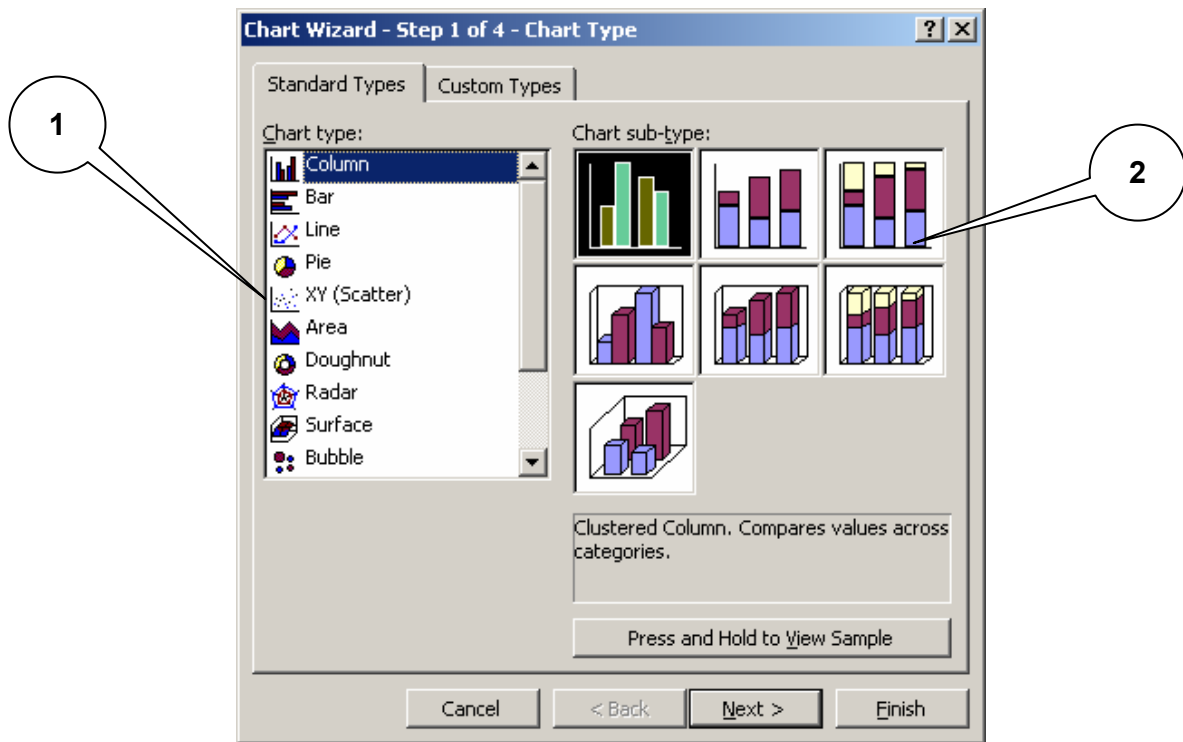
Yêu cầu: Dùng biểu đồ dạng cột để biểu diễn số lượng chỉ tiêu sinh viên của các ngành.

2) Các thao tác tạo biểu đồ.

Vào menu Insert/Chart hoặc Click vào biểu tượng  trên thanh Standard.

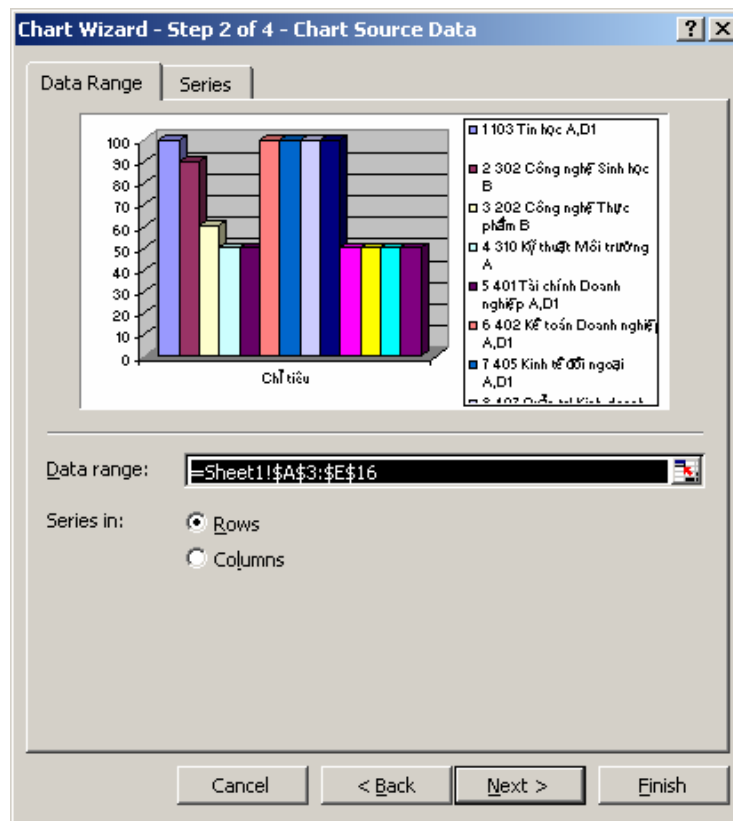
Bước 1: Chọn loại biểu đồ.

- 1: Chọn loại biểu đồ.
- 2: Chọn loại biểu đồ con.



Hình 4.9

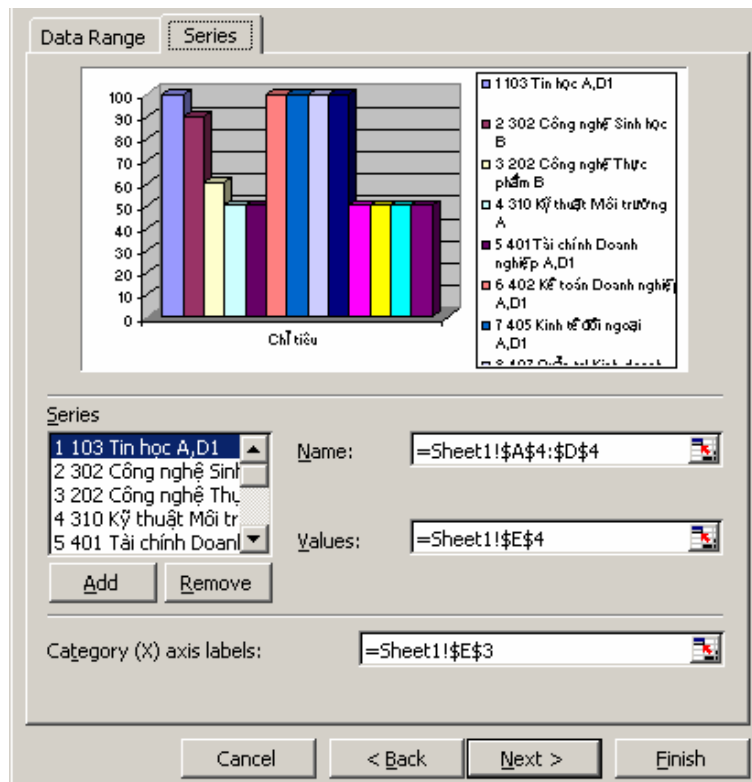
Bước 2: Nhấn Next, sau đó chọn vùng dữ liệu.



Hình 4.10

Lớp Data range (hình 4.10)

- Vùng Data range: Vùng dữ liệu của biểu đồ.
- Series in: Dữ liệu của từng đối tượng nằm theo hàng (row) hay theo cột (Column).



Hình 4.11. Thẻ Series

Lớp Series (Hình 4.11): Vùng dữ liệu từng thành phần trong biểu đồ

Series: Mỗi tên trong danh sách xác định một dãy số liệu trong biểu đồ. Dữ liệu tương ứng được định nghĩa trong hộp *Name* và *Value*.

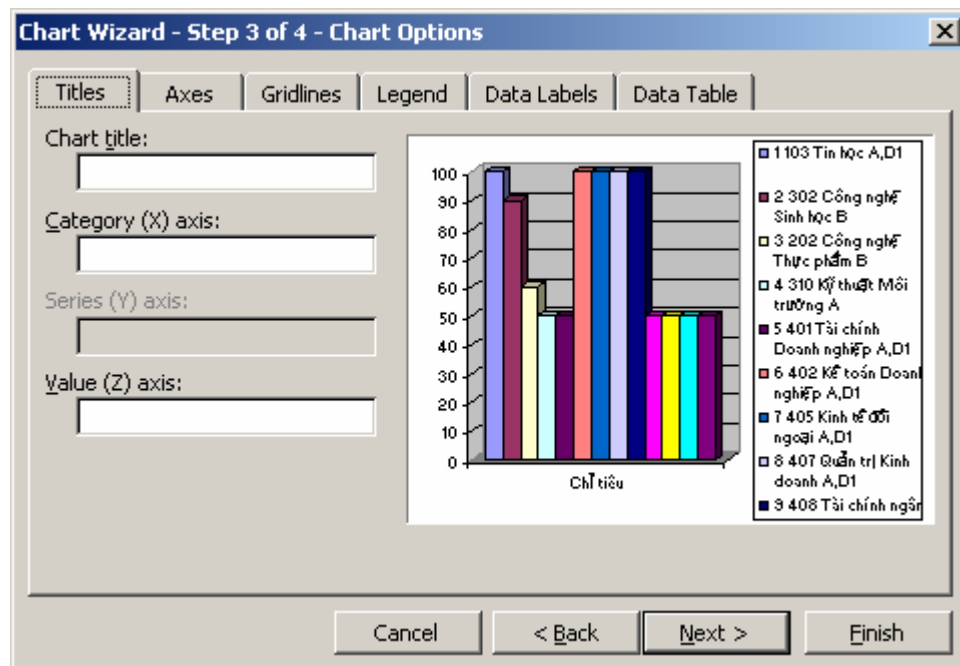
Name: tên cho dãy số liệu của đối tượng được chọn trong danh sách Series, là địa chỉ ô tiêu đề của dãy số liệu, nếu không có thì hộp *Name* sẽ trống khi đó bạn phải nhập vào. **Chú ý:** Những giá trị này sẽ là nhãn (*label*) trong phần chú giải xác định mỗi dãy số liệu trong bản đồ.

Value: địa chỉ của dãy số liệu được chọn trong danh sách *Series*.

Nút Add: Thêm dãy số liệu mới.

Nút Remove: Xóa dãy số liệu không cần đến.

Category (X) axis labels: Vùng dữ liệu làm tiêu đề trục X.

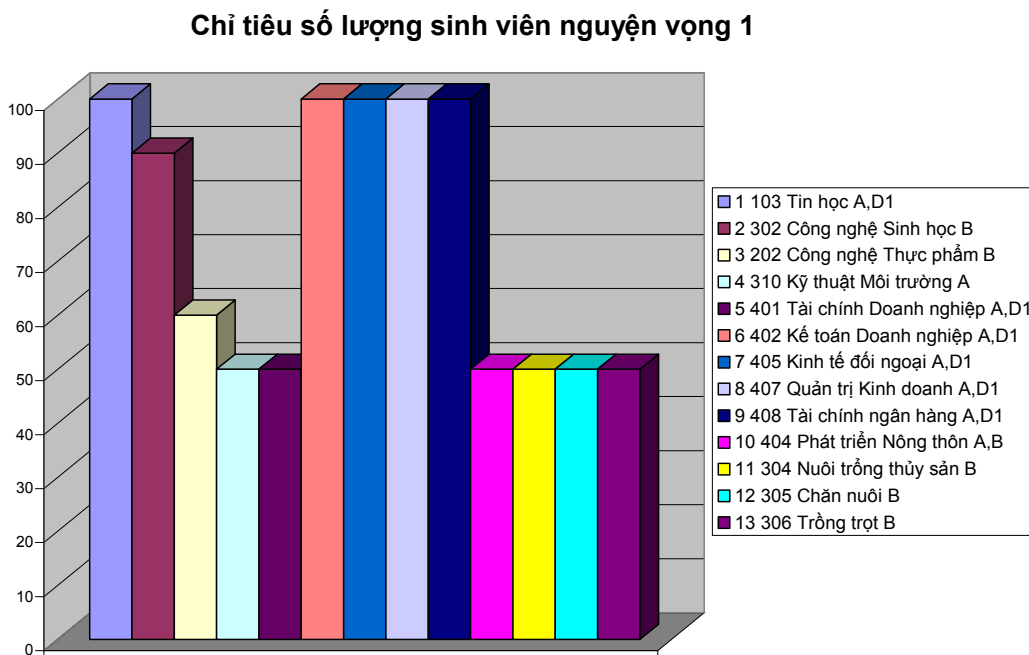
Bước 3: Xác định các tùy chọn cho biểu đồ*Hình 4.12. Xác định các tùy chọn cho biểu đồ*

- **Titles:** Đặt các tiêu đề cho biểu đồ.
- **Axes:** Tùy chọn cho các trục tọa độ.
- **Gridline:** Tùy chọn cho các đường lưới.
- **Legend:** Tùy chọn cho phần chú giải.
- **Data Label:** Thêm hoặc bỏ các nhãn dữ liệu cho các thành phần của biểu đồ.
- **Data Table:** Thêm bảng dữ liệu vào nguồn của biểu đồ.

Bước 4: Xác định vị trí đặt biểu đồ

- **As new sheet:** Tạo một sheet mới chỉ chứa biểu đồ.
- **As Object in:** Chọn sheet để đặt biểu đồ

Click vào nút Finish để hoàn thành.



Hình 4.13 . Kết quả của biểu đồ đã được thiết kế

3) Các thao tác chỉnh sửa biểu đồ.

Sau khi đã tạo biểu đồ theo 4 bước trên. Chúng ta có thể chỉnh sửa lại các thông số của biểu đồ bằng cách click chọn vào biểu đồ cần chỉnh sửa và thay đổi các thông số sau:

- Char Type: Chọn lại kiểu biểu đồ.
- Source Data: Chọn lại dữ liệu nguồn.
- Chart Options: Chọn lại các tùy chọn.
- Location: Chọn lại vị trí biểu đồ.
- Add Data: thêm dãy số liệu mới.

4) Định dạng biểu đồ

Tùy theo loại biểu đồ thành phần biểu đồ cần định dạng, excel sẽ cung cấp nội dung định dạng phù hợp.

Cách thực hiện:

Click chọn thành phần cần định dạng.

Vào menu Format/ Selected và sau đó chọn các định dạng theo ý muốn.

- *Patterns*: Cho phép định dạng đường viền và màu cho biểu đồ.
- *Font*: Cho phép định dạng các thành phần là văn bản trong biểu đồ.
- *Properties*: Cho phép thay đổi các thuộc tính của biểu đồ.


Chương 3.

TRANG TRÍ VÀ IN BẢNG TÍNH

3.1. THAY ĐỔI ĐỘ RỘNG CỘT, CHIỀU CAO DÒNG


3.1.1. Thay đổi độ rộng cột.

Bước 1: Chọn các cột muốn thay đổi độ rộng.

Bước 2: Trên thanh tiêu đề cột, chuyển trỏ chuột đến *đường biên phải* của cột cần thay đổi, khi trỏ chuột có dạng  thì rê chuột sang phải/trái để tăng/giảm độ rộng cột.

3.1.2. Thay đổi chiều cao dòng.

Bước 1: Chọn các dòng muốn thay đổi độ cao.

Bước 2: Trỏ chuột trên đường biên giữa các dòng của thanh chỉ số dòng, khi trỏ chuột có dạng  thì rê chuột theo lên/xuống để giảm/tăng chiều cao của dòng.

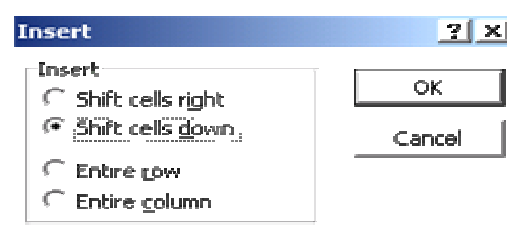
Lưu ý: Ngoài ra, có thể thay đổi độ rộng cột và chiều cao của dòng bằng cách dùng lệnh **Format\Columns\Width** hoặc **Format\Row\Height**.

3.2. CHÈN THÊM VÀ XÓA BỎ CÁC DÒNG, CỘT, Ô

3.2.1. Chèn thêm dòng , thêm cột, thêm ô

Bước 1: Chọn vùng muốn chèn thêm dòng, cột, ô.

Bước 2: Chọn menu **Insert/Cells**, chọn một trong các lựa chọn sau trong hộp thoại **Insert**:



Hình 4. 14

- *Shift Cells Right*: chèn vào các ô mới và đẩy dữ liệu tại vùng được chọn sang phải.
- *Shift Cells Down*: chèn vào các ô mới và đẩy dữ liệu tại vùng được chọn xuống dưới.
- *Entire Row*: chèn nguyên dòng.
- *Entire Column*: chèn nguyên cột.

3.2.2. Chèn nguyên dòng

Bước 1: Drag trên chỉ số dòng hoặc xác định vùng chứa số dòng cần chèn.

Bước 2: Chọn menu **Insert/Rows**. Các dòng đang chọn sẽ bị đẩy xuống dưới.

3.2.3. Chèn nguyên cột

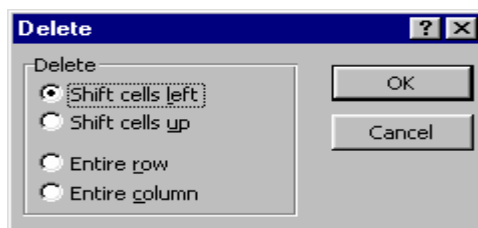
Bước 1: Drag trên tên cột hoặc xác định vùng chứa số cột cần chèn.

Bước 2: Chọn menu **Insert/Columns**. Các cột đang chọn sẽ bị đẩy sang phải.

3.3. Xoá dòng, xoá cột, xoá ô

Bước 1: Chọn vùng muốn xoá.

Bước 2: Chọn menu **Edit/Delete**.



Hình 4.15. Delete Cell

- *Shift Cells Left*: Xoá các ô đã chọn, các ô phía phải sẽ dồn sang trái.
- *Shift Cells Up*: Xoá các ô đã chọn, các ô phía dưới sẽ dồn lên trên.
- *Entire Row*: Xoá nguyên dòng.
- *Entire Column*: Xoá nguyên cột.

Bước 3: Click **OK**.

3.4. ĐỊNH DẠNG DỮ LIỆU

Dữ liệu khi nhập vào ô sẽ được hiển thị theo các định dạng mặc nhiên. Các yếu tố định dạng mặc nhiên này phụ thuộc vào:

1. Các qui định các định dạng về cách nhập số (number), ngày (date), giờ (time), ký hiệu tiền tệ (currency),...trong **Control Panel/Regional and Language options/customize**.
2. Phong chữ, cỡ chữ qui định trong menu **Tools/Option/General/Standard Font** và **Tools/Option/General/Size** của Excel.

Để thuận tiện, trong tài liệu này có một vài qui ước sau:

- + Ký hiệu dấu chấm thập phân (Decimal symbol): *dấu phẩy* (,)
- + Ký hiệu phân cách giữa các nhóm số (Digit grouping symbol): *dấu chấm* (.)
- + Ký hiệu phân cách các đối số trong hàm (List separator): *dấu chấm phẩy* (;)

3.4.1. Định dạng dữ liệu kiểu ngày (Date)

Bước 1: Chọn khối dữ liệu muốn định dạng kiểu ngày.

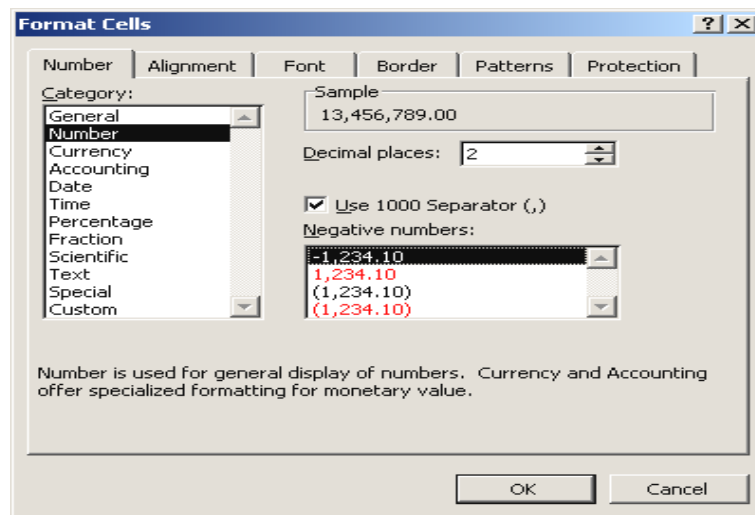
Bước 2: Chọn menu **Format/Cells**, chọn thẻ **Number**, chọn **Date** trong **Category**.

Bước 3: Chọn một kiểu ngày trong khung **Type**. Click **OK**.

3.4.2. Định dạng dữ liệu kiểu số (Number)

Bước 1: Chọn khối dữ liệu muốn định dạng kiểu số.

Bước 2: Chọn menu **Format/Cells**, chọn thẻ **Number**, chọn **Number** trong **Category**.



Hình 4.16. Format Number

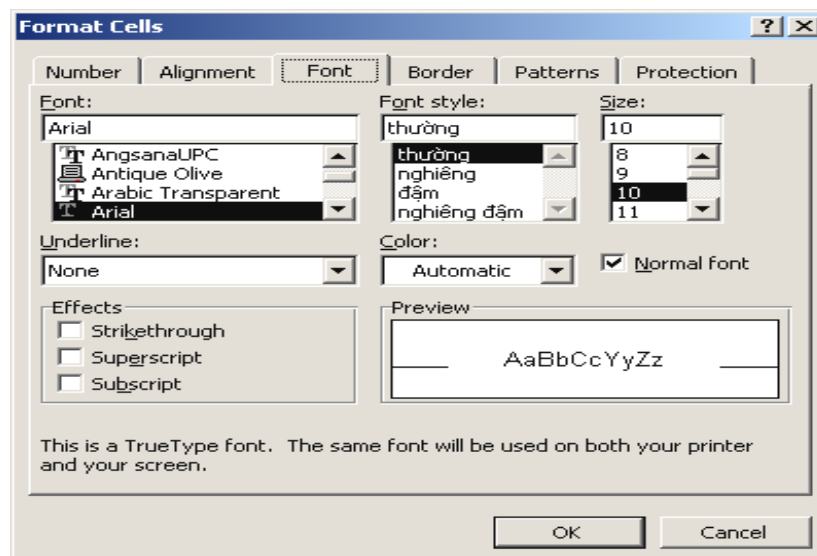
③ Chọn các định dạng: Số chữ số thập phân (**Decimal places**); sử dụng dấu phân cách hàng nghìn (**Use 1000 separator (,)**); dạng hiển thị số âm (**Negative numbers**).

Bước 3: Click **OK**.

3.4.3. Định dạng ký tự (Font).

Bước 1: Chọn khối dữ liệu muốn định dạng kiểu chữ.

Bước 2: Chọn menu **Format/Cells** sau đó chọn thẻ **Font**. Chọn các định dạng về font chữ tương tự như trong Word.



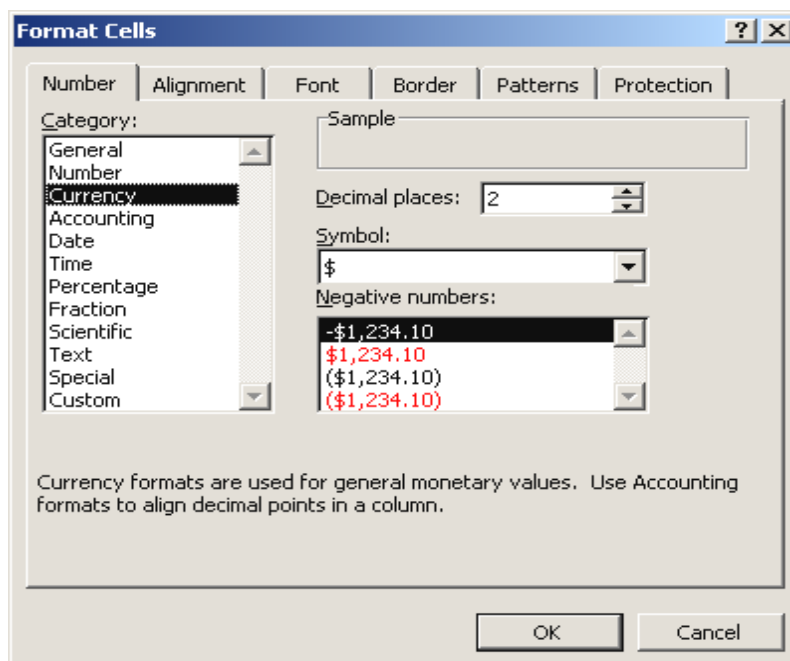
Hình 4.17. Format Cells/ Font

Bước 3: Click **OK**.

3.4.4. Định dạng tiền tệ (Currency).

Bước 1: Chọn khối dữ liệu muốn định dạng.

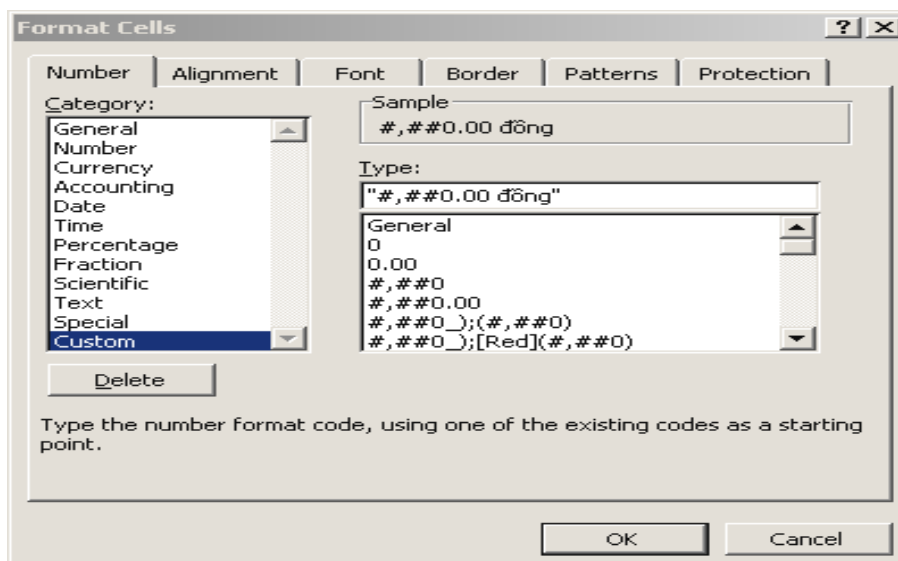
Bước 2: Chọn thẻ **Number**, chọn **Currency** trong **Category**.



Bước 3: Chọn số chữ số thập phân (**Decimal places**); ký hiệu tiền tệ (**Symbol**); dạng hiển thị số âm (**Negative numbers**).

Bước 4: Click **OK**.

3.4.5. Định dạng tùy biến (Custom) cho kiểu tiền tệ và ngày.



Hình 4. 18

Bước 1: Chọn khối dữ liệu muốn định dạng kiểu tiền tệ.

Bước 2: Chọn menu **Format/Cells**, chọn thẻ **Custom**.

Bước 3: Trong khung **Type**, gõ vào các mã định dạng **dd/mm/yyyy** để hiển thị kiểu ngày dạng ngày/tháng/năm hoặc **"#,##0.00 đồng"** để hiển thị dữ liệu kiểu tiền tệ ở dạng có chữ **đồng** phía sau số tiền (có 2 số lẻ thập phân).

Bước 4: Click **OK**.

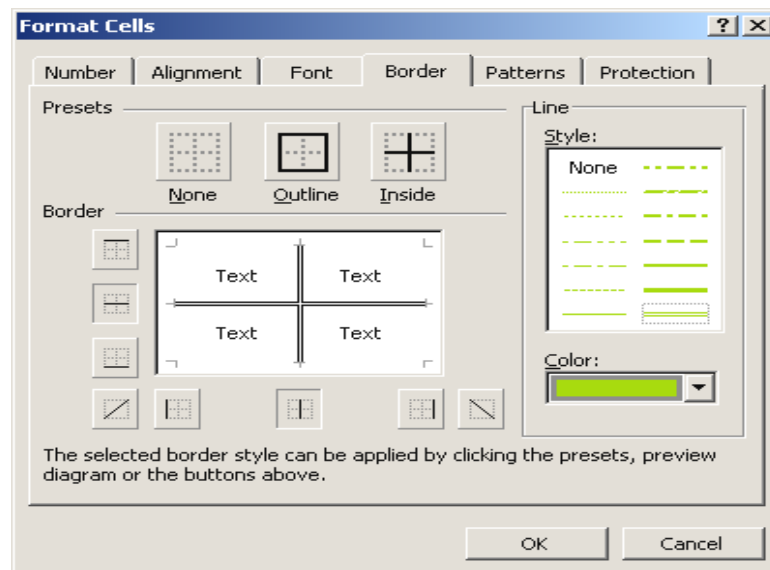
3.5. Kẻ khung giữa các ô trong vùng

Giữa các ô của Excel có các lưới phân cách, các lưới này có thể in ra máy in hay không là do người sử dụng chọn. Tuy nhiên nó không phải là các đường kẻ mong muốn mà chỉ đơn thuần là lưới phân cách giữa các ô. Các lưới này có thể bật/tắt bằng cách chọn **Tool /Options /View /Gridlines**.

Muốn có các đường kẻ theo mong muốn thì phải định dạng như sau:

Bước 1: Chọn các ô cần định dạng.

Bước 2: Chọn lệnh **Format\Cells**, chọn tab **Border**.



Hình 4. 19

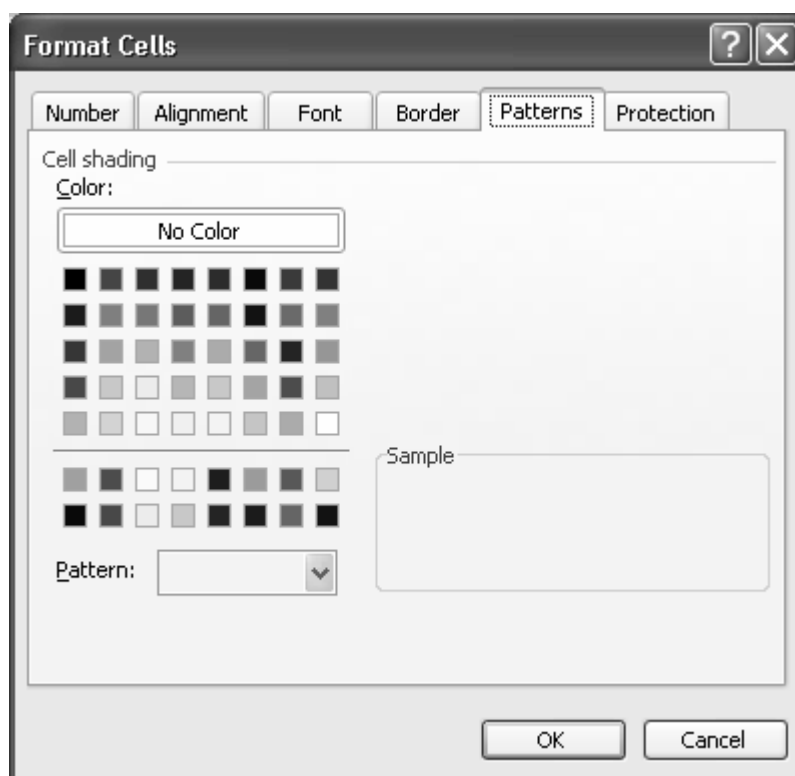
Bước 3: Chọn kiểu (style) và màu (Color) của đường kẻ trong khung **Line**.

Bước 4: Click vào các cạnh muốn kẻ trong khung **Presets** và **Border**. Click **OK**.

3.6. Tô nền trong ô

Bước 1: Chọn các ô cần định dạng.

Bước 2: Chọn menu **Format/Cells** chọn thẻ **Pattern**.



Hình 4. 20

Bước 3: Chọn màu (Color) và mẫu tô (Pattern).

Bước 4: Click **OK**.

3.7. IN ẤN TRONG EXCEL.

3.7.1. Định dạng trang in.

Bước 1: Chọn bảng tính cần định dạng.

Bước 2: Thực hiện lệnh **File\Page setup** để mở hộp thoại **Page Setup**.

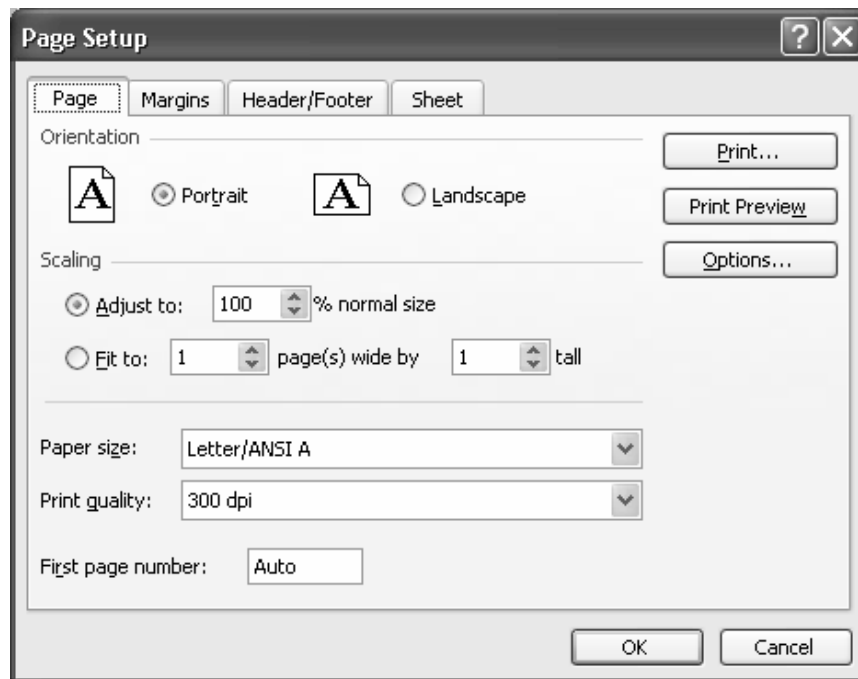
Bước 3: Chọn tab **Page**, sau đó chọn hướng in (Landscape hoặc Portrait), kích cỡ giấy in (Paper size), điều chỉnh tỷ lệ in so với kích thước thật (Adjust to...)

Bước 4: Chọn thẻ Margins rồi thiết lập các tùy chọn sau:

- Chọn lề trên (Top), dưới (Bottom), trái (Left) và phải (Right).
- Chừa phần tiêu đề đầu và chân trang (Header/Footer)
- Chọn cách canh dữ liệu trong bảng tính ở giữa trang giấy theo chiều ngang/dọc (*Horizontally/Vertical*).

Bước 5: Chọn thẻ **Header/Footer** để chèn các thành phần như số trang, đường dẫn đến tập tin,... vào tiêu đề ở đầu/cuối trang.

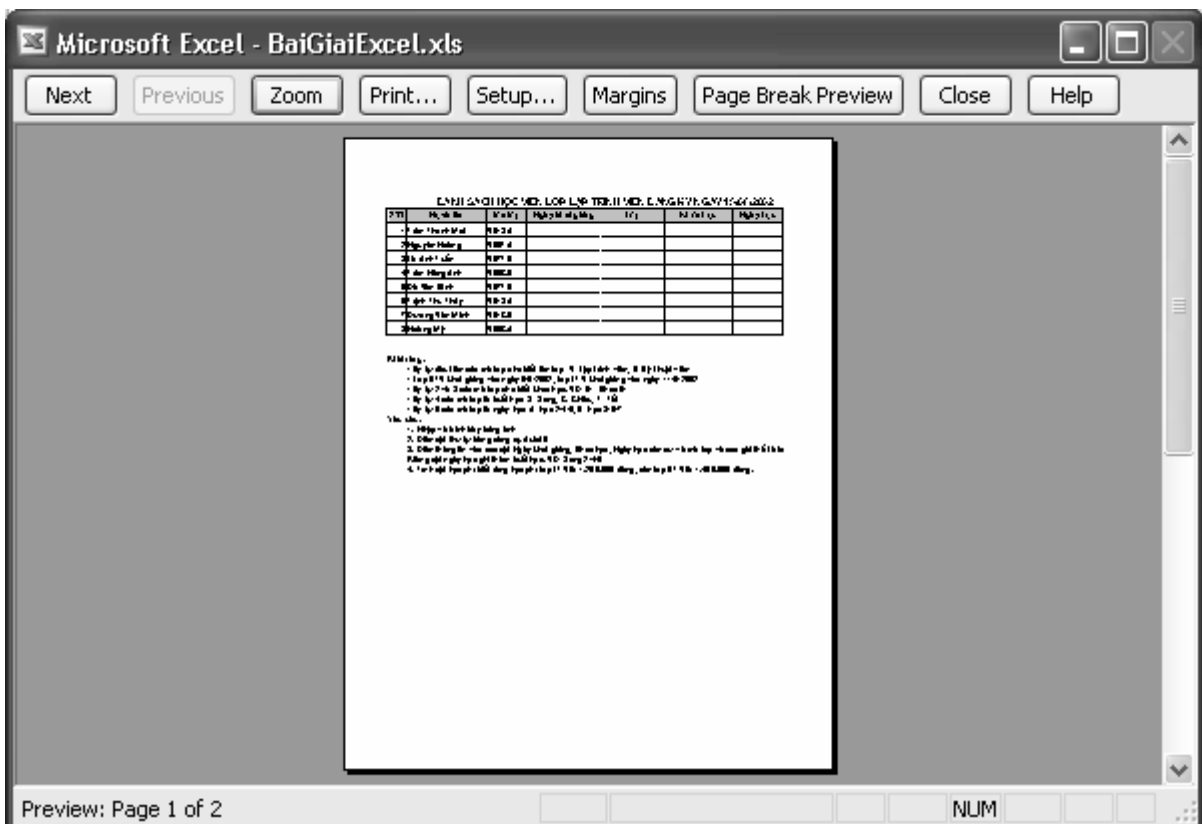
Bước 6: Click **OK**.



Hình 4.22. Hộp thoại Page Setup

3.7.2. Xem trước khi in

Trước khi in một worksheet, bạn nên chuyển sang chế độ xem trước khi in bằng cách click vào nút lệnh **Print Preview** trên thanh công cụ hoặc lệnh **File\Print preview** để chỉnh sửa các sai sót về định dạng nếu có.



Hình 4. 23 Hộp thoại Print Preview

Next: Chuyển đến trang kế tiếp (nếu có).

Previous: Chuyển về trang trước đó (nếu có).

Zoom: Hiển thị trang in ở dạng phóng to hay thu nhỏ.

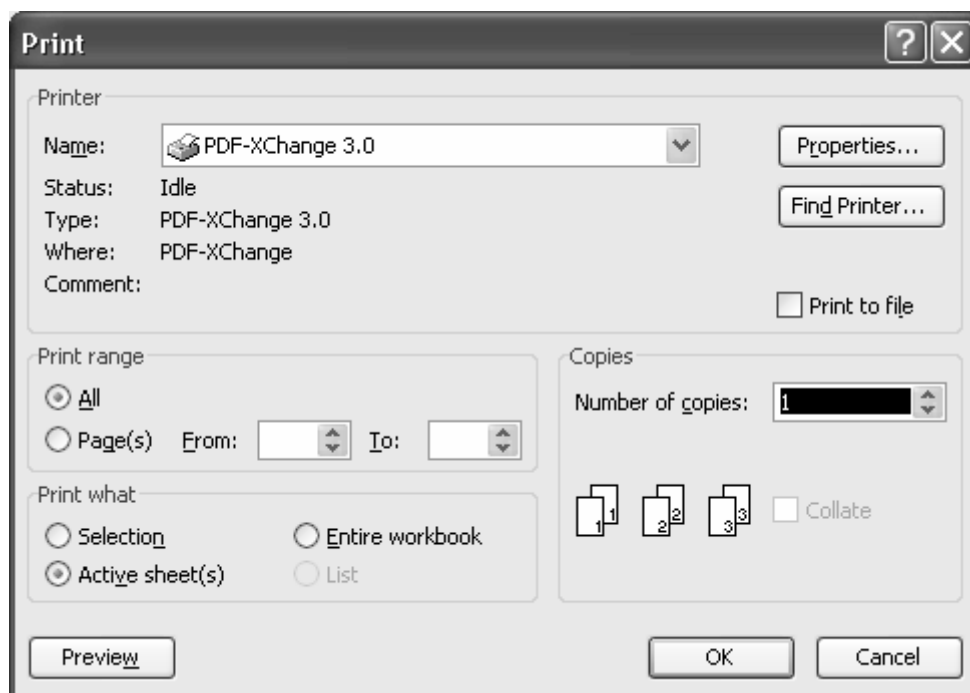
Print: Mở hộp thoại in.

Setup: Hiển thị hộp thoại định dạng trang in.

Margins: Cho phép thực hiện nhanh việc thay đổi lề trang, đầu và cuối trang, độ rộng cột bằng cách click và rê chuột.

3.7.3. Thực hiện in

Bước 1: Chọn lệnh *File\Print* để mở hộp thoại *Print*.



Hình 4. 24. Hộp thoại *Print*

Bước 2: Xác định các tham số trong hộp thoại:

- Phạm vi in (*Print range*): In tất cả các trang (*All*), hoặc chỉ in các trang được chọn (*Page(s) From ... To ...*).

- Xác định số bản in cho mỗi trang in (*Copies*).

Bước 3: Click **OK** để in.

BÀI TẬP

Bài thực hành 1: Thực hành các thao tác cơ bản.

1. Khởi động Excel theo các cách khác nhau.
2. Quan sát vị trí của các thanh tiêu đề, thanh thực đơn, các thanh công cụ, dòng nhập, các thanh trượt, sheet tab trên màn hình.
3. Nhập nội dung bất kỳ vào ô A1, sao đó lưu vào thư mục **My Documents** với tên tập tin là **Bai_TH.xls**.
4. Đóng workbook **Bai_TH.xls**.
5. Mở workbook **Bai_TH.xls**.
6. Chọn một worksheet khác trong cùng workbook.
7. Đổi tên worksheet mới chọn thành **Bang tinh hien hanh**.
8. Di chuyển worksheet hiện hành về vị trí đầu tiên bên trái.
9. Đổi tên worksheet hiện hành thành **Bang tinh dau tien**.
10. Chèn thêm worksheet mới vào sau worksheet hiện hành và đặt tên là **Bang tinh thu hai**.
11. Xóa tất cả các worksheet ở bên phải **Bang tinh thu hai**.
12. Mở workbook **Bai_TH.xls**.
13. Dùng chuột chọn lần lượt các ô **A300, AA300, IV1, IV65536, C65536, A1**.
14. Dùng các phím di chuyển trên bàn phím thực hiện yêu cầu ở câu trên.
15. Dùng chuột thực hiện thao tác chọn ô và vùng bất kỳ trên worksheet.
16. Dùng phím **SHIFT** kết hợp các phím di chuyển trên bàn phím thực hiện yêu cầu ở câu trên.
17. Dùng chuột lần lượt thực hiện các thao tác:
 - Chọn toàn bộ dòng 5.
 - Chọn toàn bộ cột C.
 - Chọn toàn bộ worksheet.
 - Chọn đồng thời dòng 5 và cột C.
 - Chọn đồng thời vùng **A1:C20** và **D25:H35**
18. Đóng workbook.

Bài thực hành 2. Nhập liệu tính toán các ô còn trống theo yêu cầu và lưu lại với tên BaiThucHanh02.xls

	A	B	C	D	E	F
1	BẢNG DỰ TRÙ VẬT TƯ					
2						
3	STT	Tên Vật Tư	ĐVT	Đơn giá	Số Lượng	Thành Tiền
4	1	Xi măng P400	Bao	45000	1845	
5	2	Cát	M3	30000	16	
6	3	Gạch thẻ	1000 Viên	400000	12	
7	4	Sắt 10	Kg	5200	345	
8	5	Gạch men	M2	52000	35	
9	6	Sắt 20	Kg	6200	20	
10	7	Gạch granit	M2	132000	12	
11	8	Gạch viên	Viên	2000	2520	
12	9	Bột trét tường	Kg	1000	150	
13	10	Sơn nước	Kg	12000	150	
14	11	Sơn dầu	Kg	25000	35	
15	12	Gạch trang trí	viên	2500	100	
16	Tổng cộng					

Yêu cầu:

*Thành tiền = Đơn giá * Số lượng.*

Tổng cộng = Tổng của cột thành tiền.

Bài thực hành 03: Thực hiện các yêu cầu sau rồi lưu tập tin vào đĩa với tên là BaiThucHanh03.xls

1. Nhập và trình bày bảng dự toán công trình sau:

	A	B	C	D	E	F	G	H	I
1	DỰ TOÁN CÔNG TRÌNH								
2			Mã số công trình:						
3			Tên công trình:						
4			Đơn vị thi công:						
5	STT	Hạng mục	Dài	Rộng	Cao	Giá	NC	Thành tiền	Ghi chú
6	1	Xây tường	4	4	3	100	2		
7	2	Xây mái	10	4	1	1500	3		
8	3	Xây thang	12	1	6	1600	2		
9	4	Quét vôi	20	4	4	400	2		
10	5	Đào móng	20	4	2	3000	4		
11	6	Làm nền	10	4	1	850	1		
12						Tổng cộng:			

2. Tính cột thành tiền theo công thức sau:

- $Thành\ tiền = Dài * Rộng * Cao * Giá + NC * 20000$

3. Tính tổng cộng của cột **Thành Tiền**.

Bài thực hành 04: Nhập liệu và tính toán các ô còn trống và lưu lại với tên *BaiThucHanh04.xls*

	A	B	C	D	E	F	G	H	I
1	LẬP CÔNG THỨC								
2	STT	A	B	A+B	A*B	A/B	A+B*2	(A+B)*2	A+B+A*B
3	1	12	4						
4	2	16	49						
5	3	3.2	26.29						
6	4	12.56	29.74						
7	5	27	156.5						
8	6	421	12						
9	7	45	97.3						
10	8	234.1	13.3						
11	9	16.4	12						
12	10	15	561						
13	11	24.5	19						
14	12	38.1	28.3						

Bài thực hành 05: Thực hiện các yêu cầu sau rồi lưu tập tin vào đĩa với tên là *BaiThucHanh05.xls*

	A	B	C	D	E	F	G
1	BẢNG LƯƠNG NHÂN VIÊN THÁNG 01/2002						
2	Họ và tên	HSL	HSPC	NC	Lương	Phụ cấp	Thực lĩnh
3	Nguyễn Minh Hải	4.98	1.8	22			
4	Lý Văn Hùng	3.58	1.5	23			
5	Võ Duy Tâm	2.81	1.4	22			
6	Châu Ngọc Giao	2.5	1.3	22			
7	Trương Minh Cường	2.28	1.2	22.5			
8	Tổng cộng						
9	Cao nhất						
10	Thấp nhất						
11	Bình quân						

- Chèn vào một cột trống trước cột **Họ và Tên** và đặt tiêu đề cho cột này là **STT**. Điền số thứ tự vào cột này. (Số thứ tự có dạng 01, 02, 03.....)
- Tính **Lương**, **Phụ cấp** và **Tổng cộng**, biết rằng:
 Lương cơ bản là **210000** và số ngày công chuẩn là **22**.
 $Lương = (HSL * Lương cơ bản * NC) / Số ngày công chuẩn$.
 $Phụ cấp = 1/2 Lương$.
 $Thực lĩnh = Lương + Phụ cấp$.
- Tính tổng của các cột **Lương**, **Phụ cấp** và **Thực lĩnh**.
- Thiết lập công thức cắt bỏ phần số lẻ của cột **Thực lĩnh** và làm tròn các số lẻ (nếu có) của cột **Lương** và cột **Phụ cấp**.

Bài thực hành 06: Thực hiện các yêu cầu sau rồi lưu tập tin vào đĩa với tên là *BaiThucHanh06.xls*

1. Tạo một tập tin mới, nhập và định dạng bảng tính sau: Nếu điểm nhập vào không hợp lệ (>0 và ≤ 10) thì excel sẽ tự động thông báo dữ liệu không hợp lệ (Sử dụng chức năng *Data/ Valication*)

	A	B	C	D	E	F	G	H	I	J	K
1	BẢNG XẾP HẠNG										
2	STT	Họ và tên	Toán	Lý	Hoá	Sinh	Văn	N.Ngữ	Điểm TB	Loại	Hạng
3	1	Trương Thị Mỹ	5	10	5	5	4	4			
4	2	Nguyễn Thị Thu	9	6	7	4	9	5			
5	3	Lý Thị Thu Hồng	5	8	7	5	9	4.5			
6	4	Lê Thị Ngọc	9	8.5	10	9.5	8	9.5			
7	5	Quách Anh Ngọc	4	7	6	4.5	9	7.5			
8	6	Trần Thị Thanh	5	9	6	3	3	9			
9	7	Bùi Thị Yến	8	8	9	5	10	10			
10	8	Nguyễn Thanh Thảo	9	7	5	8	10	8			
11	9	Nguyễn Phú Trung	3	5	10	9	5	6			
12	10	Phạm Thị Thanh	7	9	7	7	8	4			
13	11	Nguyễn Ngọc Ngạn	5	5	5	8	6	10			
14	12	Lưu Thị Vân	5	3	3	4	6	5			

2. Tính điểm trung bình của từng học sinh. Làm tròn điểm trung bình đến 1 chữ số thập phân. Ví dụ: 4.5 hoặc 5.7,....
3. Xếp loại cho từng học sinh vào cột **Loại** căn cứ vào các qui định sau:
 - Điểm TB $> 9,0$: Xuất sắc.
 - Điểm TB từ 8,0 – 9,0 : Giỏi.
 - Điểm TB từ 6,5 - cận 8,0 : Khá.
 - Điểm TB từ 5,0 – cận 6,5 : Trung bình.
 - Điểm TB dưới 5,0 : Yếu.
4. Xếp hạng cho từng học sinh (dùng hàm RANK).

Bài thực hành 07: Thực hiện các yêu cầu sau rồi lưu tập tin vào đĩa với tên là *BaiThucHanh07.xls*

1. Nhập vào nội dung bảng báo cáo nộp thuế
2. Tính số tiền giảm biết rằng nếu ngày nộp **trước ngày 31/12/2002** thì mỗi ngày nộp sớm sẽ được **giảm 0.5%** trên số tiền thuế.
3. Tương tự, cho biết mỗi ngày nộp trễ phải chịu phạt **0.4% trên số tiền thuế**, tính cột **Tiền phạt**.
4. Tính **Tiền nộp = Tiền thuế - Tiền giảm + Tiền phạt**.
5. Tính tổng cộng các cột **Tiền thuế, Tiền giảm, Tiền phạt, và Tiền nộp**.

	A	B	C	D	E	F	G
1	BÁO CÁO NỘP THUẾ THÁNG 12/2002						
2	STT	Khách hàng	Tiền thuế	Ngày nộp	Tiền giảm	Tiền phạt	Tiền nộp
3	001	Lương Văn Tịnh	25000000	24/12/2002			
4	002	Lê Lâm Huấn	37600000	24/12/2002			
5	003	Lê Văn Tuấn	15000000	07/01/2002			
6	004	Nguyễn Đoàn Xuân	18000000	16/11/2002			
7	005	Nguyễn Minh Ngọc	1500000	12/12/2002			
8	006	Lê Ngọc Nhân	32000000	28/12/2002			
9	007	Đình Nhật Vương	8600000	08/01/2003			
10	008	Nguyễn Thị Huyền Hải	12000000	07/01/2003			
11	009	Nguyễn Chí Bình	35000000	30/12/2002			
12	010	Nguyễn Thanh Nguyệt	48000000	26/12/2002			
13	011	Hà Minh Phong	45000000	01/01/2003			
14	012	Nguyễn Thanh Phương	108000000	31/12/2002			
15		Tổng cộng					

Bài thực hành 08: Thực hiện các yêu cầu sau rồi lưu tập tin vào đĩa với tên là BaiThucHanh08.xls

	A	B	C	D	E	F	G	H
1	DANH SÁCH HỌC VIÊN LỚP LẬP TRÌNH VIÊN ĐĂNG KÝ NGÀY 15/08/2002							
2	STT	Họ và tên	Mã lớp	Ngày khai giảng	Lớp	Khóa học	Ngày học	Học phí
3	1	Trần Thanh Mai	V61SA					
4	2	Nguyễn Hoàng	V55TA					
5	3	Lê Anh Tuấn	V57TB					
6	4	Trần Hồng Anh	V59CB					
7	5	Đỗ Văn Bình	V57TB					
8	6	Trịnh Thu Thủy	V61SA					
9	7	Dương Văn Minh	V61CB					
10	8	Hoàng Mỹ	V59CA					

Biết rằng:

- Ký tự đầu tiên của Mã lớp cho biết tên lớp: **V**: Lập trình viên, **K**: Kỹ thuật viên.
- Lớp Kỹ thuật viên khai giảng ngày 9/9/2002, lớp Lập trình viên khai giảng ngày 11/9/2002
- Ký tự 2 và 3 của Mã lớp cho biết khóa học. Ví dụ: 61: **Khoá 61**
- Ký tự 4 của Mã lớp là buổi học: **S**: Sáng; **C**: Chiều; **T**: Tối
- Ký tự 5 của Mã lớp là ngày học: A: học 2-4-6; B: học 3-5-7

Yêu cầu:

1. Nhập và trình bày bảng tính như trên.
2. Điền cột số thứ tự bằng công cụ **AutoFill**.

- Điền thông tin vào các cột **Ngày khai giảng**, **Khoá học**, và **Ngày học** căn cứ vào **Mã lớp** và các giả thiết trên. Riêng cột **Ngày học** ghi thêm buổi học, Ví dụ: **Sáng 2-4-6**.
- Tính cột học phí biết rằng học phí lớp **lập trình viên** là **1.200.000** đồng, còn lớp **kỹ thuật viên** là **1.400.000** đồng.

Bài thực hành 09: Thực hiện các yêu cầu sau rồi lưu tập tin vào đĩa với tên là *BaiThucHanh09.xls*

- Nhập và trình bày bảng tính sau.

	A	B	C	D	E	F	G	H	I
1	PHIẾU HÀNG HOÁ								
2	STT	Mã nhập	Tên hàng	Hãng sản xuất	Ngày nhập	Số lượng	Đơn giá	Thành tiền	Thuế
3	1	TVSS	?	?	14/07/1997	300	2000000	?	?
4	2	CSSN			02/05/1997	400	500000		
5	3	TVSN			18/12/1997	800	2200000		
6	4	TVSS			04/11/1996	300	2100000		
7	5	CSSN			12/10/1997	400	450000		
8	6	TVSS			15/10/1996	600	2250000		
9	7	CSSN			11/06/1997	1000	480000		
10						Tổng cộng			

- Biết rằng: Ký tự **1&2** của **Mã nhập** cho biết tên hàng: **CS: Cassettes**; **TV: Ti vi**. Hai ký tự cuối cho biết hãng sản xuất. **SN: Sony**; **SS: Samsung**. Thực hiện các yêu cầu sau:
 - Điền vào cột **Tên hàng** và **Hãng sản xuất** dựa vào giả thiết trên.
 - Tính **Thành tiền** = **Đơn giá** * **Số lượng**.
 - Tính cột **Thuế** = **Thành tiền** * **Thuế suất**, biết rằng mặt hàng **Tivi** có thuế suất **15%**, mặt hàng **Cassette** có thuế suất **12%**.
 - Tính **Tổng cộng** các cột **Thành tiền** và **Thuế**.

Bài thực hành 10: Thực hiện các yêu cầu sau rồi lưu tập tin vào đĩa với tên là *BaiThucHanh10.xls*

- Tạo một tập tin mới.
- Chọn **Sheet1**, đổi tên thành **Xuất_Nhập_Tồn**, nhập và định dạng bảng tính sau:

	A	B	C	D	E	F	G	H	I	J	K	L
1	BẢNG KÊ NHẬP XUẤT TỒN THÁNG 01/2001											
2	Mã hàng	Tên hàng	ĐVT	Đơn giá	Tồn đầu kỳ		Nhập trong kỳ		Xuất trong kỳ		Tồn cuối kỳ	
3					SL	T.Tiền	SL	T.Tiền	SL	T.Tiền	SL	T.Tiền
4	MHAN	Máy hàng gió đá	Cái	10000000	2		3		4			
5	MBOM	Máy bơm nước	Bộ	200000	3		8		9			
6	T5MM	Thép tấm dày 5 mm	Kg	4200	1000		2000		3000			
7	ĐHVN	Đồng hồ đo vạn năng	Chiếc	300000	2		10		11			
8	MTP2	Máy vi tính PII-450	Bộ	10000000	3		1		2			
9	MINL	Máy in Laser HP 1100	Cái	7000000	2		0		2			
10	HD10	Đĩa cứng 10 GB	Cái	1400000	8		5		13			
11	MH14	Màn hình 14" Proview	Cái	1540000	5		3		5			

3. Tính các ô còn trống trong bảng tính, biết rằng:

$$\text{Thành tiền} = \text{Số lượng} * \text{Đơn giá}$$

$$\text{SL Tồn cuối kỳ} = \text{SL Tồn đầu kỳ} + \text{SL Nhập trong kỳ} - \text{SL Xuất trong kỳ}$$

4. Định dạng các ô số trong bảng tính có dấu phân cách hàng nghìn.

5. Lưu và đóng tập tin.

Bài thực hành 11: Thực hiện các yêu cầu sau rồi lưu tập tin vào đĩa với tên là BaiThucHanh11.xls

	A	B	C	D	E	F	G	H	I	J	K
1	BẢNG KÊ PHÂN PHỐI NHU YẾU PHẨM										
2	Phiếu Xuất Kho	Ngày Xuất	Số Lượng	Loại Hàng						T.Tiền	Ghi Chú
3				Gạo		Đường		Sữa			
4	SL	T.Tiền	SL	T.Tiền	SL	T.Tiền					
5	C-001/G	10/1/02	80								
6	C-002/S	10/4/02	40								
7	K-003/D	10/7/02	60								
8	K-004/G	10/10/02	70								
9	K-005/G	10/12/02	80								
10	K-006/S	10/16/02	40								
11	C-007/D	10/18/02	60								
12	C-008/G	10/20/02	70								
13											
14	Bảng 1			Bảng 2							
15	Loại Hàng	Đơn giá		Loại Hàng	Cung cấp		Kinh Doanh				
16		C.Cấp	K.Doanh		SL	T.Tiền	SL	T.Tiền			
17	Gạo	50	80	Gạo							
18	Đường	80	120	Đường							
19	Sữa	90	120	Sữa							

1. Thiết lập công thức xác định số lượng gạo, sữa và đường vào các cột tương ứng căn cứ vào ký tự cuối của Phiếu xuất kho: **G**: gạo, **S**: sữa, và **D**: đường. Lưu ý thêm là mỗi phiếu chỉ xuất 1 loại hàng. **Ví dụ:** ở dòng đầu tiên trong bảng phân phối trên: số lượng gạo là 80, còn số lượng đường và sữa là 0.

2. Lập công thức tính cột **T.Tiền** của từng loại hàng, biết rằng:

$$\text{Thành tiền} = \text{SL} * \text{Đơn giá.}$$

Mỗi loại hàng có 2 loại đơn giá là **Cung cấp** và **Kinh doanh** và được cho trong bảng 1.

Nếu ký tự đầu của phiếu xuất kho là **C** thì phiếu hàng đó được cung cấp, ngược lại phiếu hàng đó được bán (kinh doanh).

3. Tính các ô còn trống trong bảng 2.

Bài thực hành 12. Trình bày bảng tính và thực hiện các yêu cầu dưới đây sau đó lưu lại với tên BaiThucHanh12.xls

Giải thích:

CSC: Chỉ số cũ; CSM: Chỉ số mới.

Yêu cầu

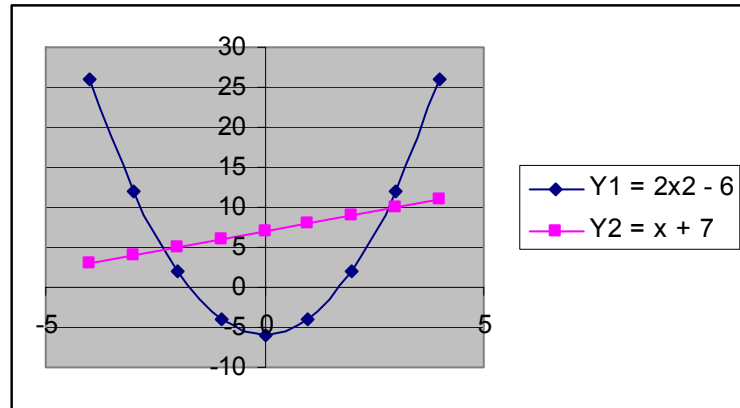
1. Điền số thứ tự vào cột số thứ tự theo dạng **01, 02, 03....**
2. Thêm các cột **KW, Vượt, Tiền điện, Tiền phạt, VAT** trước cột tổng cộng
3. **KW:** Mức tiêu thụ trong tháng tính bằng $KW = \text{Chỉ số mới} - \text{chỉ số cũ}$.
4. **Vượt:** Số KW vượt định mức. **Số KW định mức cho mỗi hộ là 120 KW**
5. **Tiền điện = Số KW Định mức * Giá tiền điện**, biết giá tiền định mức của khu vực 1 là 800đ/ 1KW. Các khu vực còn lại là 1000đ/ 1KW.

	A	B	C	D	E	F	G
1	TÍNH TIỀN ĐIỆN						
2	STT	HỌ VÀ TÊN CHỦ HỘ	CSC	CSM	Khu Vực	TỔNG	
3		Trần Thị Thanh Bình	115	240	2		
4		Phạm Trường Lâm	120	350	1		
5		Phan Cẩm Loan	161	421	3		
6		Quách Thế Long	145	165	2		
7		Trần Thị Minh Hằng	225	320	2		
8		Nguyễn Thành Tâm	142	178	1		
9		Trần Văn Ngọc	125	156	1		
10		Huỳnh Thanh Hoa	456	499	2		
11		Lê Hoàng Tùng	431	488	3		
12		Nguyễn Ngọc Thanh	589	691	3		
13		Trần Thanh Hồng	1754	1978	1		
14		Lê Văn Hùng	346	456	1		
15		Nguyễn Hữu Chiến	1245	1564	2		
16		Trần Văn Thông	132	342	2		
17		Lê Hữu Hoàng	357	451	1		
18		Hồ Văn Lợi	751	851	1		
19							
20	Mức tiêu thụ lớn nhất trong tháng						
21	Mức tiêu thụ trung bình trong tháng						

6. **Phạt:** Chỉ phạt những hộ có mức sử dụng vượt quá định mức. Mỗi KW vượt sẽ được tính với **giá 1500đ/ 1KW** đối với **tất cả các khu vực**.
7. **VAT = (Tiền điện + Tiền phạt) x 10%**
8. **Tổng cộng = Tiền điện + Tiền phạt + Thuế VAT.**
9. Tìm mức tiêu thụ **lớn nhất** trong tháng
10. Tìm mức tiêu thụ **trung bình** trong một tháng.
11. Sắp xếp bảng tính theo thứ tự **tăng dần** của **cột tổng cộng**.

Bài thực hành 13: Sử dụng biểu đồ XY(Scatter) trong Excel để vẽ đồ thị của 2 hàm số trên cùng một hệ trục tọa độ X,Y

- $Y1 = 2x^2 - 6$
- $Y2 = x + 7$



Bài thực hành 14: Thực hiện các yêu cầu sau rồi lưu tập tin vào đĩa với tên BaiThucHanh14.xls

	A	B	C	D	E	F	G	H
1	Bảng Kế Khai Bán Hàng Quý I							
2	STT	Mã hàng	Tháng	Tên hàng	Đơn giá	S-lượng	D-thu	Lãi
3	1	G01	1			20		
4	2	B02	2			30		
5	3	D01	3			11		
6	4	M01	2			200		
7	5	S02	2			50		
8	6	B01	1			200		
9	7	G03	2			150		
10	8	D03	3			90		
11	9	D02	2			120		
12	10	S02	1			300		
13								
14	Bảng phụ							
15	Mã Loại	Tên hàng	Đơn giá					
16	S	Sữa	5000					
17	M	Muối	3000					
18	B	Bánh	5000					
19	D	Đường	10000					
20	G	Gạo	4000					
21								
22	Bảng lãi suất							
23	Mã Loại	B	D	G	M	S		
24	Lãi suất	7%	10%	5%	4%	5%		

Chú thích: Ký tự đầu tiên của mã hàng là mã loại hàng.

1. Điền vào cột tên hàng từ bảng phụ.
2. Tính cột đơn giá từ bảng phụ biết nếu số lượng lớn hơn 100 thì giảm 10% đơn giá.

3. Tính doanh thu = đơn giá * số lượng
4. Tính tiền lãi = doanh thu * lãi suất (lấy từ bảng lãi suất)

Bài thực hành 15: Thực hiện yêu cầu dưới đây rồi lưu tập tin với tên BaiThucHanh15.xls

	A	B	C	D	E	F	G	H	I	J
1	BÁO CÁO BÁN HÀNG TỪ NGÀY 5/2 ĐẾN 12/2/2001									
2	STT	Mã hàng	Tên Hàng	Ngày	ĐVT	SL	ĐG bán	D.Thu	C.Phí	T.Lãi
3	1	ON		11/2/01		1200				
4	2	D4		6/2/01		2				
5	3	CA		12/2/01		5				
6	4	TH		7/2/01		500				
7	5	D1		8/2/01		3				
8	6	CA		10/2/01		4				
9	7	TH		9/2/01		200				
10	8	CE		11/2/01		15				
11	9	CE		6/2/01		2				
12	10	ON		11/2/01		800				
13	11	TA		12/2/01		300				
14	12	CA		5/2/01		10				
15	13	ON		8/2/01		500				
16	14	TH		12/2/01		300				
17	TỔNG CỘNG									
18	Bảng Phụ					Bảng Thống Kê				
19	Mã Hàng	Tên hàng	ĐVT	ĐG vốn	Phí B/V	Mặt hàng	SL	D.Thu	T.Lãi	
20	CA	Cát	m3	45000	5000	Cát				
21	CE	Xi măng	bao	45000	300	Xi măng				
22	D1	Đá 1/2	m3	160000	5000	Đá 1/2				
23	D4	Đá 4/5	m3	120000	5000	Đá 4/5				
24	ON	Gạch ống	viên	300	10	Gạch ống				
25	TA	Gạch tàu	viên	1800	30	Gạch tàu				
26	TH	Gạch thẻ	viên	250	8	Gạch thẻ				

1. Mở một tập tin mới, nhập và định dạng bảng tính dưới đây, sau đó sắp xếp lại thứ tự các mặt hàng trong Báo cáo bán hàng theo ngày tăng dần.

2. Điền các cột **tên hàng**, và **đơn vị tính** nhờ vào bảng phụ.

3. Tính cột **đơn giá bán**, **doanh thu**, **chi phí** và **tiền lãi**, biết rằng:

$$\text{Đơn giá bán} = \text{đơn giá vốn} * 1,2.$$

$$\text{Doanh thu} = \text{Số lượng} * \text{Đơn giá bán}.$$

$$\text{Chi phí} = \text{Phí B/V} + \text{thuế (5\% doanh thu)} + \text{phí khác (2\% doanh thu)}.$$

$$\text{Tiền lãi} = \text{Doanh thu} - \text{Giá vốn} - \text{Chi phí}.$$

$$\text{Giá vốn} = \text{Số lượng} * \text{Đơn giá vốn}.$$

Đơn giá vốn và phí bốc vác được cho trong bảng phụ.

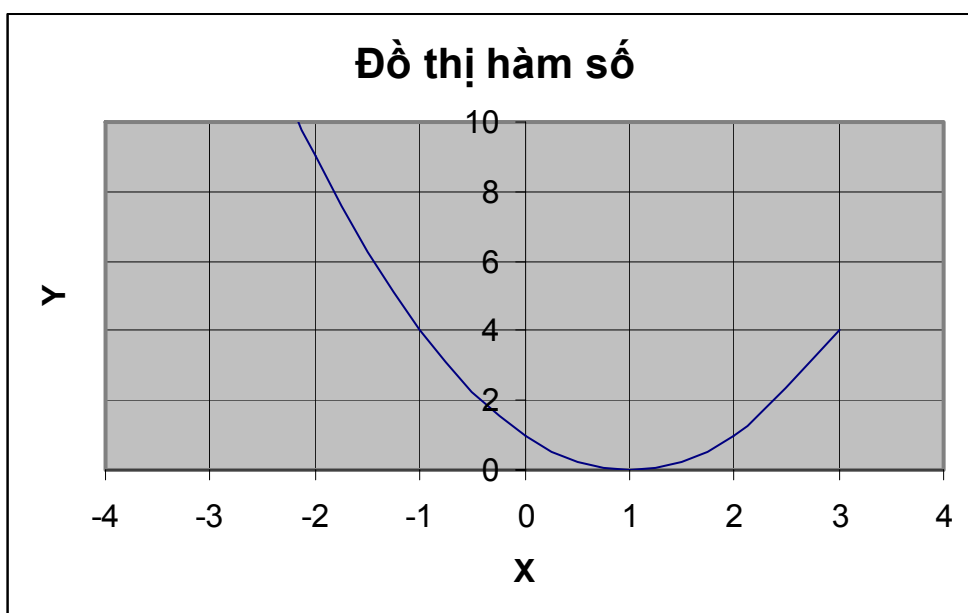
4. Tính tổng **doanh thu**, **chi phí** và **tiền lãi**.

5. Thống kê **số lượng**, **doanh thu** và **tiền lãi** vào bảng thống kê.

6. Thiết lập các điều kiện để lọc và rút trích ra từng mặt hàng.

7. Tổng hợp (**subtotal**) **doanh thu** và **tiền lãi** theo ngày.

Bài thực hành 16: Sử dụng biểu đồ XY(Scatter) trong Excel để vẽ đồ thị của hàm số $y = x^2 - 2x + 1$ trên cùng một hệ trục tọa độ X,Y



Bài thực hành 17: Thực hiện yêu cầu dưới đây

1. Mở một tập tin mới, nhập và trình bày bảng tính sau.
2. Điền các cột *Tên hàng* và *Đơn vị tính* dựa vào bảng phụ.
3. Tính cột đơn giá, biết rằng: Nếu bán sau ngày 25 thì đơn giá được giảm đi 2%.
4. Tính cột *Thành tiền* = *Số lượng* * *Đơn giá*.
5. Thống kê số lượng từng mặt hàng vào bảng tổng kết tháng, trong đó nhóm các loại máy tính, đĩa cứng và đĩa mềm lại với nhau.
6. Rút trích ra một bảng mới chỉ bao gồm các cột *tên hàng*, *đơn vị tính*, *số lượng* và *thành tiền*.
7. Sắp xếp các mẫu tin theo cột tên hàng. Tổng hợp (Subtotal) các cột *số lượng* và *thành tiền* theo cột *tên hàng*.
8. Định dạng lại cách hiển thị ở cột *ngày bán* theo dạng **ngày-tháng-năm**; canh giữa các ô dữ liệu kiểu ký tự, ngày, số thứ tự; canh phải các ô có trị số; định dạng các ô số có dấu phân cách hàng nghìn.
9. Vẽ biểu đồ thống kê số tiền theo mã hàng.

	A	B	C	D	E	F	G	H
1	CỬA HÀNG MÁY TÍNH							
2	Tháng 04/1996							
3	STT	Mã hàng	Tên hàng	ĐVT	Số lượng	Ngày bán	Đơn giá	Thành tiền
4	1	PC586			14	02-04-96		
5	2	PC486			10	12-04-96		
6	3	HD850			3	09-04-96		
7	4	FD120			10	11-04-96		
8	5	HD420			4	22-04-96		
9	5	FD144			10	06-04-96		
10	6	PC586			4	27-04-96		
11	7	HD850			4	06-04-96		
12	8	FD120			12	09-04-96		
13	9	FD144			5	22-04-96		
14	10	PC586			20	25-04-96		
15	11	FD144			5	12-04-96		
16	12	HD850			10	18-04-96		
17	Bảng phụ				Bảng tổng kết tháng			
18	Mã hàng	Tên hàng	ĐVT	Đơn giá	Mã hàng	Số lượng	Thành tiền	
19	PC486	Máy tính 486	cái	700	PC			
20	PC586	Máy tính 586	cái	1000	HD			
21	HD420	Đĩa cứng 420	ổ	80	FD			
22	HD850	Đĩa cứng 850	ổ	120				
23	FD120	Đĩa mềm 1,2	hộp	4				
24	FD144	Đĩa mềm 1,44	hộp	6				

Bài thực hành 19. Thực hiện yêu cầu dưới đây và lưu tập tin với tên *BT19.XLS*

Chú thích: 2 ký tự đầu của mã nhập chỉ mã hàng hóa, 2 ký tự sau chỉ mã sản xuất.

- Dựa vào bảng danh mục hàng hóa và danh mục sản xuất để xác định cột Tên Hàng và hãng sản xuất.
- Xác định Đơn giá dựa vào bảng Danh mục hàng hóa và lưu ý giảm giá 5% trong trường hợp sau: Mặt hàng là Tivi hoặc Cassette được bán sau ngày 20 hằng tháng.
- Thành tiền = Số lượng x Đơn giá nhưng được đổi thành USD như sau: Nếu bán trong quý IV(ba tháng cuối năm) thì 1 USD = 15.500 đồng ngược lại, 1 USD = 16.000 đồng. Kết quả làm tròn đến hàng đơn vị.
- Xác định cột thuế dựa vào cột thành tiền biết rằng: nếu mặt hàng là nồi cơm điện thì không tính thuế; Tivi hay cassette thì tính 5% còn lại 2%.

CỬA HÀNG ĐIỆN TỬ VIỄN THÔNG								
STT	MÃ NHẬP	TÊN HÀNG	HÃNG SẢN XUẤT	NGÀY NHẬP	SỐ LƯỢNG	ĐƠN GIÁ	THÀNH TIỀN	THUẾ
01	TVSS			14/07/2006	300			
02	CSSN			5/2/2006	400			
03	TVSN			18/12/2006	600			
04	TLSS			4/11/2006	800			
05	CSSN			12/10/2006	200			
06	TVSS			15/10/2006	500			
07	CSSN			11/6/2006	100			
08	COSN			14/07/2006	200			
09	MGSS			2/5/2006	300			
10	BUPL			3/2/2006	500			
11	TLHI			5/7/2006	400			
Danh mục hàng hóa								
	MÃ	TÊN HÀNG	ĐƠN GIÁ	Thông kê				
	TV	Ti vi	,500,000 đ	TÊN HÀNG	SỐ TIỀN			
	CS	Cassette	400,000 đ	Samsung				
	TL	Tủ lạnh	,200,000 đ	Sony				
	MG	Máy giặt	,300,000 đ	Hitachi				
	CO	Nồi cơm điện	300,000 đ	Phillip				
	BU	Bàn ủi điện	190,000 đ					
Danh mục sản xuất								
	MÃ	SS	SN	HI	PL			
	TÊN HÀNG	Samsung	Sony	Hitachi	Phillip			

- Sắp xếp bảng tính tăng dần theo tên hàng, nếu cùng tên hàng thì sắp xếp theo hãng sản xuất.
- Thông kê số tiền thu được vào bảng thông kê.
- Trích các mặt hàng có thuế từ 2% trở xuống.
- Dựa vào bảng thông kê vẽ đồ thị hình cột biểu diễn số tiền bán hàng thu được theo từng hãng sản xuất.

Bài thực hành 20: Trình bày và nhập bảng tính dưới đây.

Yêu cầu:

- Chèn một cột trống trước cột mã hàng và điền số thứ tự cho cột này dưới dạng **001, 002, 003 ...**
- Xác định **Tên hàng, Đơn giá** biết rằng: Nếu ký tự thứ 3 của mã hàng là 1 thì dựa vào ký tự 1 và 4, ngược lại thì dựa vào ký tự 2 và 5 của mã hàng để lấy tên hàng, đơn giá từ bảng 2.
- Xác định **Hãng sản xuất** : Dựa vào ký tự cuối của mã hàng và bảng 1.
- Xác định cột **Giảm giá**, biết rằng: **Nếu ngày bán từ 15/10/2002 đến 01/11/2002 thì được giảm 10% đơn giá.**

5. Xác định **Cột thành tiền = Số lượng x (Đơn giá – Giảm giá) x Tỷ giá**. Định dạng có dấu phân cách nhóm số và ký hiệu tiền tệ là đ (Ví dụ: 100,000 đ)

	A	B	C	D	E	F	G	H
1	BẢNG THỐNG KÊ							
2	Tỷ giá: 15800 đ							
3	Mã hàng	Tên hàng	Nước sản	Ngày bán	Số lượng	Đơn giá	Giảm giá	Thành tiền
4	MA1ASM			05/10/2002	10			
5	BH2BDT			15/12/2002	15			
6	BM2SAN			15/11/2002	5			
7	RC1ASN			25/10/2002	9			
8	RA1ALT			02/11/2002	20			
9	CR2SAM			01/12/2002	16			
10	RB1ASM			20/12/2002	10			
11								
12	Bảng 1			Bảng 2				
13	Ký tự cuối của mã hàng	Nước sản xuất		Mã	Tên hàng	Đơn giá		
14	M	Mỹ		MA	Main	125 USD		
15	N	Nhật		HD	HardDisk	15 USD		
16	T	Trung Quốc		RA	Ram	20 USD		
17				PR	Máy in	102 USD		

6. Nhập vào bảng thống kê sau và thống kê số lượng theo từng loại hàng

Tên hàng	Số lượng
<i>Main</i>	
<i>HardDisk</i>	
<i>Ram</i>	
<i>Máy in</i>	

7. Dựa vào bảng thống kê vẽ đồ thị hình bánh (Pie) biểu diễn các số liệu thống kê đó.
8. Sắp xếp bảng tính tăng dần theo **Tên hàng**, nếu cùng tên hàng thì sắp xếp giảm dần theo số lượng.
9. Rút trích các mặt hàng bán được **trong tháng 10** hoặc bán **trước ngày 15** hàng tháng

PHẦN 5.

MICROSOFT POWERPOINT 2003

1. GIỚI THIỆU POWERPOINT 2003

1.1. Giới thiệu.

Microsoft Powerpoint 2003 là một phần mềm dùng để tạo các trình diễn chuyên nghiệp có nhiều tiện ích để trình bày vấn đề có hình thức đẹp, sinh động, dễ sử dụng và hiệu quả ở dạng phim đèn chiếu, bản ghi chú, tờ quảng cáo, đề cương bài giảng,... trong cùng một tập tin trình diễn, phục vụ đông đảo khán thính giả với nhiều mục đích khác nhau. Các khung nhìn (slide) trong một tập tin powerpoint cho phép bạn chèn vào các đối tượng (Object) như âm thanh, hình ảnh, đồ thị, bảng biểu, các đối tượng đồ họa,... bên cạnh văn bản thông thường. Ngoài ra nó cũng cho phép bạn định dạng về màu sắc, hình dáng và các hiệu ứng hoạt hình,... cho các đối tượng này.

1.2. Khởi động Microsoft Powerpoint.

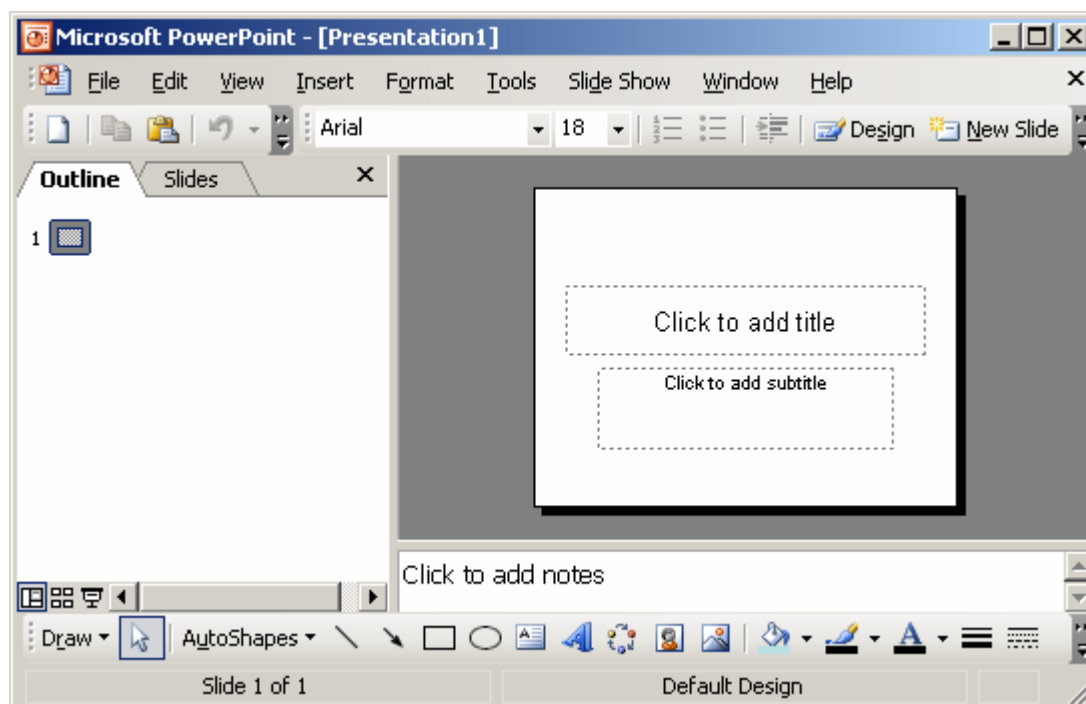
Bạn có thể khởi động Powerpoint bằng các cách sau:

- Click vào *Start\Programs\Microsoft Office\ Microsoft Power Point.*

- Click vào biểu tượng *Powerpoint*  trên desktop

1.3. Giao diện Powerpoint 2003.

Sau khi khởi động, màn hình giao diện của Powerpoint có dạng tương tự sau:



Hình 5.1

Ngoài các thành phần như thanh tiêu đề, thanh trình đơn, các thanh công cụ, thanh trạng thái, các thanh trượt tương tự các giao diện của Excel hoặc Winword, cửa sổ làm việc của Powerpoint gồm các thành phần sau:

- **Khung slide:** Là khung lớn nhất trong ba khung. Bạn có thể thêm văn bản, các hình ảnh đồ họa, các bảng, các biểu đồ, và các đối tượng khác vào trình diễn của mình trong khung slide.

- **Khung outline:** Khung này nằm bên trái cửa sổ dùng để hiển thị dàn bài của trình diễn. Bạn có thể dùng khung này để nhập nội dung văn bản, sắp xếp và bố trí lại các slide, hoặc di chuyển slide đến một vị trí khác.


- **Khung notes:** Nằm phía dưới khung slide. Dùng để ghi các chú thích.

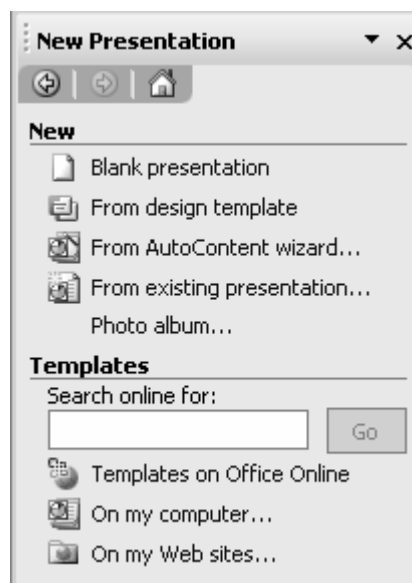
1.4. Thoát khỏi Powerpoint 2003.

- ① Chọn File\Exit.
- ② Click vào nút *Close* ở góc trên cửa sổ Powerpoint.
- ③ Nhấn tổ hợp phím *Alt + F4*.

2. THIẾT KẾ MỘT TRÌNH DIỄN.

2.1. Tạo một trình diễn mới.

- ① Cách 1: Click vào nút lệnh **New**  trên thanh công cụ.
- ② Cách 2: Thực hiện lệnh **F**ile\New, sau đó chọn **Blank presentation** và click *OK* trong hộp thoại *New Presentation*).




Hình 5. 2

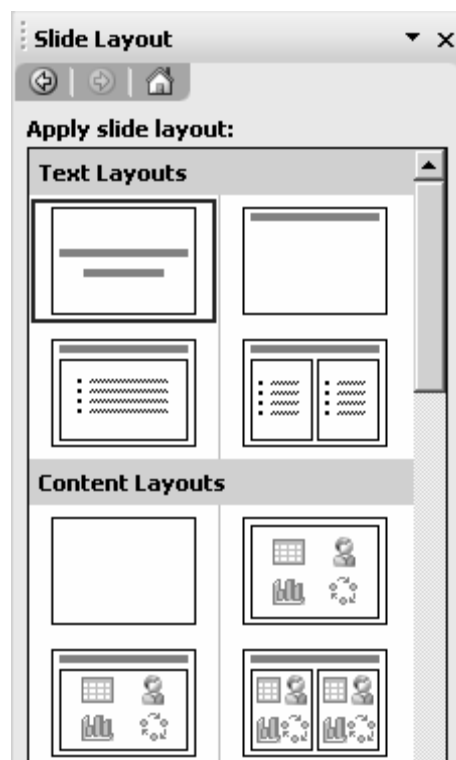
- ③ Cách 3: Nhấn tổ hợp phím **Ctrl + N**.

Các cách trên đều mở hộp thoại **New slide** để bạn chọn mẫu cho slide đầu tiên của trình diễn.

2.2. Tạo một khung nhìn (slide) mới.

Để thêm một slide vào sau slide hiện hành, bạn thực hiện như sau:

Bước 1: Chọn lệnh **Insert\New slide** hoặc nhấn tổ hợp phím **Ctrl + M** hoặc click vào nút lệnh **New slide**  trên thanh công cụ để mở hộp thoại *New slide*.



Hình 5. 3

Bước 2: Trong hộp thoại này, bạn có thể chọn 1 trong 24 mẫu slide có sẵn trong khung *Choose an AutoLayout*.

Bước 3: Click *OK*.

Mặc dù Powerpoint cung cấp một số lượng lớn các kết hợp trình bày (*AutoLayout*) để bạn lựa chọn, nhưng các cách trình bày này chỉ bao gồm tám thành phần kết hợp lại, bao gồm:

- **Tiêu đề (title):** Là một hộp văn bản (textbox) mà trong đó bạn có thể nhập tiêu đề cho slide.
- **Danh sách (list):** Chèn một danh sách với các dấu hoa thị (Bullet) hoặc chỉ mục (Numbering) ở đầu dòng.
- **Bảng (table):** Chèn một bảng có số dòng và số cột do bạn chọn.
- **Văn bản (text):** Là một hộp chứa (placeholder) văn bản trên một slide. Để bạn có thể thêm và định dạng văn bản như mong muốn.
- **Đồ thị (chart):** Chèn một đồ thị ở nhiều dạng khác nhau.
- **Sơ đồ tổ chức (organization chart):** Là một sơ đồ tổ chức ở dạng cây phân cấp.

▪ **Clipart:** Chèn một hình ảnh nghệ thuật được chọn từ thư viện hình ảnh (clip gallery) của bộ Office 2003.

2.3. Chèn các đối tượng vào slide

Có 2 trường hợp:

① Đối với các yếu tố có sẵn trong các slide mẫu, để chèn một đối tượng như văn bản, hình ảnh, bảng,... vào trong các hộp chứa (placeholder) tương ứng, bạn chỉ cần làm theo các hướng dẫn của Powerpoint ghi sẵn trong các hộp chứa đó.


Đối với các hộp chứa có ghi sẵn các từ *Click to add ...*, bạn chỉ cần click vào và bắt đầu nhập nội dung vào. Các hộp này thường chứa các tiêu đề, các hộp văn bản,....

Trong các hộp chứa có ghi sẵn các từ *Double click to add ...*, bạn cần phải click đúp vào trước khi thực hiện các bước tiếp theo. Các hộp chứa ở dạng này thường chứa các hình ảnh, các bảng, các biểu đồ,....

② Nếu muốn chèn thêm các đối tượng vào slide hiện hành, bạn sẽ thực hiện thông qua trình đơn *Insert* của Powerpoint. Riêng các đối tượng hình vẽ, bạn phải dùng thanh công cụ *Drawing* (thường nằm dưới đáy màn hình) để vẽ.

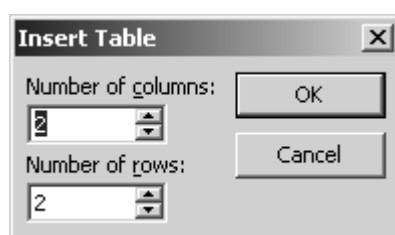
Các bước chi tiết để chèn các đối tượng vào slide được trình bày dưới đây.

2.3.1. Nhập văn bản (text).

Click vào hộp chứa văn bản, tiêu đề, bảng,... rồi nhập vào. Nếu muốn thêm một hộp văn bản (textbox) vào slide, bạn click vào nút lệnh *Textbox*  trên thanh công cụ *Drawing*. Sau đó dùng chuột để vẽ một *textbox* trên slide và nhập văn bản vào.

2.3.2. Chèn bảng (table).

Bước 1: Thực hiện lệnh *Insert\Table* hoặc click đúp vào hộp chứa bảng trên slide (nếu có) để mở hộp thoại *Insert table*.



Hình 5.4

Bước 2: Xác định số cột (*column*) và số dòng (*row*) của bảng.

Bước 3: Click OK. Sau đó nhập văn bản vào các ô.

2.3.3. Chèn hình ảnh, chữ nghệ thuật, clipart và hình vẽ

Thực hiện tương tự như cách chèn hình ảnh, chữ nghệ thuật, clipart và các hình vẽ như trong Microsoft Word 2003.

2.3.4. Chèn âm thanh và phim ảnh vào các slide

Bạn có thể chèn các tập tin âm thanh (sound) và phim ảnh (movie) vào các slide trình diễn của bạn thêm sinh động. Để thi hành được các tập tin âm thanh và phim ảnh trong quá trình trình diễn, máy tính của bạn phải có một card âm thanh, microphone, và loa.

Có 2 cách để chèn các tập tin âm thanh và phim ảnh là chèn từ Clip gallery của bộ Office 2003 hoặc từ một vị trí trên đĩa cứng do bạn chỉ định.

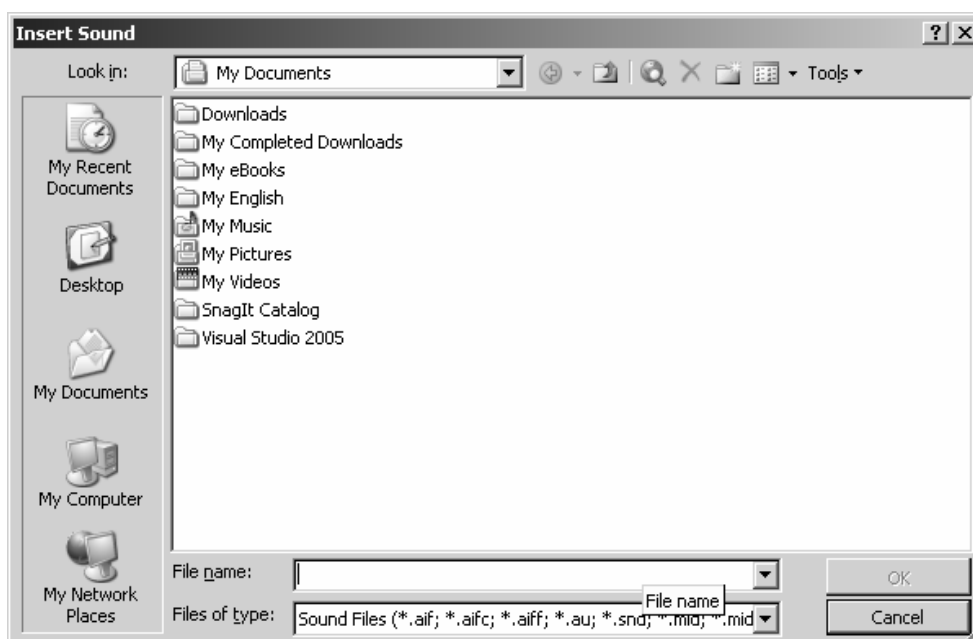
Cách thực hiện như sau:

Bước 1: Thực hiện lệnh *Insert\Movies and sounds* và chọn một trong 4 lệnh sau từ trình đơn con:

- *Insert movie from gallery:* Chèn phim ảnh từ Clip gallery.
- *Insert movie from file:* Chèn phim ảnh từ một tập tin trên đĩa.
- *Insert sound from gallery:* Chèn âm thanh từ Clip gallery.
- *Insert sound from file:* Chèn âm thanh từ một tập tin trên đĩa.

Bước 2: Để chèn phim ảnh và âm thanh từ Clip gallery, bạn thực hiện tương tự như chèn clip art.

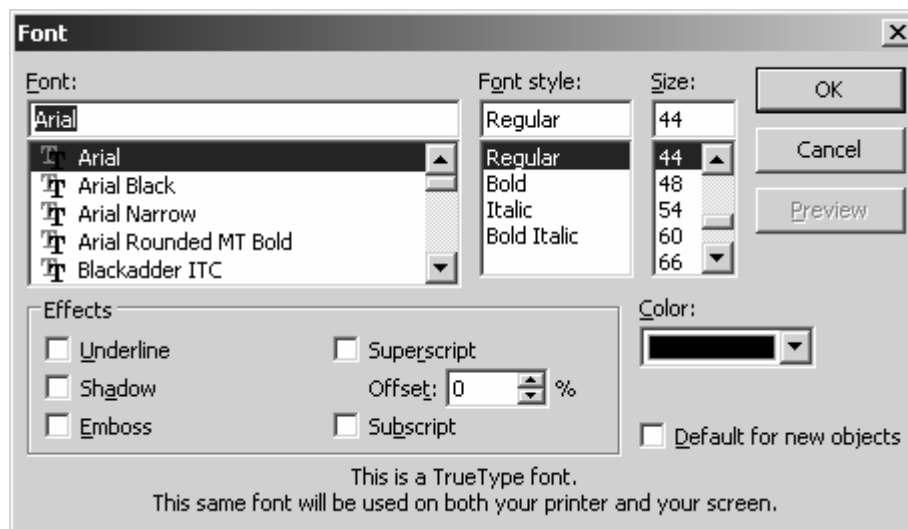
Nếu chọn lệnh chèn từ một tập tin trên đĩa, Powerpoint sẽ mở hộp thoại *Insert movie* hoặc *Insert sound* (Hình 5.5) để bạn xác định đường dẫn đến tập tin cần chèn rồi click **OK**.



Hình 5.5

2.4. Định dạng văn bản trong các slide

2.4.1. Định dạng ký tự






Hình 5.6

Bước 1: Quét chọn khối ký tự cần định dạng.

Bước 2: Dùng lệnh **Format\Font** để mở hộp thoại **Font**.

Bước 3: Chọn các tùy chọn định dạng tương tự như trong Word rồi click OK.

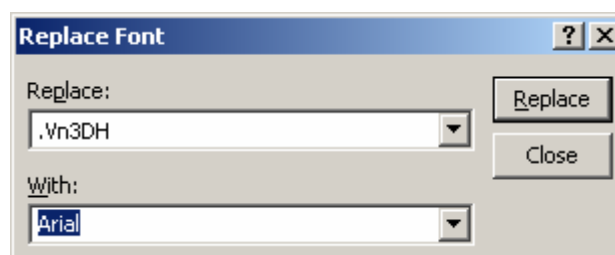
Ngoài ra, bạn có thể dùng các nút lệnh trên thanh công cụ định dạng và các nút lệnh sau để định dạng tương tự như trong Word.

- : Tăng kích cỡ ký tự lên 4 hoặc 6 points
- : Giảm kích cỡ ký tự xuống 4 hoặc 6 points
- : Tạo bóng mờ cho ký tự

* **Lưu ý:** Để thay đổi font chữ trong tất cả các slide, thay vì phải mở từng slide rồi định dạng lại như trên, bạn có thể thực hiện như sau:

Bước 1: Trong trình đơn **Format**, chọn **Replaced Fonts** để mở hộp thoại **Replace Fonts**.

Bước 2: Font chữ trong hộp **Replace** là font đang sử dụng, bạn chọn một font chữ sẽ sử dụng trong hộp **With** rồi click nút **Replace** để thay đổi.



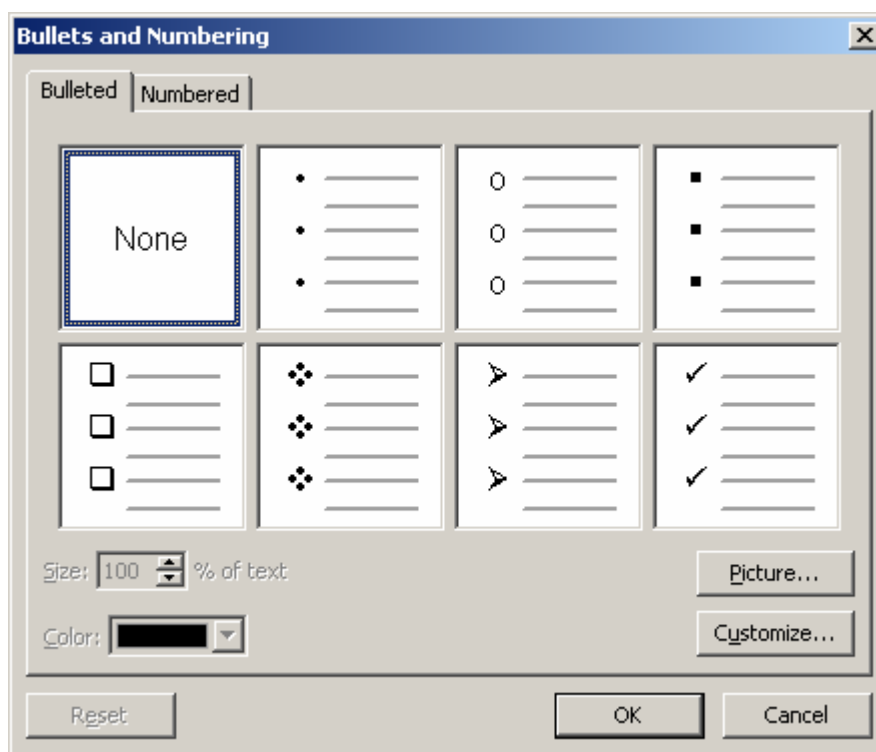
Hình 5.7

Nếu trước khi thực hiện thay đổi font mà bạn đang chọn một đối tượng nào đó thì lệnh *Replace font* chỉ có tác dụng trên đối tượng bạn đang chọn mà thôi.

2.4.2. Tạo hoa thị (bullet) và mục số (numbering).

Bước 1: Chọn khối văn bản hoặc hộp chứa muốn tạo hoa thị và mục số.

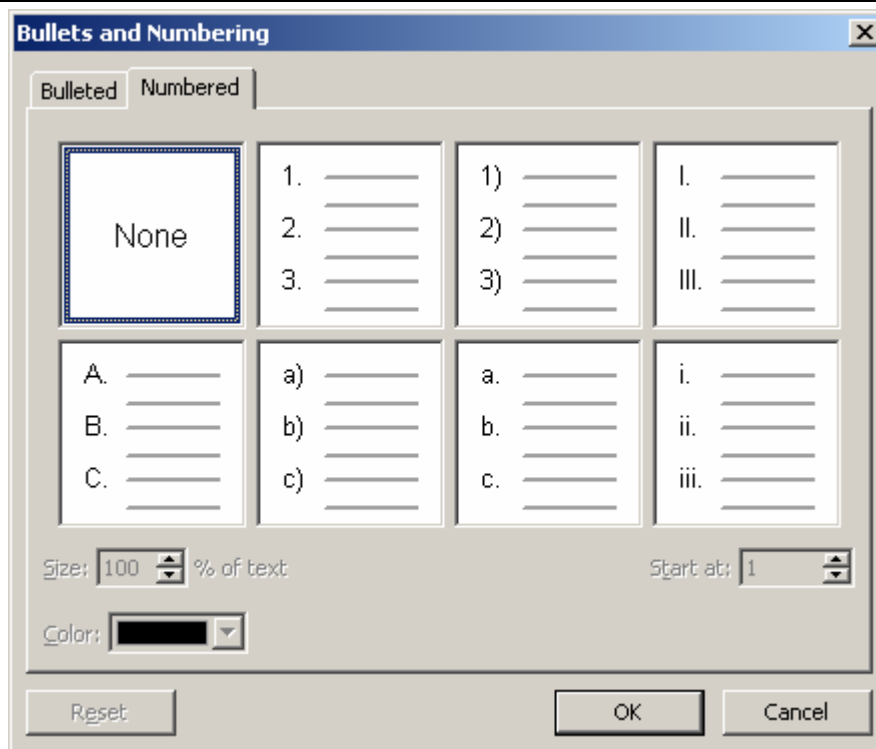
Bước 2: Dùng lệnh **Format\Bullets and Numbering** để mở hộp thoại **Bulleted** (hình 5.8) hoặc **Numbered** (hình 5.9).



Hình 5.8

Trong đó:

- **Size:** Độ lớn (tính bằng %) của ký hiệu dùng làm hoa thị/ký tự trong đoạn.
- **Color:** Màu của ký hiệu dùng làm hoa thị.
- **Picture/Character:** Chọn một hình ảnh (từ clipart)/ký hiệu khác làm hoa thị.





Hình 5.9

Trong đó:

- **Size/Color:** Tương tự trên.
- **Start at:** Số thứ tự bắt đầu đánh (có thể là một số bất kỳ).

Bước 3: Chọn các tùy chọn định dạng thích hợp rồi click **OK**.

Ngoài ra, có thể dùng các nút lệnh sau trên thanh công cụ để định dạng:

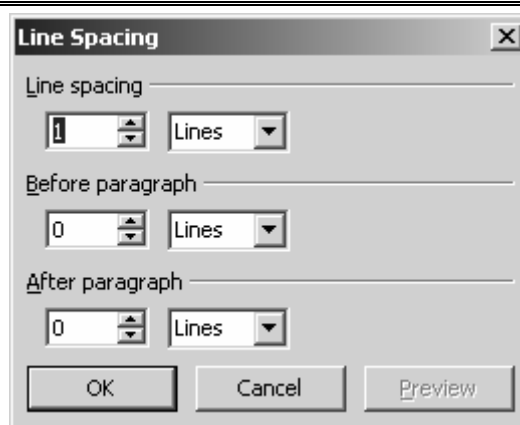
- Nút lệnh **Bullets** : Để tạo hoa thị.
- Nút lệnh **Numbering** : Để tạo mục số.

2.4.3. Định dạng đoạn (Paragraph)

2.4.3.1. Thêm khoảng cách giữa các dòng và đoạn

Bước 1: Click chuột vào bất kỳ vị trí nào trên đoạn bạn muốn thêm khoảng cách.

Bước 2: Trong trình đơn **Format**, chọn **Line Spacing** để mở hộp thoại **Line spacing**.



Hình 5.10




Bước 3: Chọn khoảng cách giữa các dòng trong cùng đoạn trong khung **Line spacing**, khoảng cách giữa đoạn đang chọn và đoạn trước trong khung **Before paragraph**, khoảng cách giữa đoạn đang chọn và đoạn kế tiếp trong khung **After paragraph**. Bạn có thể nhập khoảng cách (tính bằng dòng (lines) hoặc điểm (point) vào khung tương ứng.

Bước 4: Click *OK*.

2.4.3.2. Canh lề các dòng ký tự trong đoạn

Bước 1: Click chuột vào đoạn muốn canh lề

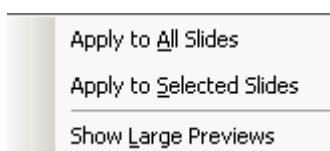
Bước 2: Trong trình đơn **Format**, click **Alignment** rồi chọn **Align Left** (canh trái), hoặc **Align Center** (canh giữa), hoặc **Align Right** (canh phải), hoặc **Align Justify** (canh đều 2 bên).

Ngoài ra bạn cũng có thể dùng các nút lệnh , ,  trên thanh công cụ để thực hiện canh trái, canh giữa và canh phải tương ứng.

2.4.4. Áp dụng các mẫu định dạng có sẵn

Bước 1: Vào menu **Format\Slide Design Template** để mở hộp thoại **Slide Design**.

Bước 2: Chọn một tập tin mẫu trong **Design Template** và click **Apply** để áp dụng các định dạng có sẵn vào tập tin đang mở.

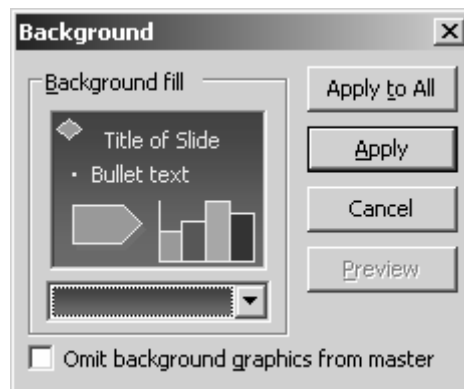




Hình 5.11

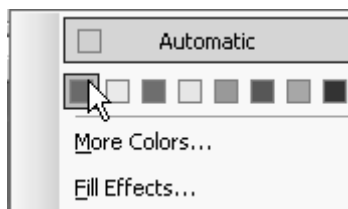
2.4.5. Chọn màu nền cho slide

Bước 1: Thực hiện lệnh **Format/Background** hoặc click phải vào một vị trí trên slide và chọn **Background** để mở hộp thoại **Background** (hình 5.12).



Hình 5.12

Bước 2: Click vào nút mũi tên để mở khung chọn màu bên dưới.



Hình 5.13

Bước 3: Thực hiện một trong các thao tác sau đây:

- Click chọn một màu có sẵn trong khung chọn màu.
- Click nút More color để chọn một màu khác.

Bước 4: Click *Apply* để áp dụng vào slide hiện hành hoặc click *Apply to All* để áp dụng cho tất cả các slide có trong trình diễn.

2.5. Tạo các hiệu ứng hoạt hình cho các đối tượng

2.5.1. Các hiệu ứng hoạt hình

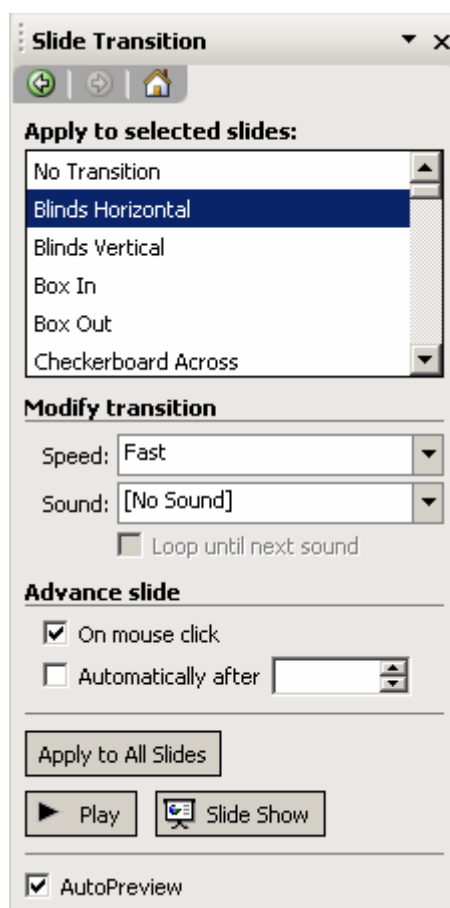
Powerpoint cung cấp rất nhiều tùy chọn liên quan đến các hiệu ứng hoạt hình trong trình diễn. Bạn có thể sử dụng các hiệu ứng hoạt hình để chuyển tiếp từ slide này đến slide khác trong trình diễn, hoặc để giới thiệu các đối tượng và văn bản trong một slide.

2.5.2. Thiết lập các hiệu ứng chuyển tiếp slide (slide transition)

Hiệu ứng chuyển tiếp slide là một trong những hiệu ứng hoạt hình thông thường nhất của Powerpoint. Bạn có thể thiết lập các hiệu ứng chuyển tiếp slide cho toàn bộ các slide có trong trình diễn hoặc chỉ cho slide hiện hành.

Cách thiết lập như sau:

Bước 1: Trong trình đơn **Slide show**, bạn chọn **Slide Transition** để mở hộp thoại **Slide transition** (hình 5.14).



Hình 5.14

Bước 2: Chọn một hiệu ứng trong danh sách và tốc độ xảy ra hiệu ứng nhanh (fast), trung bình (medium) và chậm (slow) ở khung *Effect*. Sau khi chọn một hiệu ứng, Powerpoint sẽ cho bạn xem trước hiệu ứng đó trong khung ở phía trên.

Bước 3: Nếu muốn tạo powerpoint phát ra âm thanh khi xảy ra hiệu ứng hãy chọn một tập tin âm thanh trong danh sách **Sound**.

Bước 4: Chọn cách xảy ra hiệu ứng trong khung **Advanced:**

- Click chuột: On mouse click.

- Tự động sau một khoảng thời gian tính bằng giây: **Automatically after.**

Bước 5: Click vào nút **Apply to All** để áp dụng cho tất cả các slide hoặc click **Apply** nếu chỉ muốn áp dụng cho slide hiện hành.

Để xem trước hiệu ứng chuyển tiếp vừa tạo, hãy chọn **Slide Show\Animation Preview**. Một cửa sổ nhỏ xuất hiện cho phép bạn xem hiệu ứng hoạt hình vừa tạo.

2.5.3. Sử dụng các hiệu ứng hoạt hình có sẵn

Ngoài khả năng tạo các hiệu ứng hoạt hình cho việc chuyển tiếp giữa các slide, bạn cũng có thể tạo các hiệu ứng hoạt hình cho những đối tượng trong slide, như tiêu đề, văn bản, đối tượng đồ họa,...

Để tạo hiệu ứng hoạt hình có sẵn cho một đối tượng, bạn thực hiện như sau:


Bước 1: Chọn đối tượng muốn tạo hiệu ứng.

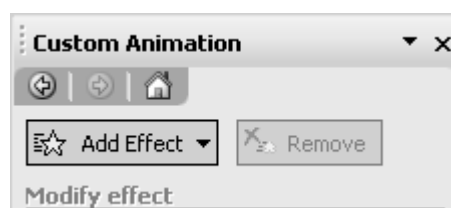
Bước 2: Trong trình đơn **Slide Show**, chọn **Preset Animation**.

Bước 3: Chọn một hiệu ứng hoạt hình có trong danh sách.

2.5.4. Tùy biến các hiệu ứng hoạt hình

Ngoài các hiệu ứng hoạt hình có sẵn, Powerpoint còn cung cấp nhiều tùy chọn tạo hiệu ứng hoạt hình cao cấp hơn như: Thiết lập thứ tự xuất hiện, xác định thời gian xuất hiện, kèm theo âm thanh,...

Bước 1: Bạn hãy chọn **Slide Show\Custom Animation** hoặc click nút **Custom Animation**  trên thanh công cụ **Animation Effects** để mở hộp thoại **Custom Animation**.



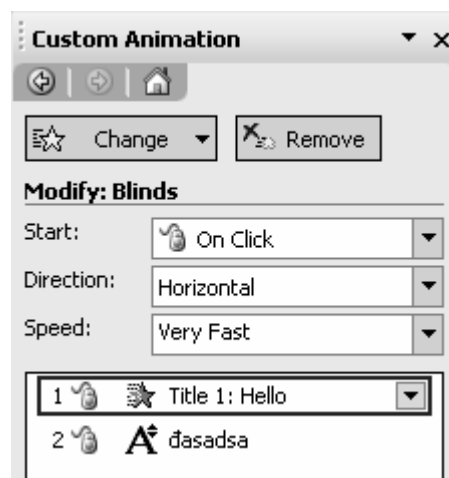
Bước 2: Xác định hiệu ứng cho từng đối tượng: Để tạo hiệu ứng cho đối tượng nào, bạn click chuột vào đối tượng đó (Ví dụ: *Title 1*) rồi thực hiện:

- Click chọn nút **Add Effects** để chọn các nhóm hiệu ứng (Hình 5.15).

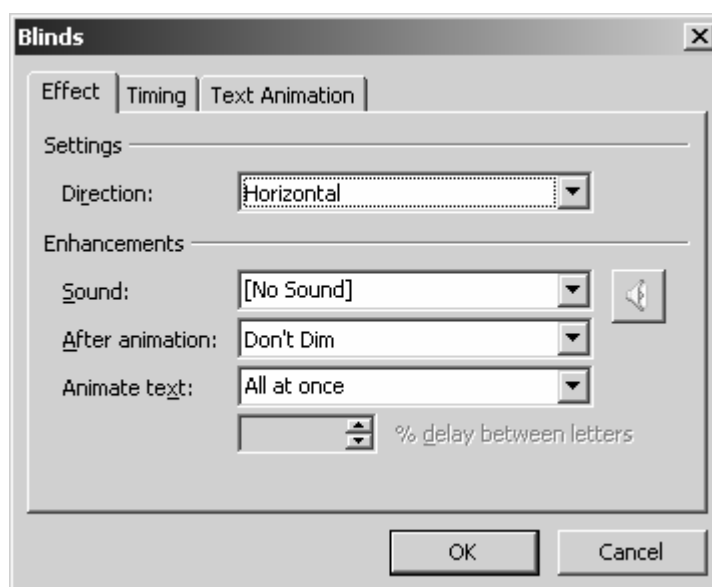


Hình 5.15

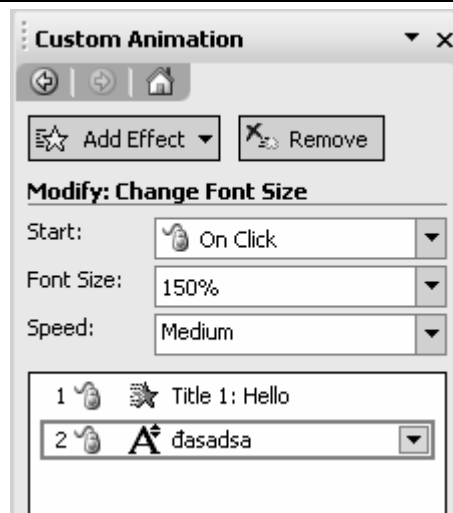
- Chọn loại hiệu ứng. Ví dụ Blinds. Nếu muốn chọn các loại hiệu ứng khác thì chọn **More Effects**
- Khi đó ta sẽ có mỗi đối tượng trong Slide sẽ có hiệu ứng tương ứng với được liệt kê theo thứ tự trong hộp thoại **Custom Animation**



- Nếu đối tượng bạn chọn là văn bản nếu muốn thiết lập chi tiết hơn cho hiệu ứng của đối tượng, click chuột phải vào hiệu ứng đã tạo chọn **Effect Options** hộp thoại dưới đây xuất hiện và chọn các thông số theo nhu cầu trình diễn của bạn.



Bước 3: Dùng chuột để để thiết lập thứ tự xuất hiện và thời gian chờ của từng đối tượng trong hộp thoại.



Hình 5.16

- Trong khung **Start**, bạn chọn **On click** nếu muốn kích hoạt hiệu ứng hoạt hình bằng cách click chuột, hoặc chọn **With Previous** (hoặc **After Previous**) tùy theo nhu cầu trình diễn của bạn.

Bước 4: Click nút **Preview** để xem trước các thiết lập của bạn.

Bước 5: Click nút **OK** để lưu lại các thiết lập đó.

2.6. Các thao tác trên các slide

2.6.1. Sao chép slide trong cùng một trình diễn

Bước 1: Chọn slide muốn sao chép.

Bước 2: Trên trình đơn **Edit**, chọn **Duplicate** hoặc nhấn tổ hợp phím **Ctrl + D**.

2.6.2. Xoá bỏ slide

Bước 1: Chọn slide muốn xoá.

Bước 2: Trên trình đơn **Edit**, chọn **Delete Slide** hoặc nhấn phím **Delete**.

2.6.3. Phóng to hoặc thu nhỏ một slide

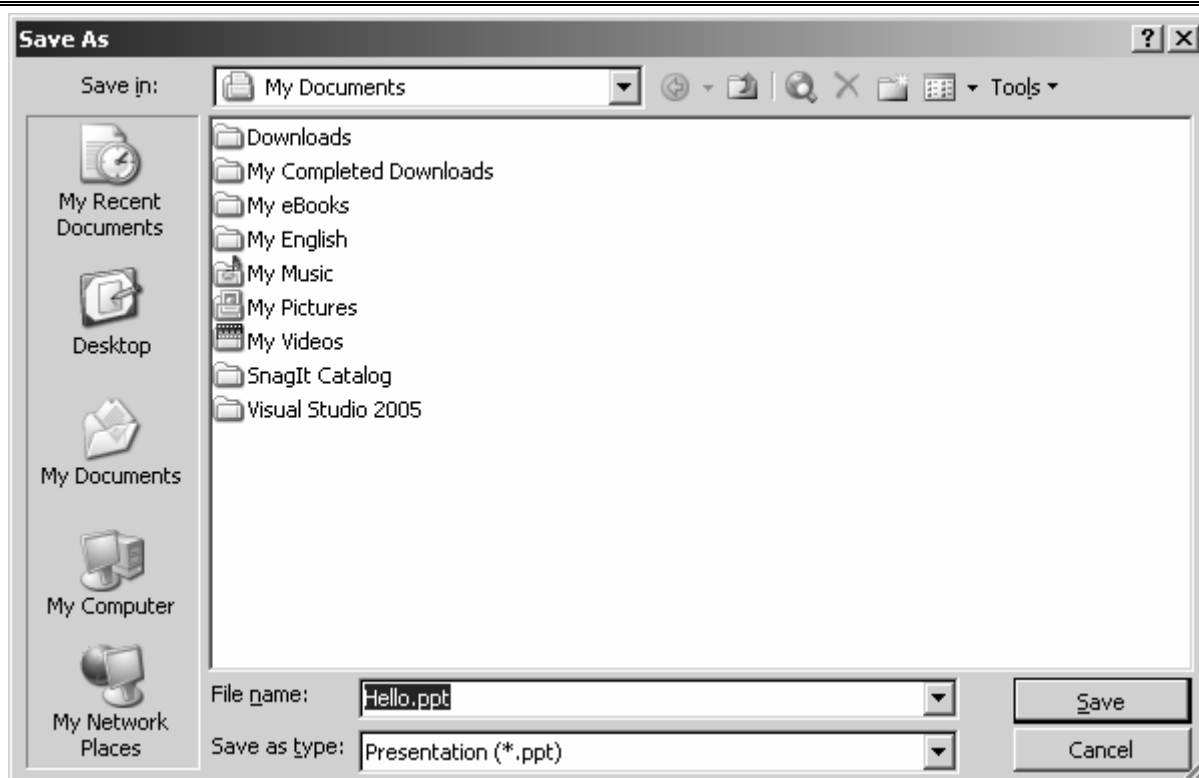
Trên thanh công cụ chuẩn, bạn click vào mũi tên bên cạnh nút **Zoom** (50%) rồi chọn tỷ lệ phóng đại thích hợp.

2.7. Các thao tác trên tập tin trình diễn

2.7.1. Sao lưu một trình diễn

Để lưu một trình diễn mới được tạo, bạn thực hiện theo các bước sau:

Bước 1: Click vào nút lệnh **Save** (đĩa mềm icon) trên thanh công cụ hoặc nhấn tổ hợp phím **Ctrl+S**. Hộp thoại **Save as** sẽ xuất hiện.




Hình 5.17

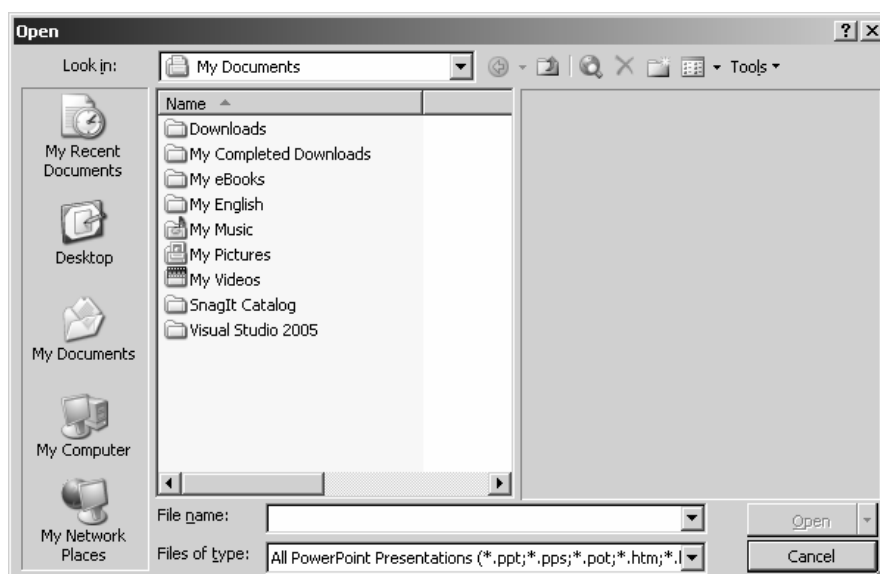
Bước 2: Chọn thư mục hay ổ đĩa bạn muốn lưu vào từ danh sách đồ xuống *Save in*.

Bước 3: Nhập tên tập tin vào hộp *File name*. Tên tập tin Powerpoint có phần mở rộng mặc định là **.PPT**.

Bước 4: Click vào nút *Save* để lưu tập tin.

2.7.2. Mở một trình diễn

Bước 1: Click vào nút *Open*  trên thanh công cụ hoặc nhấn tổ hợp phím **Ctrl+O** để mở hộp thoại **Open**.



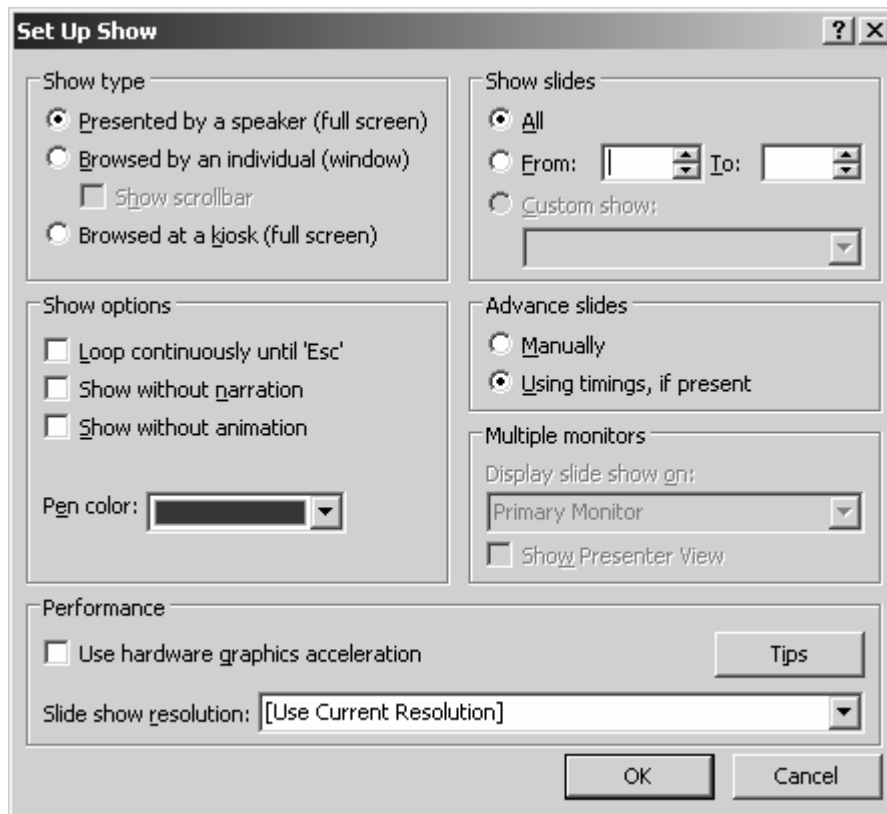
Hình 5.18

Bước 2: Chọn thư mục hay ổ đĩa chứa tập tin trong danh sách đổ xuống *Save in*, rồi sau đó chọn tập tin cần mở và click nút lệnh **Open**, *Powerpoint* sẽ mở tập tin đã chọn.

3. THI HÀNH MỘT TẬP TIN TRÌNH DIỄN

3.1. Thiết lập các tùy chọn cho trình diễn

Bước 1: Trong trình đơn **Slide show**, chọn **Set Up Show** để mở hộp thoại **Set Up Show** (hình 5.19).



Hình 5.19

Bước 2: Chọn **Presented by a speaker (full screen)** trong phần **Show type** để thực hiện trình diễn của bạn trên toàn bộ màn hình.

Bước 3: Giới hạn các slide bạn muốn sử dụng trong trình diễn:

- *All*: Trình diễn tất cả các slide.
- *From ... To ...* : Chỉ trình diễn một nhóm các slide liên tục nhau do bạn chỉ định bằng cách nhập số thứ tự slide bắt đầu và kết thúc vào ô tương ứng.

Bước 4: Trong phần *Advance slides*, chọn tùy chọn *Manually* để bạn tự điều khiển trình diễn bằng cách dùng các thao tác điều khiển.

Bước 5: Click *OK* để đóng hộp thoại.

3.2. Thực hiện một trình diễn

Để thực hiện một trình diễn, bạn thực hiện các bước sau:

Bước 1: Mở tập tin trình diễn.

Bước 2: Bắt đầu trình diễn theo một trong hai cách sau:

- Vào trình đơn *Slide Show*, chọn *View show*.
- Nhấn phím **F5**.

Bước 3: Sử dụng các thao tác điều khiển sau:

- Chuyển tới slide kế tiếp hoặc thi hành hiệu ứng hoạt hình kế tiếp: Click chuột hoặc nhấn một trong các phím **→**, **Enter**, **PgDn**, **↓**
- Quay lại slide trước đó hoặc thi hành lại hiệu ứng hoạt hình trước đó: Sử dụng một trong các phím: **←**, **Backspace**, **PgUp**, **↑**

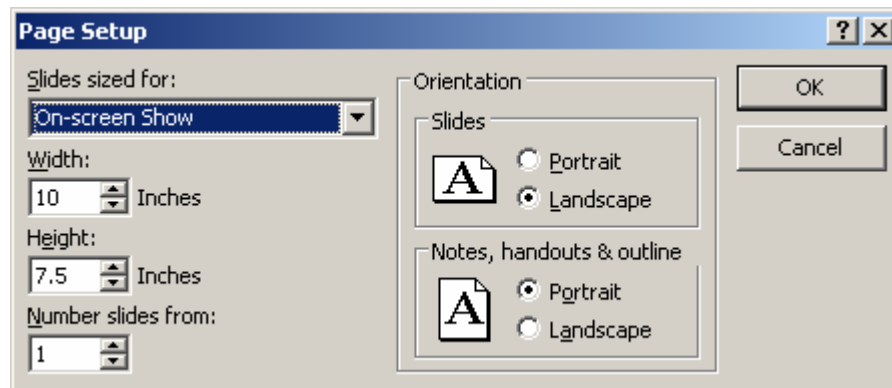
Bước 4: Thoát khỏi chế độ trình diễn bằng phím **ESC**.

3.3. In các trình diễn

3.3.1 Thiết lập các tùy chọn trong Page Setup

Trước khi in chúng ta cần phải thiết lập một số thuộc tính cho trang in như kích thước trang in, hướng trang in.

Chọn File, vào Page Setup để mở được hộp thoại sau



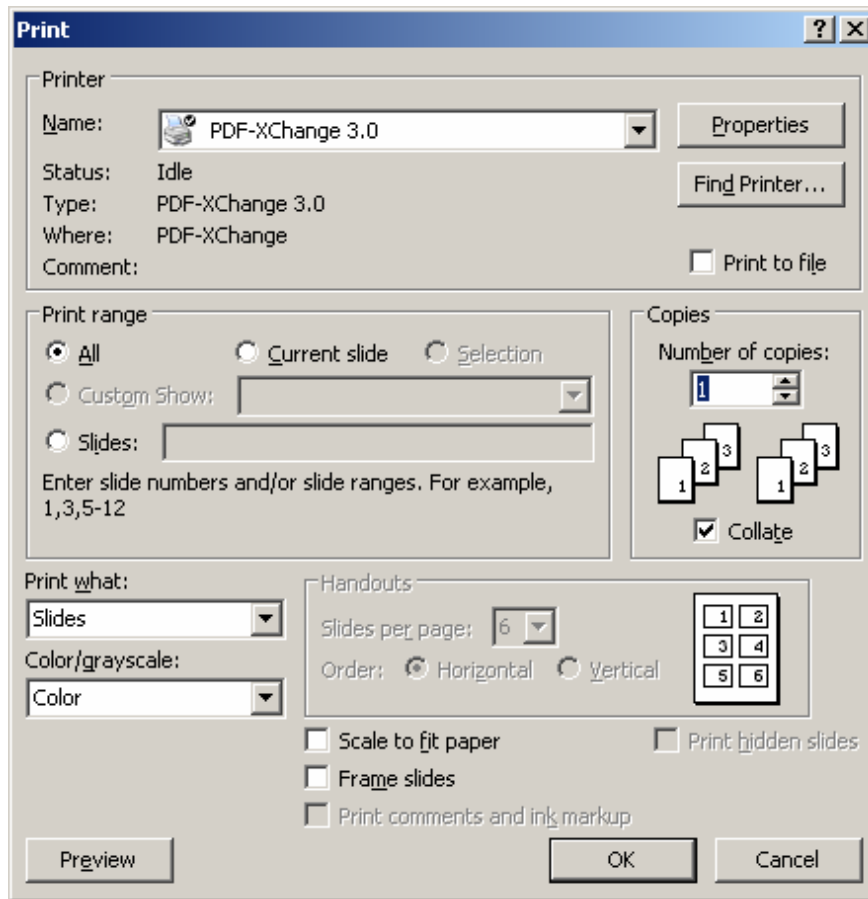
Hình 5.20

- ◆ *Slides sized for*: Chọn kích thước trang in.
- ◆ *Orientation*: Chọn hướng trang in.
 - *Portrait*: In thẳng đứng.
 - *Landscape*: In nằm ngang.
- ◆ *Notes, handouts & outline*: Chọn hướng in cho các ghi chú (notes) và dàn bài (Outline) theo kiểu thẳng đứng (*Portrait*) hoặc nằm ngang (*Landscape*)

3.3.2 Tùy chọn các thông số trong hộp thoại Print

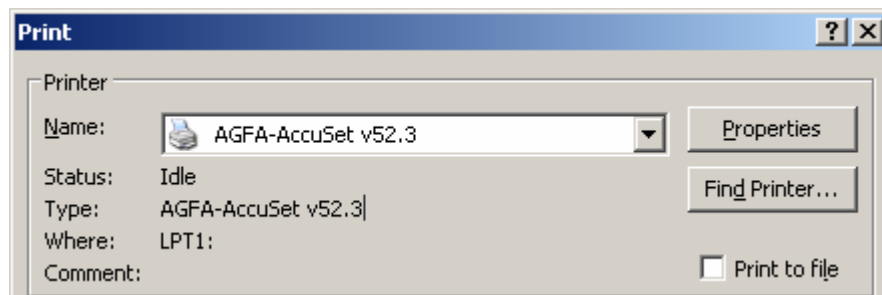
Nhằm tiết kiệm giấy mực chúng ta cần phải thiết lập các thuộc tính của trang in phù hợp với nhu cầu

- ◆ Chọn File vào Print để mở được hộp thoại sau



Hình 5.21

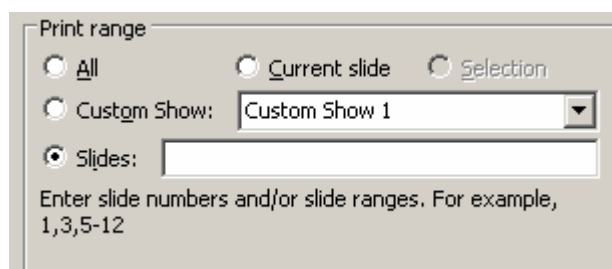
Trong khu vực Print của hộp thoại chọn máy in có từ danh sách máy in có thể kết nối với máy tính của bạn.



Hình 5.22

Nếu nhấp chuột vào nút Properties một hộp thoại sẽ hiện lên cho ta hiệu chỉnh các thuộc tính của máy in như khổ giấy, độ phân giải .v.v

Trong khu vực Print Range chọn các Slide muốn in như sau



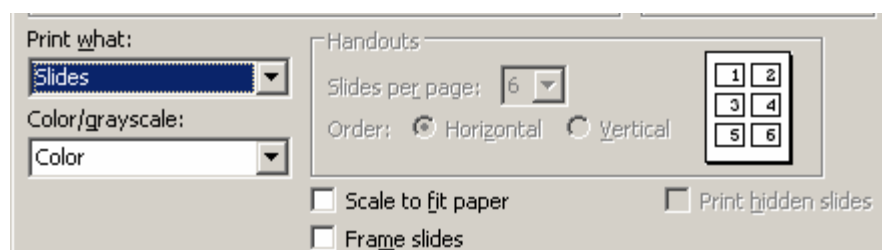
Hình 5.23

- All: In tất cả các Slide.
- Current Slile: Chỉ in slide hiện hành.
- Selection: In các Slide đã được chọn.
- Custom Show: Cho phép chọn in một Custom Show. Lựa chọn này chỉ có hiệu lực khi bạn đã tạo tối thiểu một Custom Show.
- Slide: Nhập số thứ tự của các Slide cần in.

Trong khu vực Copy nhập số bản in muốn in vào mục Number of copies và xem có chọn Collate hay không

- Nếu chọn Collate: Các bản in được in từ Slide 1 đến Slide cuối cùng sau đó mới lặp lại.
- Nếu không chọn Collate: In hết số lượng trang 1, rồi in đến số lượng trang 2 và tiếp tục như vậy cho đến hết.

Trong các khu vực còn lại.



Hình 5.23

Trong phần Print what ta cần xác định loại tài liệu muốn in từ trong danh sách.

Trong mục Slides per paper ta cần xác định số lượng Slide cần in trong một trang giấy.

Trong mục Order ta chọn để sắp xếp các slide trên trang in theo chiều ngang hay theo chiều dọc.

Một số tùy chọn khác:

- ◆ **Grayscale:** In trình diễn theo tông màu xám (đen trắng và xám). Đây là tùy chọn tốt nhất khi in bằng máy in trắng đen.
- ◆ **Pure Black and White:** In trình diễn chỉ bằng 2 màu đen trắng.
- ◆ **Include animations:** hiển thị các hoạt hình trên trình diễn của bạn giống như như biểu tượng trên trang in.

- ◆ **Scale to fit paper:** Thay đổi kích thước của Slide cho vừa với khổ giấy.
- ◆ **Frame slides:** Tạo đường viền quanh các slide. Tùy chọn này sẽ không có hiệu lực khi in chế độ Outline View.
- ◆ **Print hidden slides:** Tùy chọn này chỉ có hiệu lực khi các Slide bị ẩn.

Sau đó nhấn OK để in.

BÀI TẬP

Bài 1:

Trình bày 2 slide theo mẫu và lưu vào đĩa Z: với tên **THI-TRUONG.PPT**

Yêu cầu:

- Tô màu nền với mẫu pattern có tên Divot, foreground là màu trắng, background là màu xanh lam nhạt.
- Thiết lập hiệu ứng chuyển tiếp slide và hiệu ứng chuyển động cho các đối tượng trên slide.



Slide 2: Gold Price (GIÁ VÀNG). The slide has a light blue background with a Divot pattern. The title "GIÁ VÀNG" is centered in bold red text, with the date "Ngày 16/08/2006" below it. A table with 3 columns and 5 rows is centered on the slide. The columns are labeled "Địa điểm", "Mua", and "Bán". The rows list four locations: Hà Nội, TP HCM, Đà Nẵng, and Cần Thơ, with their respective buy and sell prices. The unit "Đơn vị: VNĐ/1 chỉ" is centered below the table.

Địa điểm	Mua	Bán
Hà Nội	1.243.000	1.263.000
TP HCM	1.245.000	1.260.000
Đà Nẵng	1.245.000	1.245.000
Cần Thơ	1.245.000	1.245.000

Đơn vị: VNĐ/1 chỉ

Bài 2:

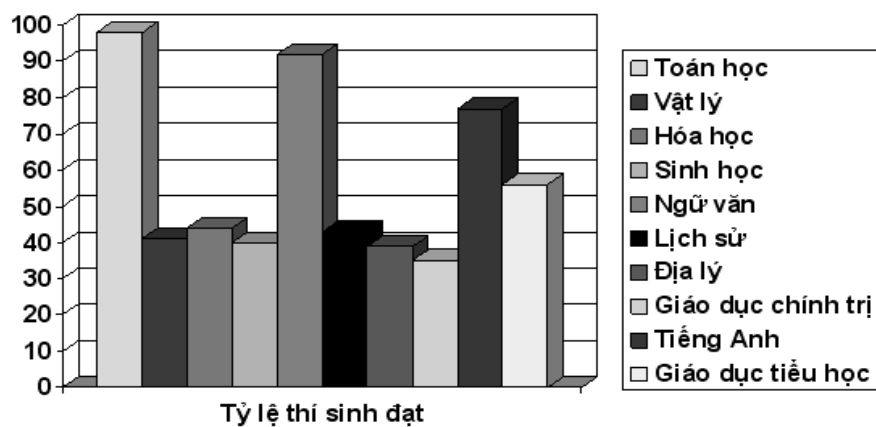
Tạo 2 slide theo mẫu và thực hiện các yêu cầu sau:

1. Chọn lựa một mẫu thiết kế tùy ý.
2. Tạo hiệu ứng chuyển tiếp Slide ngẫu nhiên cho các Slide.
3. Tạo hiệu ứng hoạt hình ngẫu nhiên cho các thành phần trong mỗi Slide
4. Lưu nội dung thiết kế lên đĩa **Z:** với tên **TUYENSINH2006.PPT**

Thống kê số liệu tuyển sinh 2006 Các ngành học thuộc Khoa sư phạm

Toán học	98
Vật lý	41
Hóa học	44
Sinh học	40
Ngữ văn	92
Lịch sử	43
Địa lý	39
Giáo dục chính trị	35
Tiếng Anh	77
Giáo dục tiểu học	56

Biểu đồ so sánh tỷ lệ thí sinh đạt Các ngành thuộc Khoa sư phạm



Bài 3:

Câu 1: Nhập và trình bày hai Slide theo mẫu. Lưu vào ổ đĩa Z: với tên **QUANGCAO.PPT**.

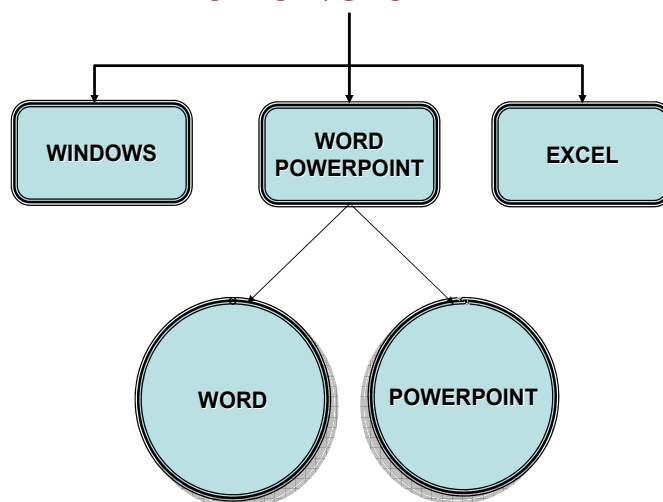
Câu 2: Tạo màu nền cho hai slide với mẫu Texture bất kỳ.

Câu 3: Tạo hiệu ứng hoạt hình và hiệu ứng chuyển tiếp bất kỳ và đầy đủ cho các thành phần trên mỗi Slide.

TRUNG TÂM TIN HỌC ĐHAG

- ✂ Máy lạnh, mỗi học viên một máy.
 - ✂ Dễ hiểu.
 - ✂ Giáo trình miễn phí.
 - ✂ Giáo viên có kinh nghiệm, tận tâm.
- ✓ Từ đầu tháng 9-2007, có thể xem điểm thi qua trang web: <http://cict.agu.edu.vn>

CHỨNG CHỈ A



Bài 4:

Câu 1: Nhập và trình bày các Slide theo mẫu. Lưu vào ổ đĩa Z: với tên Bai4.PPT.

Câu 2: Tạo màu nền cho hai slide với mẫu Texture bất kỳ.

Câu 3: Tạo hiệu ứng hoạt hình và hiệu ứng chuyển tiếp bất kỳ và đầy đủ cho các thành phần trên mỗi Slide.

Soạn thảo một báo cáo hiệu quả trên máy tính

Trình bày một cấu trúc điển hình cho một báo cáo trên máy tính hiệu quả

Văn bản

- Dùng một chủ đề trên một slide
- Dùng tiêu đề slide để làm nổi bật
 - Các nội dung chi tiết sẽ trình bày
 - Giới hạn nội dung từ 5-7 dòng

Kích thước bài báo cáo

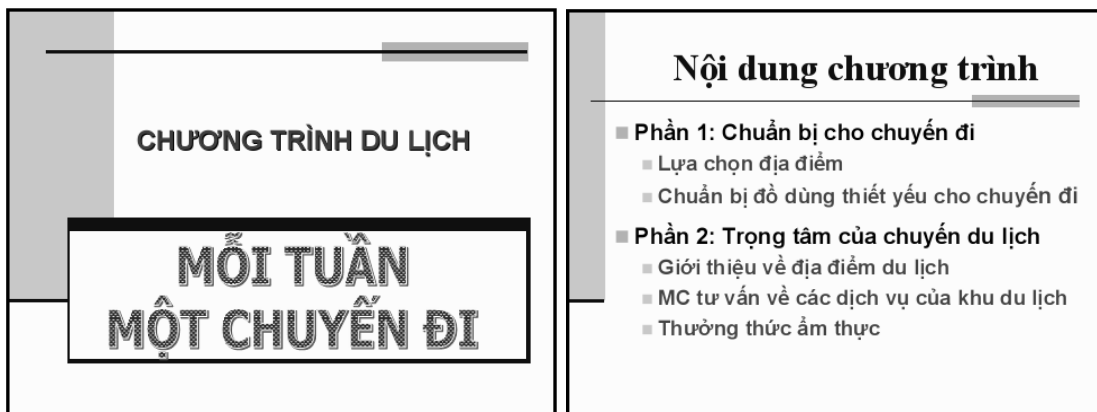
- Số slide trình bày giới hạn bởi thời gian
 - Quy tắc chung là 1 slide/ phút
 - Dùng 20 slides cho một báo cáo trong 15 phút là rất khó trình bày hết ý tưởng
- Kích thước tập tin giới hạn thời gian tải lên máy
 - Tập tin nên giới hạn ở 15 MB

Định dạng tập tin

- Tập tin nên được định dạng sao cho Power Point 2000 có thể đọc được đúng nội dung, tránh lưu ở các định dạng quá mới.
- Tập tin nên chuyển sang định dạng của Power Point 2000 nếu chạy trên Office 2000.
- Tất cả fonts, hình, bảng, và biểu đồ phải được nhúng vào PowerPoint, tránh dùng link

Bài 5: Thiết kế Slide theo mẫu. Sử dụng Template và hiệu ứng bất kỳ

(Trích đề thi chứng chỉ A Tin học – Trung tâm Tin học Đại học An Giang)



Bài 6: Thiết kế Slide theo mẫu. Sử dụng Template và hiệu ứng bất kỳ

(Trích đề thi chứng chỉ A Tin học – Trung tâm Tin học Đại học An Giang)



Bài 7: Thiết kế Slide theo mẫu. Sử dụng Template và hiệu ứng bất kỳ



PHẦN 6. PHỤ LỤC

TẠO MỤC LỤC, MAIL MERGE VÀ TRACK CHANGE

Đối với các tài liệu dài chúng ta thường có nhu cầu tạo mục lục. Việc này có thể làm bằng tay, tuy nhiên, khi phải chỉnh sửa một số nội dung nào đó cũng sẽ làm cho mục lục bị sai lệch về số trang hoặc đề mục. Microsoft Word cung cấp sẵn cho chúng ta chức năng tạo mục lục tự động rất chuyên nghiệp, chúng ta cũng có thể cập nhật lại bảng mục lục rất nhanh và chính xác khi hiệu chỉnh nội dung tài liệu. Ngoài ra, khi xem tài liệu trên máy tính, từ bảng mục lục chúng ta có thể đến nhanh một đề mục nào đó chỉ bằng một cái click chuột.

Trong phạm vi chuyên đề này, chúng ta sẽ tìm hiểu 2 cách tạo mục lục: dựa trên style và tạo thủ công.

6.1 TẠO MỤC LỤC DỰA TRÊN STYLE

Style là một tập hợp các định dạng văn bản của một đoạn văn bản. Để tạo mục lục dựa trên style, bạn phải định dạng style cho các đề mục muốn đưa vào bảng mục lục, nghĩa là áp dụng các mức tiêu đề Heading 1, Heading 2, ... cho các đề mục. Chúng ta cũng có thể quy định lại font, kích cỡ chữ, canh lề cho các mức tiêu đề Heading.

6.1.1 Sử dụng thanh công cụ Outlining


Thanh công cụ Outlining hỗ trợ định dạng style cho tài liệu. Bạn có thể bật / tắt thanh công cụ này bằng cách vào **View \ Toolbars \ Outlining**.



	Promote to Heading 1: đưa khối văn bản được chọn lên cấp style cao nhất (Heading 1).
	Promote: đưa khối văn bản được chọn lên một cấp style.
	Outline level: chỉ định cấp style cụ thể cho khối văn bản được chọn.
	Demote: giảm khối văn bản được chọn xuống một cấp style.
	Demote to Body Text: giảm khối văn bản được chọn xuống cấp style thấp nhất (Body Text)
	Update TOC: cập nhật lại bảng mục lục.
	Go to TOC: đưa con trỏ đến bảng mục lục.

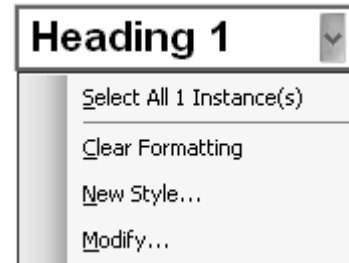
6.1.2 Tạo Heading:

Có thể tạo heading cho đề mục bằng một trong các cách sau:

- Trên thanh định dạng **Formatting**, vào hộp thoại style chọn mức Heading tương ứng.
- Click nút  hoặc vào menu **Format \ Styles and Formatting** để mở task pane **Styles and Formatting**, chọn mức Heading tương ứng.
- Sử dụng các nút trên thanh công cụ Outlining.

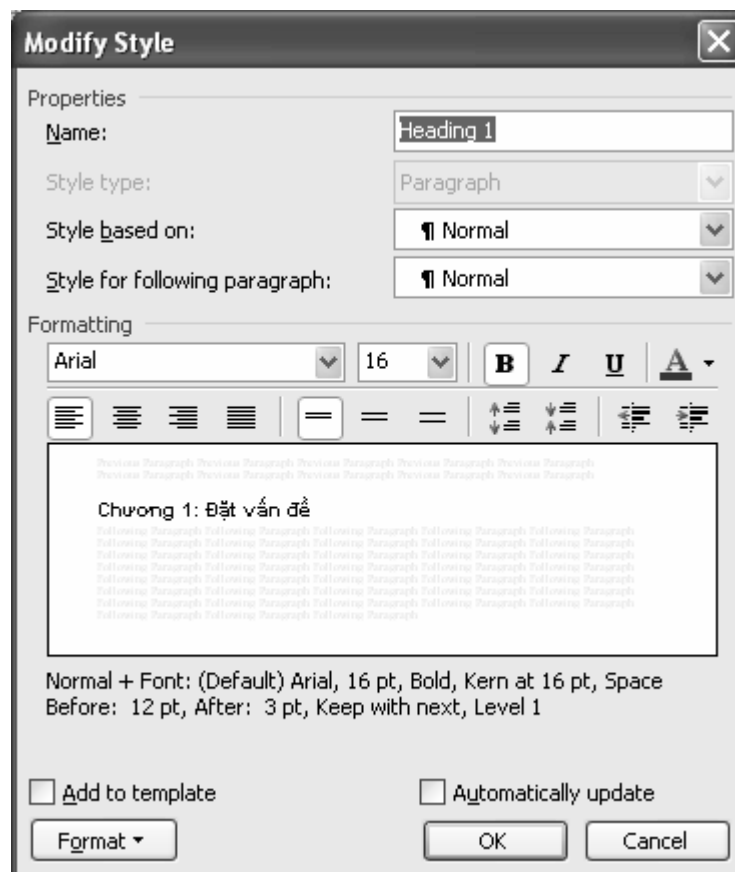
6.1.3 Thay đổi định dạng Heading:

Trên task pane Styles and Formatting, lướt chuột vào Heading muốn thay đổi định dạng sao cho bên phải xuất hiện mũi tên, click chuột vào mũi tên để mở hộp thả và chọn Modify



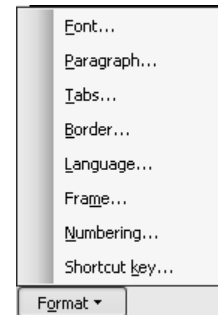
Xuất hiện hộp thoại Modify Style:

- Trong phần Formatting, sử dụng các nút định dạng được cung cấp sẵn, định dạng này sẽ áp dụng cho Heading hiện hành (tên Heading hiện hành trong ô Name)



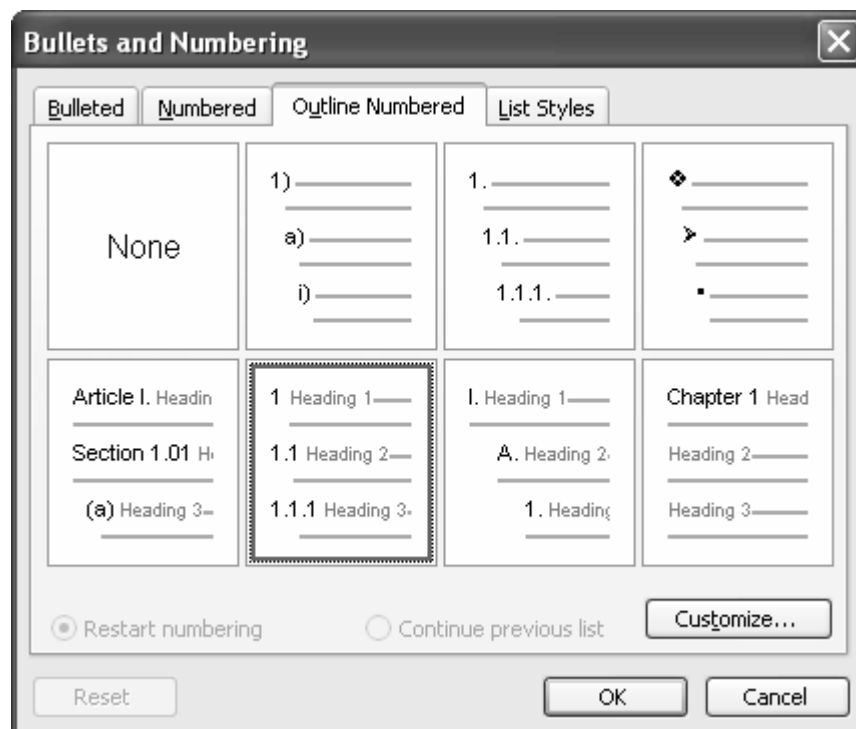
- **Add to template:** áp dụng định dạng này cho các file sau. Nếu không chọn thì định dạng này chỉ áp dụng cho file hiện hành.

- **Automatically update:** Heading tự động cập nhật khi thay đổi thêm định dạng khác, không cần vào hộp thoại Modify Style để sửa đổi.
- Nút **Format:** mở các hộp thoại định dạng cho heading:
 - **Font:** định dạng font chữ.
 - **Paragraph:** định dạng đoạn.
 - **Tabs:** định dạng tab.
 - **Border:** định dạng đường viền.
 - **Language:** thay đổi kiểm tra lỗi chính tả ngôn ngữ.
 - **Frame:** đặt heading thành frame có thể di chuyển ở mọi vị trí.
 - **Numbering:** đánh số thứ tự, kí tự đầu dòng tự động.
 - **Shortcut key:** sửa hoặc bổ sung phím tắt cho heading.

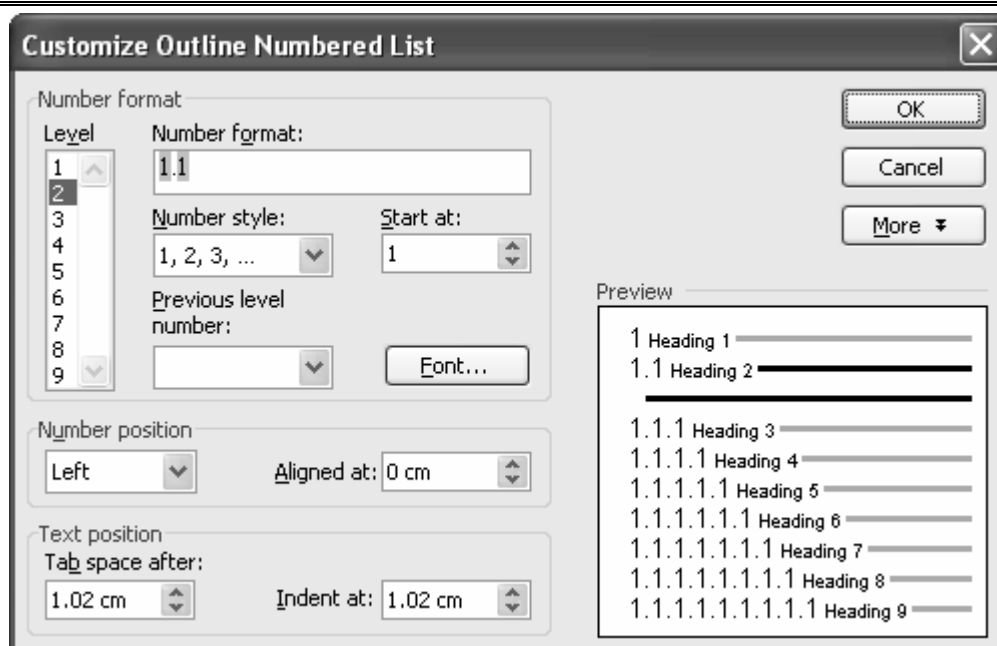


6.1.4 Định dạng Heading với Outline Numbered:

Nếu các đề mục trong bảng mục lục cần tạo có đánh số thứ tự bạn nên áp dụng Outline Numbered để định dạng. Vào **Format \ Bullets and Numbering**, chọn thẻ Outline Numbered trong hộp thoại Bullets and Numbering



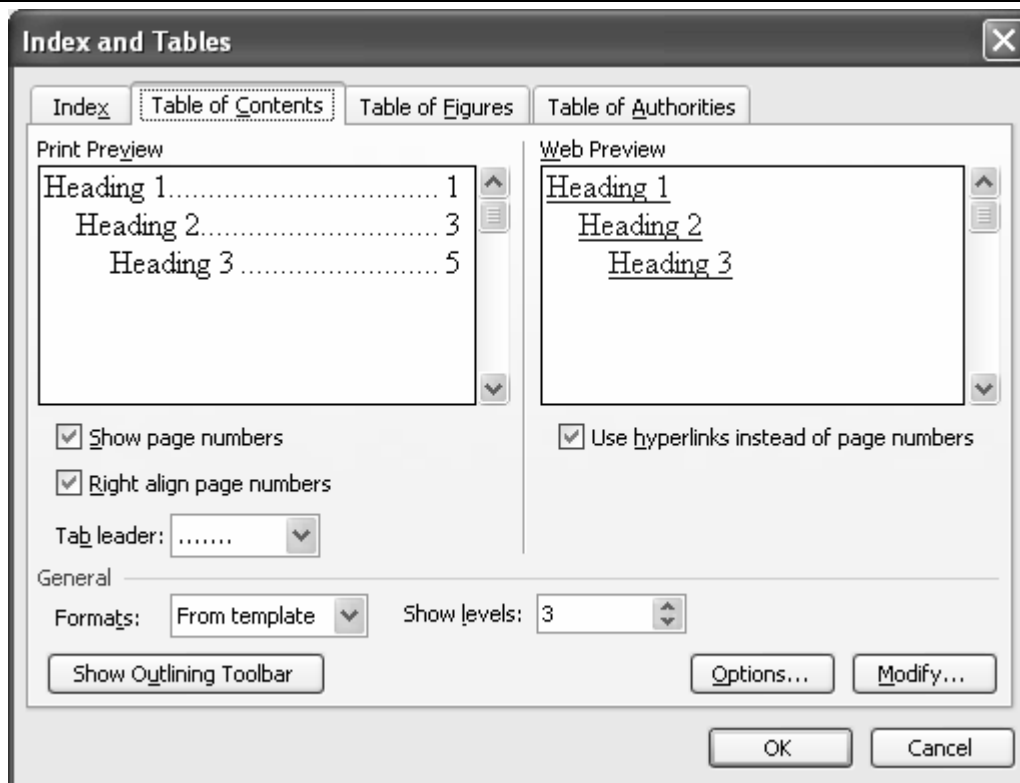
Chọn một trong các dạng Outline Numbered được cung cấp sẵn, nếu không có dạng trình bày như ý, bạn có thể quy định lại bằng cách click nút **Customize**



- **Mục Number format:**
 - **Level:** mức tiêu đề.
 - **Number format:** định các ký tự kèm theo ký số (ví dụ: chương, phần, ...).
 - **Number style:** chọn kiểu đánh số (ví dụ: 1, I, a, A,...).
 - **Start at:** số thứ tự bắt đầu.
 - **Previous level number:** cho hiển thị ký số mức tiêu đề cha.
 - **Font:** định font cho số thứ tự.
- **Mục Number position:** định vị trí của ký số, có thể định chính xác trong mục Aligned at.
- **Mục Text Position:**
 - **Tab space after:** định khoảng trắng theo sau ký số.
 - **Indent at:** canh lề cho đoạn văn bản áp dụng mức tiêu đề đó.
- Để áp dụng định dạng này cho mức tiêu đề heading nào, click nút More, trong mục Link level to style chọn mức heading tương ứng.

6.1.5 Chèn bảng mục lục

Đặt con trỏ tại vị trí muốn chèn bảng mục lục (sau khi đã áp dụng style cho các đề mục), vào menu **Insert \ Reference \ Index and Tables** để mở hộp thoại Index and Tables



Chọn thẻ **Table of Contents**:

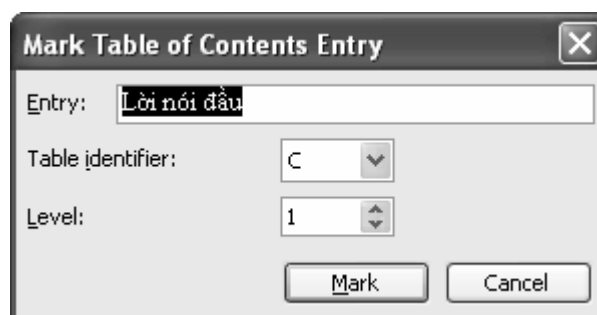
- **Print Preview**: kiểu mục lục hiển thị khi in.
 - **Show page numbers**: hiển thị số trang.
 - **Right align page numbers**: canh lề phải cho số trang.
 - **Tab leader**: chọn kiểu kí tự tab giữa đề mục heading và số trang.
- **Web Preview**: kiểu mục lục hiển thị ở dạng web.
 - **Use hyperlinks instead of page numbers**: sử dụng liên kết thay vì hiển thị số trang.
- **Show levels**: số cấp độ heading muốn đưa vào bảng mục lục.
- **Nút Show Outlining Toolbar**: hiển thị thanh công cụ Outlining.
- **Nút Options**: mở hộp thoại Table of Contents Options, tùy chọn thêm một số tính năng khác.
- **Nút Modify**: mở hộp thoại Style, định font chữ cho bảng mục lục.

6.2 TẠO MỤC LỤC THỦ CÔNG

Định dạng dựa trên style tạo cho tài liệu của bạn có phong cách soạn thảo tốt. Tuy nhiên nếu thấy hơi bị gò bó, bạn có thể tạo mục lục thủ công. Cách này bạn sẽ tạo từng Field Code loại TC (table of contents entry) cho các đề mục muốn đưa vào bảng mục lục.

6.2.1 Đánh dấu đề mục

Tô đề mục và nhấn tổ hợp phím **Alt + Shift + O** để mở hộp thoại **Mark Table of Contents Entry**:



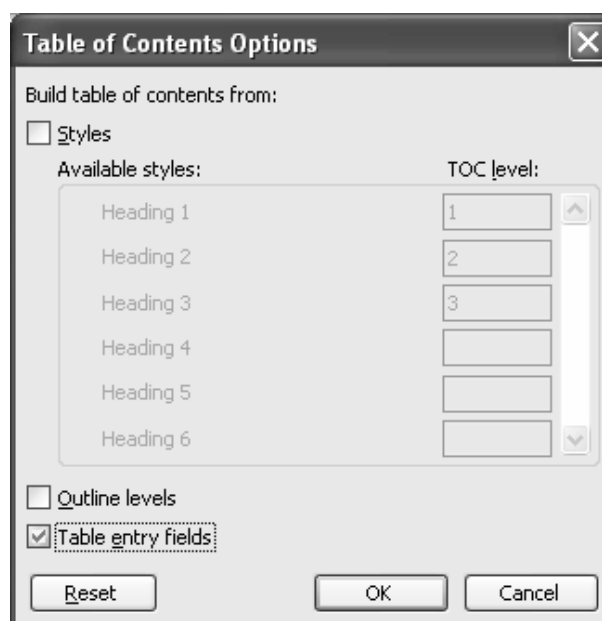
- **Entry**: ô hiển thị nội dung đề mục muốn đưa vào bảng mục lục.
- **Table identifier**: dùng để phân biệt đề mục này sẽ thuộc bảng nào, nếu tài liệu có nhiều bảng (table of contents – bảng mục lục, table of figure – danh sách hình ảnh, ...)
- **Level**: định cấp bậc cho đề mục.

6.2.2 Chèn bảng mục lục

Bước 1: Đặt con trỏ tại vị trí muốn chèn bảng mục lục (sau khi đã đánh dấu các đề mục).

Bước 2: Vào menu **Insert \ Reference \ Index and Tables** để mở hộp thoại **Index and Tables**.

Bước 3: Trong thẻ **Table of Contents**, chọn nút **Option** để mở hộp thoại **Table of Contents Options**.



Bước 4: Mặc định trong hộp thoại **Table of Contents Options** chọn sẵn *checkbox Styles (Build table of content from styles)*, có nghĩa là bảng mục lục sẽ được tạo ra

dựa trên style. Đối với trường hợp tạo mục lục theo cách thủ công, ta phải chọn **checkbox Table entry fields**.

Bước 5: Click **OK** để trở về hộp thoại **Index and Tables**.

Bước 6: Click **OK** lần nữa, bảng mục lục dựa trên các entry do bạn đánh dấu bằng tay sẽ được tạo ra.

6.2.3 Cập nhật bảng mục lục

Một số phương pháp cập nhật lại bảng mục lục khi nội dung tài liệu có hiệu chỉnh:

- Nhấp phải vào chỗ bất kỳ trong bảng mục lục, chọn **Update Field**.
- Đặt con trỏ vào chỗ bất kỳ trong bảng mục lục, nhấn phím F9.
- Chọn khối và nhấn F9: chỉ cập nhật vùng được chọn, dùng khi không muốn ảnh hưởng đến các bảng khác trong tài liệu (nếu chèn nhiều bảng).
- Vào hộp thoại **Tools \ Options**, trong thẻ **Print** chọn **checkbox Update Field**: tài liệu sẽ được cập nhật toàn bộ khi được in.

Khi cập nhật bảng mục lục, hộp thoại **Update Table of Contents** xuất hiện:

- **Update page numbers only:** chỉ cập nhật số trang, nên chọn khi bạn thực hiện các thao tác định dạng trên bảng mục lục.
- **Update entire table:** cập nhật toàn bộ bảng mục lục, nên chọn khi bạn có thêm đề mục mới vào nội dung tài liệu.









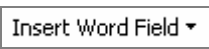











6.3 MAIL MERGE

Chức năng Mail Merge giúp chúng ta tạo nhanh một loạt tài liệu có cùng nội dung, có thể khác nhau ở vài thuộc tính. Chẳng hạn khi muốn tạo và in văn bản cùng một nội dung cho nhiều người khác nhau, các văn bản này chỉ khác nhau về các thuộc tính liên quan đến người nhận như họ tên, năm sinh, địa chỉ,... chức năng Mail Merge sẽ giúp ích cho bạn rất nhiều trong việc này.

6.3.1 Sử dụng thanh công cụ Mail Merge:

Bạn có thể bật hoặc tắt thanh công cụ Mail Merge bằng cách vào menu **View \ Toolbars \ Mail Merge**.

	Main Document Setup: chọn kiểu tài liệu hiện hành.
	Open Data Source: chọn dữ liệu nguồn.
	Mail Merge Recipients: hiển thị danh sách dữ liệu nguồn.

	Insert Address Block: chèn địa chỉ.
	Insert Greeting Line: chèn lời chào.
	Insert Merge Fields: chèn trường merge.
	Insert Word Field: chèn các trường của word.
	View Merged Data: xem kết quả hiển thị sau khi merge.
	Highlight Merge Fields: highlight các trường merge.
	Match Fields:
	Propagate Labels:
	First Previous Go to Next Last Record: di chuyển giữa các mẫu tin.
	Find Entry: tìm mẫu tin.
	Check for Errors: kiểm tra lỗi.
	Merge to New Document: chép ra tập tin mới.
	Merge to Printer: in ra máy in.
	Merge to E-mail: gửi email.
	Merge to Fax: gửi qua fax (nếu máy tính đã cài đặt máy fax).

6.3.2 Sử dụng chức năng Mail Merge:

Giả sử bạn cần soạn giấy giới thiệu cho nhiều sinh viên đến thực tập ở những nơi khác nhau, bạn sẽ chuẩn bị:

Tập tin chính (main document): chứa phần nội dung cố định. Chẳng hạn mẫu giấy giới thiệu của bạn có nội dung như sau:

TRƯỜNG ĐẠI HỌC AN GIANG KHOA KỸ THUẬT - CÔNG NGHỆ - MÔI TRƯỜNG *** Số: ... /GT	CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM Độc lập - Tự do - Hạnh phúc ----- Long Xuyên, ngày ... tháng ... năm ...
<h3>GIẤY GIỚI THIỆU</h3>	
Kính gửi:	
TRƯỜNG ĐẠI HỌC AN GIANG GIỚI THIỆU	
Sinh viên:	
Lớp:	
Đến:	
Để xin được thực tập cuối khóa theo chương trình đào tạo Cử nhân Tin học.	
Rất mong quý cơ quan giúp đỡ và tạo điều kiện cho sinh viên được khảo sát, tham khảo, thu thập các dữ liệu cần thiết.	
Kính chào và cảm ơn quý cơ quan.	
TRƯỜNG KHOA	

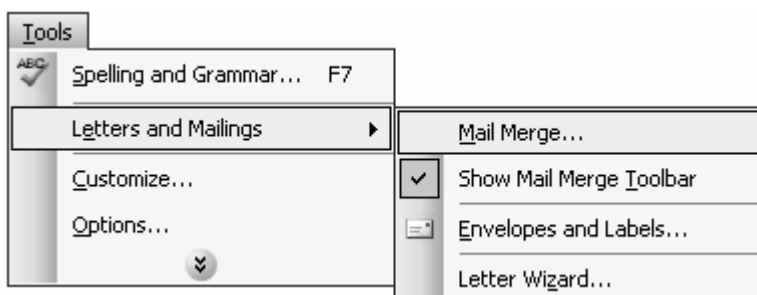
Tập tin dữ liệu (data source): chứa các thuộc tính động có dạng bảng, đó có thể là tập tin word, excel hoặc access. Chẳng hạn dữ liệu nguồn trong ví dụ này là bảng như sau:

Số	Họ tên SV	Lớp	Nơi nhận
51	Nguyễn Quốc Việt	ĐH4TH1	Sở Bưu chính Viễn thông AG
52	Võ Thị Phương Thảo	ĐH4TH1	Công ty TNHH Vĩnh Lợi
53	Huỳnh Thanh Tùng	ĐH4TH1	Công ty Xuất Nhập Khẩu AG

Nếu không tạo sẵn tập tin dữ liệu nguồn, bạn cũng có thể nhập dữ liệu trong khi sử dụng chức năng Mail Merge.

Cách thực hiện:

Trước tiên, bạn mở tập tin mới và soạn mẫu giấy giới thiệu như trên. Sau đó, vào menu **Tools \ Letters and Mailings \ Mail Merge** để mở **Mail Merge task pane**



Bước 1 Trên Mail Merge task pane, tùy mục đích của bạn mà chọn kiểu tài liệu thích hợp trong mục **Select document type**.

- **Letters:** soạn thư.
- **Email messages:** gửi mail.
- **Envelopes:** in địa chỉ ngoài bì thư.
- **Labels:** in nhãn địa chỉ
- **Directory:** tạo một tài liệu đơn chứa thông tin nhiều trường hoặc danh sách địa chỉ

Trong ví dụ này, chúng ta sẽ chọn mục Letters.

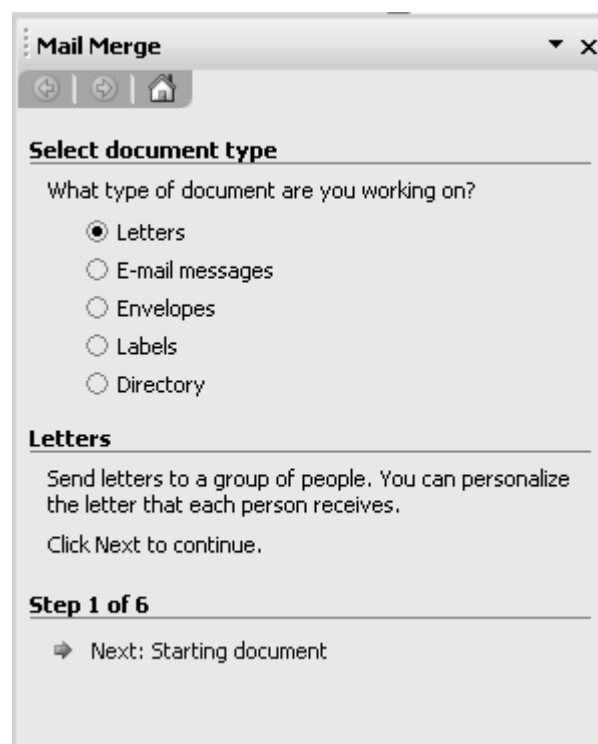
Chọn **Next: Starting document**.

Bước 2 Xác định tài liệu chính trong mục **Select Starting document**.

- **Use the current document:** sử dụng tài liệu hiện hành.
- **Start from a template:** tạo mới từ một dạng template.
- **Start from existing document:** chọn mở tài liệu có sẵn.

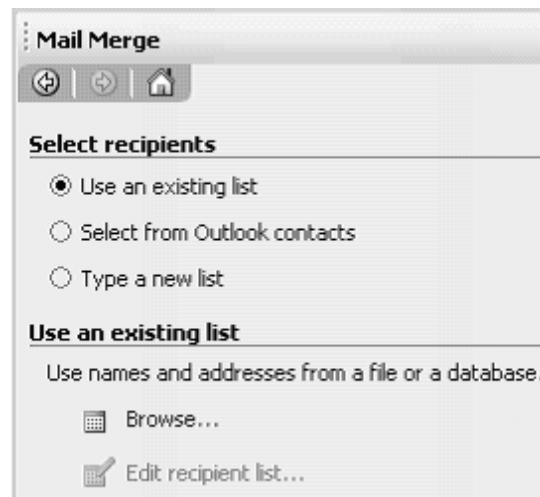
Đối với ví dụ này, chúng ta sử dụng tài liệu hiện hành nên sẽ chọn **Use the current document**.

Chọn **Next: Select Recipients**.



Bước 3 Chọn một trong các tùy chọn của mục **Select Recipients** để xác định dữ liệu nguồn.

- **Use an existing list:** sử dụng tập tin dữ liệu có sẵn.
- **Select from Outlook contacts:** lấy từ sổ địa chỉ của trình quản lý thư Outlook.
- **Type a new list:** tạo danh sách mới.

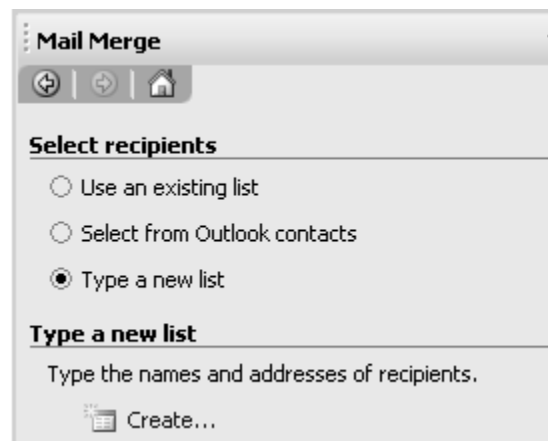


★ Nếu chúng ta đã tạo sẵn tập tin dữ liệu, ta sẽ chọn Use an exiting list và Browse để mở tập tin đó.

★ Nếu chưa có tập tin dữ liệu nguồn, ta cũng có thể chọn Type a new list và Create để tạo mới.

Trong hộp thoại New Address List có cung cấp sẵn một số trường thông dụng như First Name, Last Name,...

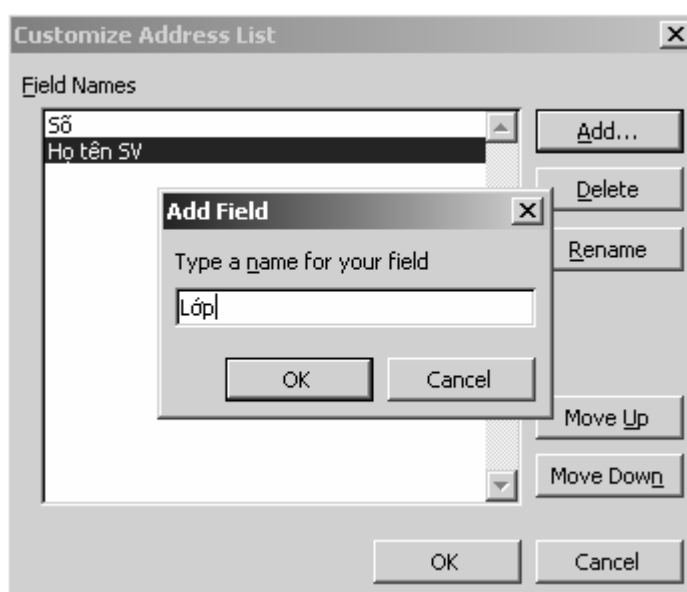
- Để thêm mẫu tin mới sử dụng nút New Entry.
- Để xóa mẫu tin hiện hành sử dụng nút Delete Entry.



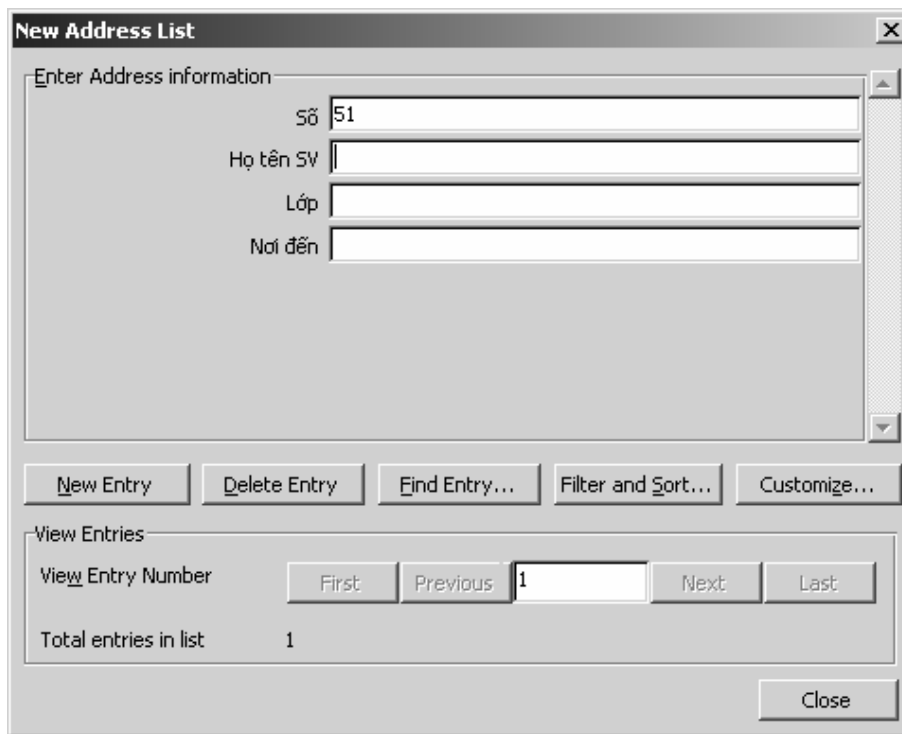
- Để tìm một mẫu tin sử dụng nút Find Entry.
- Để lọc và sắp xếp các mẫu tin sử dụng nút Filter and Sort.
- Để tùy biến các trường theo nhu cầu, sử dụng nút Customize. Trong hộp thoại Customize Address List, dùng các nút Add, Delete, ... để tạo các trường mong muốn.

Ví dụ: chúng ta sẽ tùy biến các trường để nhập liệu cho ví dụ trên. Click nút Customize trong hộp thoại New Address List, hộp thoại Customize Address List sẽ mở ra, sử dụng các nút lệnh trên hộp thoại này để được các tên trường như mong muốn (các tên trường này chính là dòng tiêu đề đầu tiên của bảng).

- Add: thêm trường mới, gõ tên trường vào hộp thoại Add Field, OK.
- Delete: xóa trường đang chọn.
- Rename: đổi tên trường đang chọn.
- Move up: chuyển trường đang được chọn lên trên một vị trí.
- Move down: chuyển trường đang được chọn xuống dưới một vị trí.



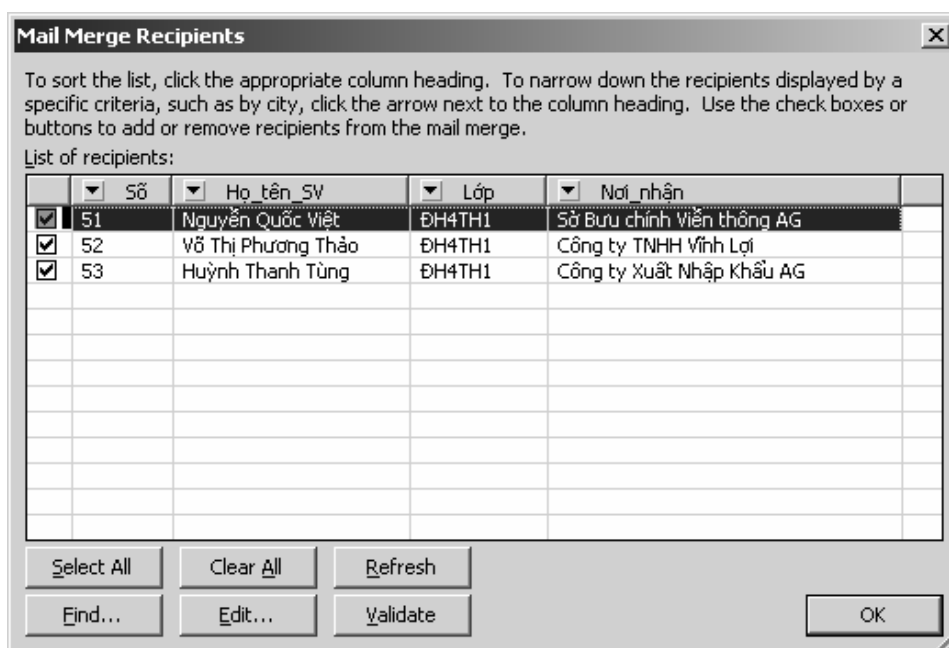
Sau khi đã tùy biến tên trường như mong muốn, chúng ta sẽ tiến hành nhập liệu.



Nhập liệu xong nhấn Close, word sẽ yêu cầu bạn lưu dữ liệu này dưới dạng Office Address List (là một tập tin cơ sở dữ liệu access).

★ Bạn cũng có thể lấy dữ liệu từ sổ địa chỉ của trình quản lý email Outlook (trong trường hợp sử dụng Mail Merge cho mục đích gửi email).

Sau khi xác định dữ liệu nguồn, bảng dữ liệu Mail Merge Recipients hiển thị, bạn có thể chọn hoặc bỏ các mẫu tin bằng cách chọn checkbox tương ứng.



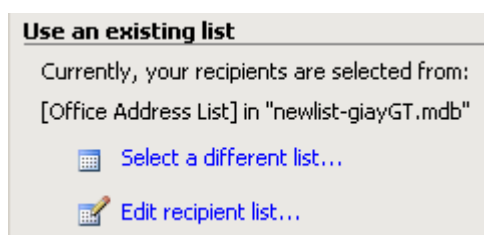
Chức năng các nút lệnh trong hộp thoại **Mail Merge Recipients**

- **Select All:** chọn tất cả các mẫu tin.
- **Clear All:** hủy chọn tất cả các mẫu tin.
- **Refresh:** cập nhật mới danh sách sau khi có hiệu chỉnh.
- **Find:** tìm mẫu tin.
- **Edit:** cập nhật mẫu tin.
- **Validate:** kiểm tra địa chỉ, zip code có hợp lệ không.

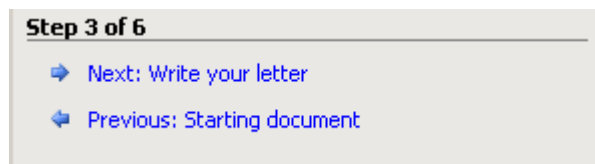
Nhấn OK khi hoàn tất.

Quan sát trên task pane bạn sẽ thấy dưới tên tập tin dữ liệu hiện hành là các tùy chọn:

- **Select a different list:** chọn tập tin dữ liệu khác.
- **Edit recipient list:** hiệu chỉnh bảng dữ liệu hiện hành.



Hoàn tất bước 3, chọn **Next:**
Write your letter.



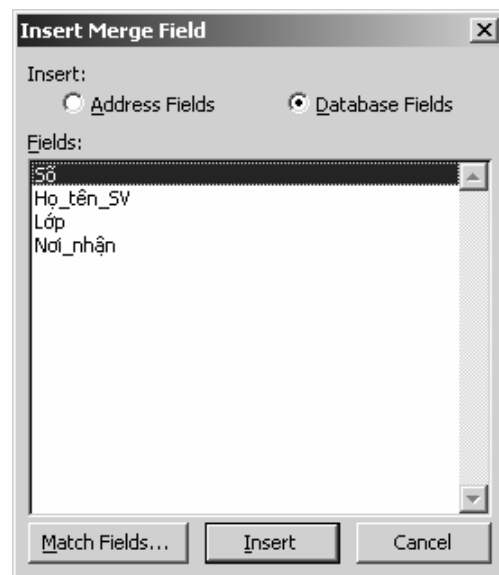
Bước 4 Bạn có thể chỉnh sửa nội dung tập tin chính, thêm các trường vào tài liệu hiện hành:

- **Address block:** địa chỉ người nhận.
- **Greeting line:** lời chào.
- **Postal bar code:** mã vạch.
- **More items:** lấy từ dữ liệu nguồn.

Để chèn một trường vào tài liệu chính, đặt con trỏ tại điểm chèn rồi chọn mục tương ứng trên task pane (hoặc các nút lệnh tương ứng trên thanh công cụ Mail Merge). Các trường được chèn vào sẽ được đặt trong cặp dấu ngoặc, có dạng tương tự như <<Merge Field>>.

Ví dụ: tiếp tục ví dụ trên, ta đặt con trỏ tại vị trí cần chèn số hiệu giấy giới thiệu và click More items. Hộp thoại Insert Merge Field sẽ mở ra, ta chọn trường tương ứng là “Số”, click nút Insert.

Tương tự cho các trường còn lại.



Sau khi chèn các trường cần thiết, tài liệu sẽ trông có dạng như sau:

Số: 52/GT Long Xuyên, ngày 01 tháng 01 năm 2007

GIẤY GIỚI THIỆU

Kính gửi: Công ty TNHH Vĩnh Lợi

TRƯỜNG ĐẠI HỌC AN GIANG GIỚI THIỆU

Sinh viên: Võ Thị Phương Thảo
 Lớp: ĐH4TH1
 Đến: Công ty TNHH Vĩnh Lợi

Sau đó nhấn **Next: Review your letter.**

Bước 5 Bạn có thể xem lại tài liệu sau khi merge:

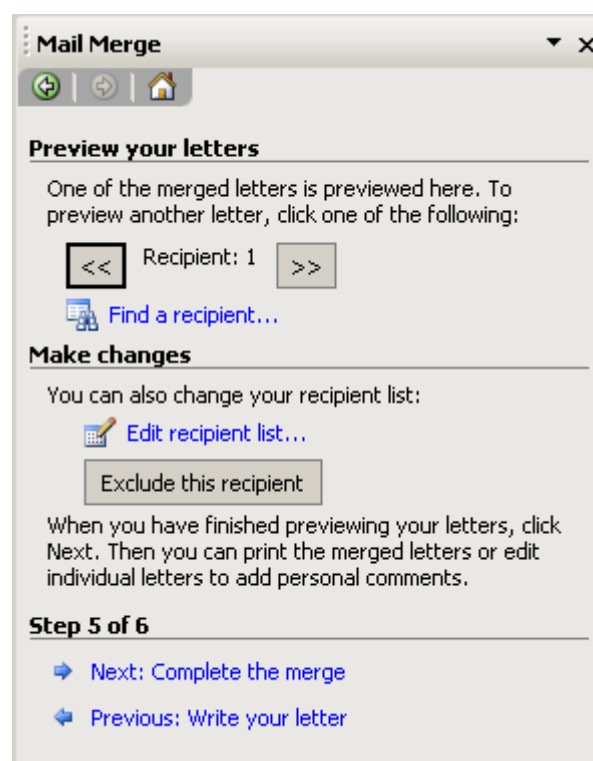
- Dùng các nút << và >> để di chuyển xem tài liệu theo từng mẫu tin,
- Tìm theo mẫu tin bằng cách chọn Find a recipient.

Tại bước này, bạn cũng có thể hiệu chỉnh dữ liệu bằng cách chọn Edit recipient list, hoặc loại bỏ nội dung thư merge của một mẫu tin cụ thể bằng nút Exclude this recipient.

Khi đã vừa ý, chọn **Next: Complete the merge.**

Bước 6 Đến bước này xem như đã hoàn tất.

- Chọn Print để in tài liệu đã merge ra máy in.
- Chọn Edit individual letters để hiệu chỉnh phần nội dung đã merge theo từng mẫu tin cụ thể.



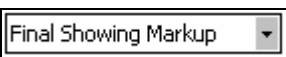









6.4 TRACK CHANGE

Trong một số trường hợp, bạn muốn lưu lại những gì bạn vừa sửa trên văn bản một cách rõ ràng, hoặc phân phối tài liệu mà vẫn muốn kiểm soát những thay đổi của nội dung so với bản gốc, ... thì tiện ích Track Change của Word sẽ giúp ích rất nhiều. Khi bạn yêu cầu Track Change, Word sẽ đánh dấu mọi sự thay đổi nội dung trên văn bản để bạn theo dõi.

6.4.1 Thanh công cụ Reviewing


Thanh công cụ Reviewing giúp bạn duyệt tài liệu có sử dụng chức năng track change. Để bật / tắt thanh công cụ **Reviewing**, bạn có thể vào menu **View \ Toolbars \ Reviewing**.

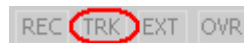


	<p>Hộp thả Display for Review cho phép chọn lựa giữa 4 chế độ xem:</p> <ul style="list-style-type: none"> - Final Showing Markup: hiển thị văn bản sau khi hiệu chỉnh cùng các nội dung được đánh dấu. - Final: hiển thị văn bản sau khi hiệu chỉnh. - Original Showing Markup: hiển thị văn bản ban đầu cùng các nội dung được đánh dấu. - Original: hiển thị văn bản ban đầu.
	<p>Chọn hiển thị các biểu tượng khi duyệt tài liệu</p>
	<p>Xem phần hiệu chỉnh trước đó.</p>
	<p>Xem phần hiệu chỉnh kế tiếp.</p>
	<p>Chấp nhận phần hiệu chỉnh được chọn.</p>
	<p>Từ chối phần hiệu chỉnh, xóa lời bình.</p>
	<p>Chèn lời bình.</p>
	<p>Tô highlight cho nội dung được chọn.</p>
	<p>Bật / tắt chức năng track change.</p>
	<p>Mở cửa sổ Reviewing.</p>

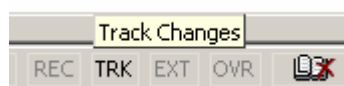
6.4.2 Sử dụng chức năng Track Change

➤ Trước hết, bạn phải kích hoạt chức năng Track Changes bằng một trong các cách sau:

- Nhấn tổ hợp phím Ctrl + Shift + E.
- Vào menu Tools \ Track Changes.
- Double-click biểu tượng TRK trên thanh trạng thái
- Nhấn nút  trên thanh công cụ Reviewing.



Sau khi được kích hoạt, biểu tượng TRK trên thanh trạng thái sẽ sáng:



➤ Khi tài liệu được hiệu chỉnh thì các quá trình thay đổi chỉnh sửa văn bản nguồn sẽ được đánh dấu lại. Tài liệu được xem ở các chế độ khác nhau như Normal View, Print Layout View,... sẽ được hiển thị hơi khác nhau.

Ví dụ:

Bạn đang có một tài liệu với nội dung ban đầu là:

Tôi đang học Track Change |

Hãy bật chức năng Track Change, sau đó hiệu chỉnh một vài nội dung:

- Xóa chữ “học”:

Ở dạng Print Layout View, tài liệu trông có dạng như sau:

| Tôi đang |Track Change. Deleted: học

Ở dạng Normal View, tài liệu trông có dạng như sau:

| Tôi đang học Track Change.

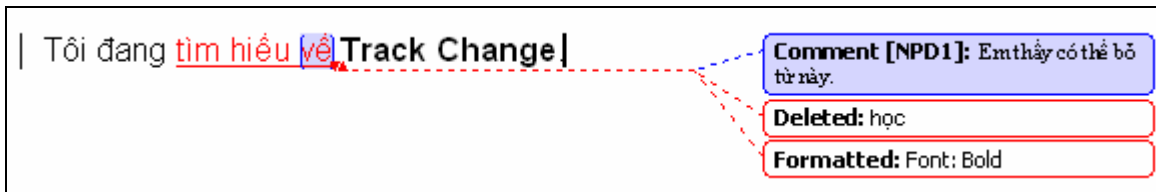
- Chèn thêm một số từ mới:

| Tôi đang tìm hiểu về Track Change. Deleted: học

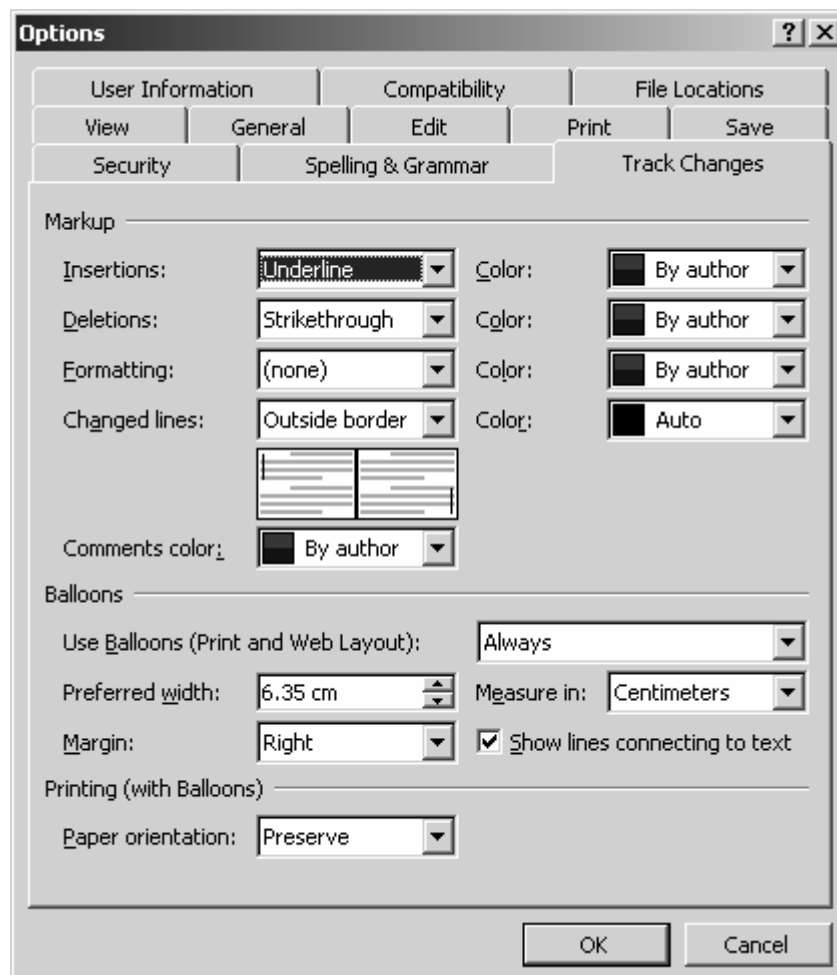
- Định dạng in đậm từ “Track Change”

| Tôi đang tìm hiểu về Track Change. Deleted: học Formatted: Font: Bold

- Nếu không muốn hiệu chỉnh trực tiếp lên tài liệu, bạn có thể để lại ghi chú hay lời bình bằng chức năng **Insert \ Comment** (hoặc nút Insert Comment trên thanh công cụ Reviewing)



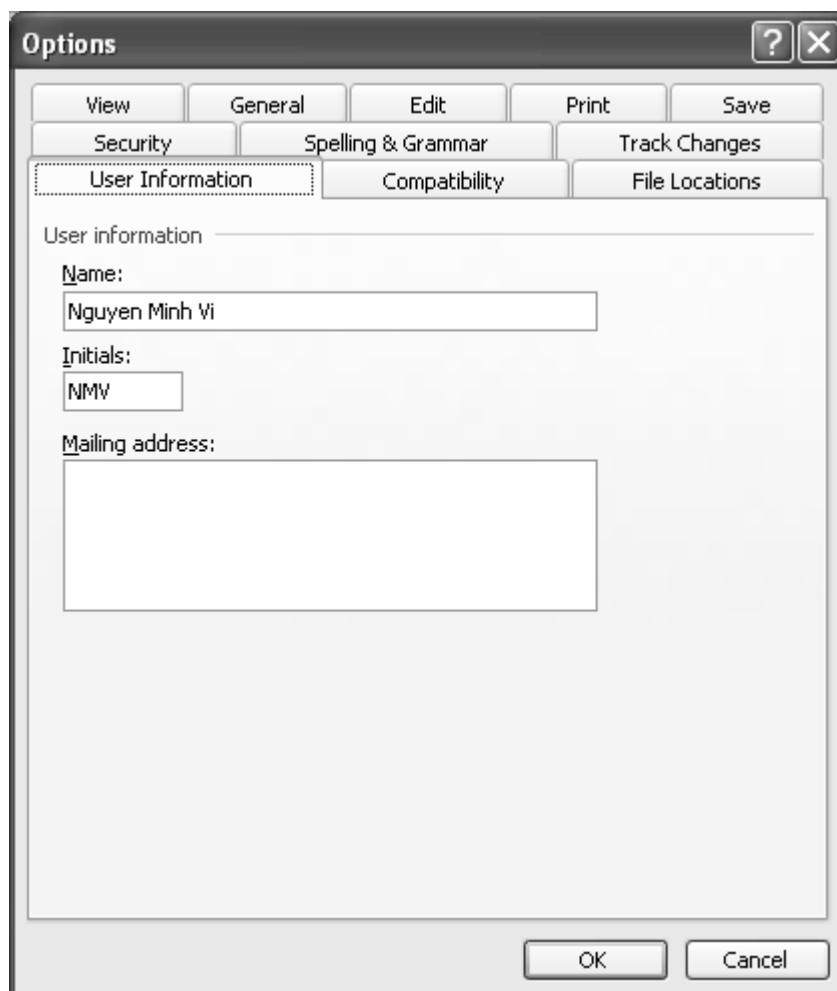
- Bạn có thể thiết lập lại các tùy chọn cho Track Change nếu không thích cách hiển thị mặc định của Word. Dùng một trong các cách sau để định lại tùy chọn cho Track Change:
 - Vào menu **Tools \ Options**, chọn thẻ Track Changes.
 - Trên thanh công cụ Reviewing, chọn nút Show → Options.



Mục Markup cho phép thiết lập lại cách đánh dấu các thao tác hiệu chỉnh như **Insertions** (chèn), **Deletions** (xóa), **Formatting** (định dạng), **Changed lines** (đánh dấu dòng), **comment** (chèn lời bình).

Bạn có thể quy định màu cho từng loại thao tác hiệu chỉnh trong hộp thả Color tương ứng. Tuy nhiên, nếu có nhiều người chỉnh sửa trên cùng tài liệu thì nên chọn By Author, trường hợp này Word sẽ quy định màu khác nhau cho mỗi người.

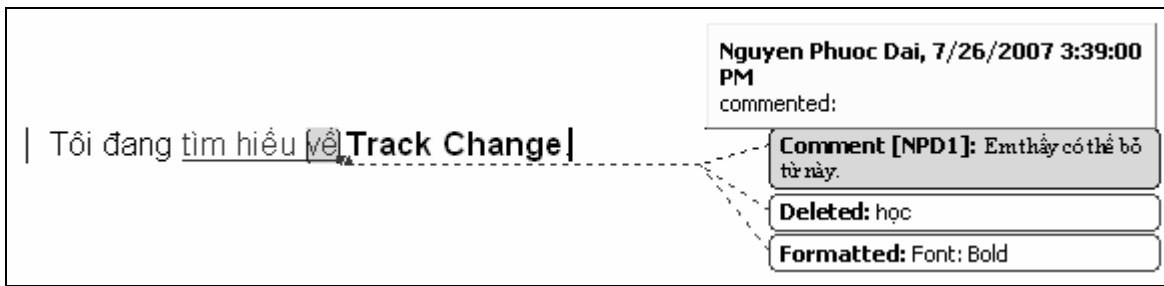
- Để Word phân biệt được giữa các người dùng, bạn phải xác nhận đúng thông tin về mình trong thẻ User Information của hộp thoại **Options (Tools \ Options)** trước khi bắt đầu việc hiệu chỉnh.








Khi xem lại tài liệu, bạn muốn biết ai đã thay đổi phần nội dung nào thì chỉ cần mở cửa sổ Reviewing.

Main document changes and comments		
Inserted tìm hiểu về	Nguyen Minh Vi	7/26/2007 3:33:00 PM
Comment [NPD1] Em thấy có thể bỏ từ này.	Nguyen Phuoc Dai	7/26/2007 3:39:00 PM
Deleted học	Nguyen Minh Vi	7/26/2007 3:15:00 PM
Formatted	Nguyen Minh Vi	7/26/2007 3:36:00 PM

Hoặc chỉ cần lướt chuột trên nội dung đó (hoặc vào balloon tương ứng), thông tin về người dùng sẽ được hiển thị.



6.4.3 Duyệt tài liệu với thanh công cụ Reviewing:

- Dùng các nút  và  để di chuyển giữa các phần nội dung hiệu chỉnh, lời bình.
- Dùng nút  để chấp nhận phần hiệu chỉnh đang duyệt.
- Dùng nút  để từ chối phần hiệu chỉnh đang duyệt, đồng thời xóa luôn lời bình trên phần nội dung đó (nếu có).
- Dùng nút  mở cửa sổ Reviewing khi duyệt tài liệu. Trong cửa sổ này, bạn có thể click chuột chọn phần hiệu chỉnh để chấp nhận hoặc từ chối một cách dễ dàng.
- Nếu có quá nhiều nội dung hiệu chỉnh làm rối mắt, bạn có thể tắt bớt các balloon bằng cách vào nút **Show → Balloon** và chọn tùy chọn hiển thị balloon tương ứng: **Always** (luôn hiển thị), **Never** (ẩn hết tất cả), **Only for Comment/Formatting** (chỉ hiển thị cho lời bình, định dạng)
- Nếu có nhiều người thao tác trên tài liệu, bạn có thể cho hiển thị hoặc ẩn nội dung hiệu chỉnh của từng người cụ thể bằng cách check chọn trong nút **Show → Reviewers**.
- Nếu tài liệu gốc được phân phối cho nhiều người hiệu chỉnh, bạn muốn ghép các tài liệu này lại thành một để duyệt thì vào menu **Tools \ Compare and Merge Documents**, word cung cấp các tùy chọn ghép tài liệu sau:
 - Merge
 - Merge into current document
 - Merge into new document

Tin học đại cương

Nguyễn Hồng Phương

Email: phuongnh-fit@mail.hut.edu.vn

Website: <http://is.hut.edu.vn/~phuongnh>

Bộ môn Hệ thống thông tin

Viện Công nghệ thông tin và Truyền thông

Đại học Bách Khoa Hà Nội

1

Thông tin chung

- Giảng viên:
 - Nguyễn Hồng Phương
 - Bộ môn Hệ thống thông tin, Viện CNTT&TT, phòng 603, 702-B1.
 - Email: phuongnh-fit@mail.hut.edu.vn
phuongnh@gmail.com
- Giờ tiếp sinh viên tại Bộ môn:
 - Sáng thứ hai hàng tuần
 - Ngoài ra, xin liên hệ trước.

2

Tổng quan về môn học

- Mục tiêu: Sau khi học xong môn học này, sinh viên nắm được:
 - có kiến thức về tin học căn bản
 - có kiến thức về giải quyết bài toán
 - có thể lập trình bằng ngôn ngữ C
- Khối lượng:
 - 4 tiết/ tuần, học trong 15 tuần trên giảng đường
 - Bài tập và Thực hành (theo quy định chung).

3

Đánh giá môn học

- Dự lớp đầy đủ, tích cực xây dựng bài
- Kiểm tra giữa kỳ
- Thực hành
- Kiểm tra cuối kỳ (theo lịch chung)



4

Tài liệu học tập



- Các tác giả thuộc Viện CNTT&TT-ĐHBK Hà Nội, Giáo trình *Tin học đại cương*, NXB Bách Khoa - Hà Nội, 2011.
- **Bài giảng trên lớp.**
- TS. Quách Tuấn Ngọc, *Giáo trình Tin học căn bản*, NXB Thống kê, 2001.
- GS. Hoàng Kiếm, *Tin học đại cương nâng cao*, NXB Giáo dục, 1998.
- GS. Phạm Văn Ất, *Kỹ thuật lập trình C: cơ sở và nâng cao*, NXB Khoa học và kỹ thuật, 1999.

5

Nội dung môn học



- Phần I: TIN HỌC CĂN BẢN
 - Chương 1: Mở đầu
 - Thông tin và xử lý thông tin
 - Máy tính và phân loại
 - Tin học
 - Chương 2: Biểu diễn dữ liệu trong máy tính
 - Hệ đếm
 - Mã hóa dữ liệu trong máy tính và đơn vị thông tin
 - Biểu diễn số nguyên
 - Thực hiện phép toán số học với số nguyên
 - Các phép toán logic với số nhị phân
 - Biểu diễn ký tự
 - Biểu diễn số thực

6

Nội dung môn học (2)



- Chương 3: Hệ thống máy tính
 - Tổ chức bên trong máy tính
 - Phần mềm máy tính
- Chương 4: Mạng máy tính
 - Lịch sử, khái niệm
 - Phân loại
 - Các thành phần cơ bản
 - Mạng Internet

7

Nội dung môn học (3)



- Chương 5: Hệ điều hành
 - Giới thiệu chung
 - Các hệ điều hành: Window, Linux,...
- Chương 6: Các hệ thống ứng dụng
 - Hệ thống thông tin quản lý
 - Hệ thông tin bảng tính
 - Hệ quản trị cơ sở dữ liệu
 - Các hệ thống thống minh

8

Nội dung môn học (4)



- Phần II: GIẢI QUYẾT BÀI TOÁN
 - Chương 1: Giải quyết bài toán
 - Khái niệm
 - Các bước giải quyết bài toán bằng máy tính
 - Các phương pháp giải quyết bài toán bằng máy tính
 - Phân loại bài toán
 - Chương 2: Thuật toán
 - Định nghĩa
 - Biểu diễn
 - Một số thuật toán thông dụng
 - Thuật toán đệ quy
 - Thuật giải heuristic

9

Nội dung môn học (5)



- Phần III: LẬP TRÌNH BẰNG NGÔN NGỮ C
 - Chương 1: Tổng quan về ngôn ngữ C
 - Lịch sử
 - Các phần tử cơ bản
 - Cấu trúc cơ bản
 - Chương trình biên dịch
 - Chương 2: Kiểu dữ liệu, biểu thức và cấu trúc lập trình trong C
 - Các kiểu dữ liệu chuẩn
 - Biểu thức
 - Các cấu trúc lập trình

10

Nội dung môn học (6)



- Chương 3: Con trỏ, mảng và chuỗi trong C
 - Con trỏ và địa chỉ
 - Mảng
 - Chuỗi ký tự
- Chương 4: Hàm
 - Khái niệm hàm
 - Khai báo và sử dụng hàm

11

Nội dung môn học (7)



- Chương 5: Cấu trúc
 - Khái niệm cấu trúc
 - Khai báo và sử dụng cấu trúc
 - Mảng cấu trúc
- Chương 6: Tập dữ liệu (bộ)
 - Khái niệm và phân loại tệp
 - Tệp văn bản
 - Tệp binary

12



Lời hay ý đẹp

“Trời có bốn mùa: Xuân, Hạ, Thu, Đông;
Đất có bốn phương: Đông, Tây, Nam, Bắc;
Người có bốn đức: **Cần, Kiệm, Liêm, Chính**;
Thiếu một mùa không thể thành Trời;
Thiếu một phương không thể thành Đất;
Thiếu một đức không thể thành **Người**.”

Hồ Chí Minh

14

PHẦN I: TIN HỌC CĂN BẢN

Chương 1: Mở đầu

Nguyễn Hồng Phương
Email: phuongnh-fit@mail.hut.edu.vn
Website: <http://is.hut.edu.vn/~phuongnh>
Bộ môn Hệ thống thông tin
Viện Công nghệ thông tin và Truyền thông
Đại học Bách Khoa Hà Nội

1

Nội dung chương 1

- 1.1. Thông tin và xử lý thông tin
- 1.2. Máy tính và phân loại máy tính
- 1.3. Tin học

2

1.1. Thông tin và xử lý thông tin

- Thông tin, dữ liệu, tri thức
 - Thông tin (Information): mang lại cho con người sự hiểu biết, nhận thức tốt hơn về những đối tượng trong tự nhiên - xã hội
 - Dữ liệu (Data): biểu diễn của thông tin được thể hiện bằng các tín hiệu vật lý. Dữ liệu trong thực tế có thể là:
 - các số liệu trong các biểu đồ
 - các ký hiệu quy ước như chữ viết
 - các tín hiệu vật lý: ánh sáng, âm thanh, nhiệt độ, áp suất,...

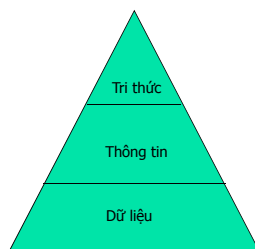
3

1.1. Thông tin và xử lý thông tin (2)

- Tri thức (Knowledge):
 - thông tin ở mức trừu tượng hơn
 - khá đa dạng
 - sự hiểu biết chung hay về một lĩnh vực cụ thể nào đó.
- Hệ thống thông tin (information system)
 - Dữ liệu $\xrightarrow{\text{xử lý}}$ Thông tin $\xrightarrow{\text{xử lý}}$ Tri thức

4

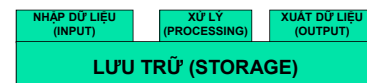
1.1. Thông tin và xử lý thông tin (3)



5

1.1. Thông tin và xử lý thông tin (4)

- Quy trình xử lý thông tin:
 - Mọi quá trình xử lý thông tin bằng máy tính hay bởi con người đều được thực hiện theo một qui trình sau:



6

1.2. Máy tính và phân loại máy tính điện tử

- Lịch sử phát triển của máy tính điện tử
 - Công cụ tính toán ngày xưa: bàn tính bằng tay
 - Máy cộng cơ học của nhà toán học Pháp Blaise Pascal (1623-1662)
 - Máy tính cơ học cộng trừ nhân chia của nhà toán học Đức Leibniz (1646-1716)
 - Máy tính điện tử thực sự bắt đầu vào những năm 1950, đến nay đã trải qua 5 thế hệ dựa vào sự tiến bộ về công nghệ điện tử và vi điện tử.

7

Lịch sử phát triển máy tính(2)

- Thế hệ 1 (1950-1958): Von Neumann Machine
 - Sử dụng các bóng đèn điện tử chân không
 - Mạch riêng rẽ, vào số liệu bằng phiếu đục lỗ
 - Điều khiển bằng tay, kích thước rất lớn
 - Tiêu thụ năng lượng nhiều, tốc độ tính chậm khoảng 300 - 3.000 phép tính/s.

8

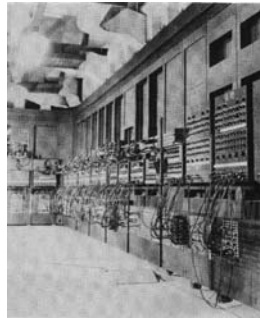
Lịch sử phát triển máy tính(3)



Bóng đèn chân không

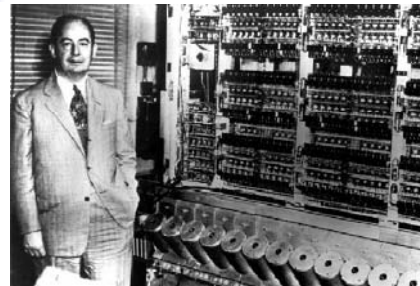
Máy tính đầu tiên:

ENIAC (Electronic Numerical Integrator And Computer)



9

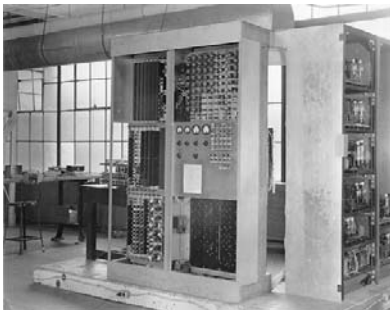
Lịch sử phát triển máy tính(4)



Von Neumann với máy tính Institute đầu tiên năm 1952

10

EDVAC (Mỹ)



11

Lịch sử phát triển máy tính (tiếp)

- Thế hệ 2 (1958 - 1964): Transistors
 - **Sử dụng bộ xử lý bằng đèn bán dẫn, mạch in**
 - Đã có chương trình dịch như Cobol, Fortran và hệ điều hành đơn giản.
 - Kích thước máy còn lớn
 - Tốc độ tính khoảng 10.000 - 100.000 phép tính/giây
 - Điển hình:
 - IBM 7000 series (Mỹ)
 - MINSK (Liên Xô cũ)

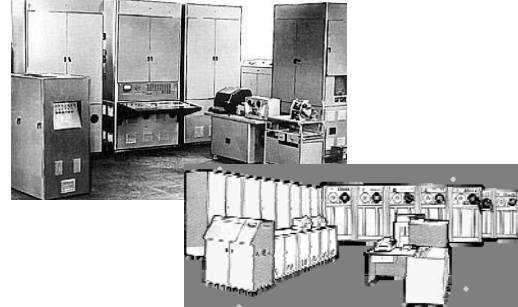
12

Thế hệ 2: IBM 7030 (1961)



13

Thế hệ 2: MINSK (Liên Xô cũ)



14

Lịch sử phát triển máy tính (tiếp)

Thế hệ 3 (1965 - 1974): Integrated Circuits

- Các bộ vi xử lý được gắn vi mạch điện tử cỡ nhỏ
- Tốc độ tính khoảng 100.000 - 1 triệu phép tính/giây.
- Có các hệ điều hành đa chương trình, nhiều người đồng thời hoặc theo kiểu phân chia thời gian.
- Kết quả từ máy tính có thể in trực tiếp từ máy in.
- Điển hình:
 - IBM-360 (Mỹ)
 - DEC PDP-8

15

Thế hệ 3: IBM 360 (Mỹ)



16

Thế hệ 3: DEC PDP-1(1960)



17

Lịch sử phát triển máy tính (tiếp)

Thế hệ 4 (1974 - 1990): LSI (Large Scale Integration), Multiprocessors

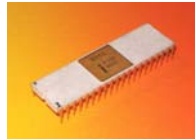
- Có các vi mạch đa xử lý
- Tốc độ tính hàng chục triệu đến hàng tỷ phép tính/giây.
- 2 loại máy tính chính:
 - Máy tính cá nhân để bàn (Personal Computer - PC) hoặc xách tay (Laptop hoặc Notebook computer)
 - Các loại máy tính chuyên nghiệp thực hiện đa chương trình, đa xử lý,...
- Hình thành các hệ thống mạng máy tính (Computer Networks).
- Các ứng dụng phong phú đa phương tiện

18

Thế hệ 4



INTEL 8080



INTEL 4004

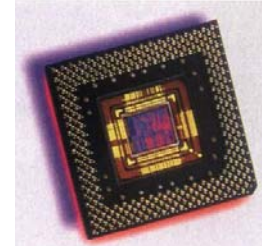
19

Thế hệ 4

INTEL 80386



Pentium



20

Thế hệ 4



Itanium

64-bit Intel
Microprocessors

21

Lịch sử phát triển máy tính

- Thế hệ 5 (1990 - nay): VLSI (Very Large Scale Integration), ULSI (Ultra), Artificial Intelligence (AI)
 - Công nghệ vi điện tử với tốc độ tính toán cao và khả năng xử lý song song.
 - Mô phỏng các hoạt động của não bộ và hành vi con người
 - Có trí khôn nhân tạo với khả năng tự suy diễn phát triển các tình huống nhận được
 - Hệ quản lý kiến thức cơ bản để giải quyết các bài toán đa dạng.

22

Phân loại máy tính

- Máy Vi tính (Microcomputer)
 - Được thiết kế cho một người dùng
 - Giá thành rẻ.
 - Được sử dụng phổ biến: máy để bàn (Desktop), máy trạm (Workstation), máy xách tay (Notebook),...
- Máy tính tầm trung (Mini Computer)
 - Tốc độ và hiệu năng tính toán mạnh hơn
 - Được thiết kế cho các ứng dụng phức tạp.
 - Giá ~ hàng vài chục nghìn USD
- Máy tính lớn (Mainframe Computer) và Siêu máy tính (Super Computer).

23

Phân loại máy tính (tiếp)

- Máy tính lớn và siêu máy tính (tiếp)
 - Phức tạp, có tốc độ siêu nhanh
 - Hiệu năng tính toán cao, cỡ hàng tỷ phép tính/giây
 - Nhiều người dùng đồng thời
 - Được sử dụng tại các Trung tâm tính toán/ Viện nghiên cứu để giải quyết các bài toán cực kỳ phức tạp, yêu cầu cao về tốc độ.
 - Giá thành rất đắt ~ hàng trăm ngàn, thậm chí hàng triệu USD

24

Phân loại máy tính hiện đại

- Máy tính để bàn (Desktop Computers)
- Máy chủ (Server)
- Máy tính nhúng (Embedded Computers)

25

Máy tính để bàn (Desktop)

- Là loại máy tính phổ biến nhất
- Các loại máy tính để bàn:
 - Máy tính cá nhân (Personal Computers - PC)
 - Máy tính trạm làm việc (Workstations)
- 1981: IBM giới thiệu máy tính IBM-PC sử dụng bộ xử lý Intel 8088
- 1984: Apple đưa ra Macintosh sử dụng bộ xử lý Motorola 68000
- Giá thành: 500 USD đến 10,000 USD

26

Máy chủ (Server)

- Thực chất là máy phục vụ
- Dùng trong mạng theo mô hình Client/Server
- Tốc độ và hiệu năng tính toán cao
- Dung lượng bộ nhớ lớn
- Độ tin cậy cao
- Giá thành: hàng chục nghìn đến hàng chục triệu USD

27

Máy tính nhúng (Embedded Computers)

- Được đặt trong thiết bị khác để điều khiển thiết bị đó làm việc
- Được thiết kế chuyên dụng
- Ví dụ:
 - Điện thoại di động
 - Máy ảnh số
 - Bộ điều khiển trong máy giặt, điều hòa
 - Router - bộ định tuyến trên mạng
- Giá thành: vài USD đến hàng trăm nghìn USD

28

1.3. Tin học

- Tin học (Informatics)
 - Ngành khoa học nghiên cứu các phương pháp, công nghệ và kỹ thuật xử lý thông tin một cách tự động.
 - Công cụ: Máy tính điện tử và các thiết bị truyền tin.
 - Nội dung nghiên cứu:
 - **Kỹ thuật phần cứng** (Hardware engineering)
 - Thiết bị, linh kiện điện tử, công nghệ vật liệu mới... hỗ trợ cho máy tính và mạng máy tính, đẩy mạnh khả năng xử lý toán học và truyền thông thông tin.
 - **Kỹ thuật phần mềm** (Software engineering)
 - Các hệ điều hành, ngôn ngữ lập trình cho các bài toán khoa học kỹ thuật, mô phỏng, điều khiển tự động, tổ chức dữ liệu và quản lý hệ thống thông tin

29

1.3. Tin học (tiếp)

- Công nghệ thông tin (Information Technology – IT)
 - Ngành nghiên cứu các hệ thống thông tin dựa vào máy tính, đặc biệt là các phần mềm ứng dụng và phần cứng máy tính.
 - IT xử lý với các máy tính điện tử và các phần mềm máy tính nhằm chuyển đổi, lưu trữ, bảo vệ, truyền tin và trích rút thông tin một cách an toàn.
- Các ứng dụng ngày nay của IT
 - *Các bài toán khoa học kỹ thuật*
 - *Các bài toán quản lý*
 - *Tự động hóa*
 - *Công tác văn phòng*
 - *Tin học và giáo dục*
 - *Thương mại điện tử*
 - *Công nghệ thông tin với cuộc sống đời thường*

30

1.3. Tin học (tiếp)

- Công nghệ thông tin và truyền thông: Information and Communication Technology (ICT).
 - Kết nối một số lượng máy tính với nhau
 - **Internet** - *Mạng máy tính toàn cầu*

31



32

Lời hay ý đẹp

Trong 10 lần thành công thì có tới 9 lần thành công nhờ sự hăng hái và niềm tin trong công việc

Teewilson

33

Chương 2: Biểu diễn dữ liệu trong máy tính

Nguyễn Hồng Phương
Email: phuongnh-fit@mail.hut.edu.vn
Website: <http://is.hut.edu.vn/~phuongnh>
Bộ môn Hệ thống thông tin
Viện Công nghệ thông tin và Truyền thông
Đại học Bách Khoa Hà Nội

1

Nội dung chương này

- 2.1. Biểu diễn số trong các hệ đếm
 - 2.1.1. Hệ đếm cơ số b
 - 2.1.2. Hệ đếm thập phân
 - 2.1.3. Hệ đếm nhị phân
 - 2.1.4. Hệ đếm bát phân
 - 2.1.5. Hệ đếm thập lục phân
 - 2.1.6. Chuyển đổi một số từ hệ thập phân sang hệ đếm cơ số b
 - 2.1.7. Mệnh đề logic

2

Nội dung chương này (tiếp)

- 2.2. Biểu diễn dữ liệu trong máy tính và đơn vị thông tin
 - 2.2.1. Nguyên tắc chung
 - 2.2.2. Đơn vị thông tin
- 2.3. Biểu diễn số nguyên
 - 2.3.1. Số nguyên không dấu
 - 2.3.2. Số nguyên có dấu
- 2.4. Phép toán số học với số nguyên
 - Cộng/trừ
 - Nhân/chia

3

Nội dung chương này (tiếp)

- 2.5. Tính toán logic với số nhị phân
- 2.6. Biểu diễn ký tự
 - 2.6.1. Nguyên tắc chung
 - 2.6.2. Bộ mã ASCII
 - 2.6.3. Bộ mã Unicode
- 2.7. Biểu diễn số thực
 - 2.7.1. Nguyên tắc chung
 - 2.7.2. Chuẩn IEEE754/85

4

2.1. Biểu diễn số trong các hệ đếm

- Hệ đếm là tập hợp các ký hiệu và quy tắc sử dụng tập ký hiệu đó để biểu diễn và xác định giá trị các số. Mỗi hệ đếm có một số chữ số/ký số (digits) hữu hạn. Tổng số chữ số của mỗi hệ đếm được gọi là **cơ số** (base hay radix), ký hiệu là b .

5

Các hệ đếm cơ bản

- Hệ thập phân (Decimal System) → con người sử dụng
- Hệ nhị phân (Binary System) → máy tính sử dụng
- Hệ mười sáu (Hexadecimal System) → dùng để viết gọn cho số nhị phân
- Hệ bát phân (Octal System)

6

2.1.1. Hệ đếm cơ số b

- Hệ đếm cơ số b ($b \geq 2$ và nguyên dương) mang tính chất sau:
 - có b chữ số (ký số) để thể hiện giá trị số. Ký số nhỏ nhất là 0 và lớn nhất là $b-1$
 - giá trị (trọng số) vị trí thứ n trong một số của hệ đếm bằng cơ số b lũy thừa n: b^n
 - Số $N_{(b)}$ trong hệ đếm cơ số b được biểu diễn bởi:

$$N_{(b)} = a_n a_{n-1} a_{n-2} \dots a_1 a_0 a_{-1} a_{-2} \dots a_{-m}$$

7

2.1.1. Hệ đếm cơ số b

- trong đó, số $N_{(b)}$ có **n+1** chữ số biểu diễn cho phần nguyên và **m** chữ số lẻ biểu diễn cho phần b_phân, và có giá trị là:

$$N_{(b)} = a_n \cdot b^n + a_{n-1} \cdot b^{n-1} + a_{n-2} \cdot b^{n-2} + \dots + a_1 \cdot b^1 + a_0 \cdot b^0 + a_{-1} \cdot b^{-1} + a_{-2} \cdot b^{-2} + \dots + a_{-m} \cdot b^{-m}$$

hay là:

$$N_{(b)} = \sum_{i=-m}^n a_i \cdot b^i$$

8

2.1.2. Hệ đếm thập phân (Decimal System, b=10)

- Hệ đếm thập phân hay hệ đếm cơ số 10 là một trong các phát minh của người Ả rập cổ, bao gồm 10 chữ số theo ký hiệu sau: **0, 1, 2, 3, 4, 5, 6, 7, 8, 9**
- Quy tắc tính giá trị của hệ đếm này là mỗi đơn vị ở một hàng bất kỳ có giá trị bằng 10 đơn vị của hàng kế cận bên phải. Ở đây $b=10$

9

2.1.2. Hệ đếm thập phân (Decimal System, b=10)

- Bất kỳ số nguyên dương trong hệ thập phân có thể biểu diễn như là một tổng các số hạng, mỗi số hạng là tích của một số với 10 lũy thừa, trong đó số mũ lũy thừa được tăng thêm 1 đơn vị kể từ số mũ lũy thừa phía bên phải nó. Số mũ lũy thừa của hàng đơn vị trong hệ thập phân là 0

10

2.1.2. Hệ đếm thập phân (Decimal System, b=10)

- Ví dụ: Số 5246 có thể được biểu diễn như sau:

$$5246 = 5 \times 10^3 + 2 \times 10^2 + 4 \times 10^1 + 6 \times 10^0 \\ = 5 \times 1000 + 2 \times 100 + 4 \times 10 + 6 \times 1$$

- Thể hiện như trên gọi là ký hiệu mở rộng của số nguyên vì

$$5246 = 5000 + 200 + 40 + 6$$

11

2.1.2. Hệ đếm thập phân (Decimal System, b=10)

- Như vậy, trong số 5246: chữ số 6 trong số nguyên đại diện cho giá trị 6 đơn vị, chữ số 4 đại diện cho giá trị 4 chục (hàng chục), chữ số 2 đại diện cho giá trị 2 trăm (hàng trăm) và chữ số 5 đại diện cho giá trị 5 nghìn (hàng nghìn)

- Phần thập phân:

- $254.68 = 2 \times 10^2 + 5 \times 10^1 + 4 \times 10^0 + 6 \times 10^{-1} + 8 \times 10^{-2}$
$$= 200 + 50 + 4 + \frac{6}{10} + \frac{8}{100}$$

12

2.1.3. Hệ đếm nhị phân (Binary System, b=2)

- Với cơ số $b=2$, chúng ta có hệ đếm nhị phân. Đây là hệ đếm đơn giản nhất với 2 chữ số là 0 và 1. Mỗi chữ số nhị phân gọi là BIT (viết tắt từ chữ **Binary digiT**). Vì hệ nhị phân chỉ có 2 chữ số là 0 và 1, nên khi muốn diễn tả một số lớn hơn cần kết hợp nhiều bit với nhau. Ta có thể chuyển đổi số trong hệ nhị phân sang số trong hệ thập phân quen thuộc.

13

2.1.3. Hệ đếm nhị phân (Binary System, b=2)

- Ví dụ: Số $11101.11_{(2)}$ sẽ tương đương với giá trị thập phân là :

	1	1	1	0	1	.	1	1
Số nhị phân :	1	1	1	0	1	.	1	1
Số vị trí :	4	3	2	1	0	-1	-2	
Trị vị trí :	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	
Hệ 10 là :	16	8	4	2	1	0.5	0.25	

như vậy:

$$11101.11_{(2)} = 1 \times 16 + 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 + 1 \times 0.5 + 1 \times 0.25 = 29.75_{(10)}$$

số 10101 (hệ 2) sang hệ thập phân sẽ là:

$$10101_{(2)} = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 16 + 0 + 4 + 0 + 1 = 21_{(10)}$$

14

2.1.4. Hệ đếm bát phân

- Nếu dùng 1 tập hợp 3 bit thì có thể biểu diễn 8 trị khác nhau : 000, 001, 010, 011, 100, 101, 110, 111. Các trị này tương đương với 8 trị trong hệ thập phân là 0, 1, 2, 3, 4, 5, 6, 7. Tập hợp các chữ số này gọi là hệ bát phân, là hệ đếm với $b = 8 = 2^3$. Trong hệ bát phân, giá trị vị trí là lũy thừa của 8.

- Ví dụ:
 $235.64_{(8)} = 2 \times 8^2 + 3 \times 8^1 + 5 \times 8^0 + 6 \times 8^{-1} + 4 \times 8^{-2} = 157.8125_{(10)}$

15

2.1.5. Hệ đếm thập lục phân (Hexadecimal System, b=16)

- Hệ đếm thập lục phân là hệ cơ số $b=16 = 2^4$, tương đương với tập hợp 4 chữ số nhị phân (4 bit). Khi thể hiện ở dạng hexa-decimal, ta có 16 chữ số gồm 10 chữ số từ 0 đến 9, và 6 chữ in A, B, C, D, E, F để biểu diễn các giá trị số tương ứng là 10, 11, 12, 13, 14, 15. Với hệ thập lục phân, giá trị vị trí là lũy thừa của 16

16

2.1.5. Hệ đếm thập lục phân (Hexadecimal System, b=16)

- Ví dụ:
 $34F5C_{(16)} = 3 \times 16^4 + 4 \times 16^3 + 15 \times 16^2 + 5 \times 16^1 + 12 \times 16^0 = 216294_{(10)}$
- Ghi chú:* Một số ngôn ngữ lập trình quy định viết số hexa phải có chữ H ở cuối chữ số. Ví dụ: Số 15 viết là F_H .

17

2.1.6. Chuyển đổi một số từ hệ thập phân sang hệ cơ số b

- Đổi phần nguyên từ hệ thập phân sang hệ cơ số b.
 - Lấy số nguyên thập phân $N_{(10)}$ lần lượt chia cho b cho đến khi thương số bằng 0. Kết quả số chuyển đổi $N_{(b)}$ là các số dư trong phép chia viết theo thứ tự ngược lại.
- Đổi phần thập phân từ hệ thập phân sang hệ cơ số b
 - Lấy phần thập phân $N_{(10)}$ lần lượt nhân với b cho đến khi phần thập phân của tích số bằng 0. Kết quả số chuyển đổi $N_{(b)}$ là các số phần nguyên trong phép nhân viết ra theo thứ tự tính toán.

18

Lưu ý 1: Đổi từ hệ 10 sang hệ 2

- Chuyển đổi phần nguyên và phần lẻ riêng
- Chuyển đổi phần nguyên: 2 cách
 - Phân tích thành tổng các số lũy thừa của 2
 - Chia cho 2 được thương và số dư, sau đó lại lấy thương chia tiếp cho 2 cho đến khi thương = 0, viết các số dư theo thứ tự ngược lại

19

Đổi từ hệ 10 sang hệ 2

Ví dụ:
 $12 = 8 + 4 = 2^3 + 2^2$
 Kết quả: $12_{(10)} = 1100_{(2)}$

Kết quả: $12_{(10)} = 1100_{(2)}$

20

Đổi từ hệ 10 sang hệ 2

- Chuyển đổi phần lẻ
 - Lấy phần lẻ nhân 2 rồi lấy phần nguyên, ...
 → biểu diễn các phần nguyên theo chiều thuận
- Ví dụ:

$0.6875_{(10)} = ?_{(2)}$

0.6875	$\times 2 = 1$	$.375$	← phần thập phân của tích
0.3750	$\times 2 = 0$	$.75$	
0.75	$\times 2 = 1$	$.5$	
0.5	$\times 2 = 1$	$.0$	

↑ phần nguyên của tích

Kết quả: $0.6875_{(10)} = 0.1011_{(2)}$

21

Đổi từ hệ 10 sang hệ 2

- $12.6875_{(10)} = 1100.1011_{(2)}$

Kết quả: $12_{(10)} = 1100_{(2)}$

$0.6875_{(10)} = 0.1011_{(2)}$

22

Đổi từ hệ 10 sang hệ 2

- Bài tập: đổi số $35.375_{(10)}$ sang hệ 2

23

Lưu ý 2: chuyển đổi nhị phân sang Hexa

- Duyệt từ phải sang trái, chia thành các nhóm 4 bit, sau đó thay từng nhóm 4 bit bằng một chữ số Hexa
- Ví dụ:
 $10\ 0011_2 = 23_{16}$

Hệ thập phân	Hệ nhị phân	Hệ mười sáu
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

24

Chuyển đổi thập phân sang Hexa

- Thập phân → Hexa: 14988 → ?
14988 : 16 = 936 dư 12 tức là C
936 : 16 = 58 dư 8
58 : 16 = 3 dư 10 tức là A
3 : 16 = 0 dư 3
Như vậy, ta có: $14988_{(10)} = 3A8C_{(16)}$

25

2.1.7. Mệnh đề logic

- Mệnh đề logic là mệnh đề chỉ nhận một trong 2 giá trị : Đúng (TRUE) hoặc Sai (FALSE), tương đương với TRUE = 1 và FALSE = 0.
- Qui tắc: TRUE = NOT FALSE và FALSE = NOT TRUE
- Phép toán logic áp dụng cho 2 giá trị TRUE và FALSE ứng với tổ hợp AND (và) và OR (hoặc) như sau:

26

Mệnh đề logic

x	y	X AND y	X OR y
TRUE	TRUE	TRUE	TRUE
TRUE	FALSE	FALSE	TRUE
FALSE	TRUE	FALSE	TRUE
FALSE	FALSE	FALSE	FALSE

27

2.2. Biểu diễn dữ liệu trong máy tính và đơn vị thông tin

- 2.2.1. Nguyên tắc chung
 - Thông tin và dữ liệu mà con người hiểu được tồn tại dưới nhiều dạng khác nhau, ví dụ như các số liệu, các ký tự văn bản, âm thanh, hình ảnh,... nhưng trong máy tính mọi thông tin và dữ liệu đều được biểu diễn bằng số nhị phân (chuỗi bit).
 - Để đưa dữ liệu vào cho máy tính, cần phải mã hoá nó về dạng nhị phân. Với các kiểu dữ liệu khác nhau cần có cách mã hoá khác nhau. Cụ thể:

28

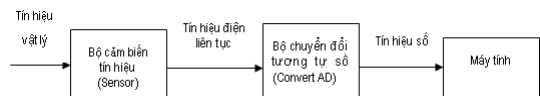
Nguyên tắc chung (tiếp)

- Các dữ liệu dạng số (số nguyên hay số thực) sẽ được chuyển đổi trực tiếp thành các chuỗi số nhị phân theo các chuẩn xác định.
- Các ký tự được mã hoá theo một bộ mã cụ thể, có nghĩa là mỗi ký tự sẽ tương ứng với một chuỗi số nhị phân.
- Các dữ liệu phi số khác như âm thanh, hình ảnh và nhiều đại lượng vật lý khác muốn đưa vào máy phải **số hoá** (*digitalizing*). Có thể hiểu một cách đơn giản khái niệm số hoá như sau: các dữ liệu tự nhiên thường là quá trình biến đổi liên tục, vì vậy để đưa vào máy tính, nó cần được biến đổi sang một dãy hữu hạn các giá trị số (nguyên hay thực) và được biểu diễn dưới dạng nhị phân.

29

Nguyên tắc chung (tiếp)

- Với các tín hiệu như âm thanh, video, hay các tín hiệu vật lý khác, qui trình mã hoá được biểu diễn như sau:



30

Nguyên tắc chung (tiếp)

- Tuy rằng mọi dữ liệu trong máy tính đều ở dạng nhị phân, song do bản chất của dữ liệu, người ta thường phân dữ liệu thành 2 dạng:
 - Dạng cơ bản:** gồm dạng số (nguyên hay thực) và dạng ký tự. Số nguyên không dấu được biểu diễn theo dạng nhị phân thông thường, số nguyên có dấu theo mã bù hai, còn số thực theo dạng dấu phẩy động. Để biểu diễn một dữ liệu cơ bản, người ta sử dụng 1 số bit. Các bit này ghép lại với nhau để tạo thành từ: từ 8 bit, từ 16 bit,...
 - Dạng có cấu trúc:** Trên cơ sở dữ liệu cơ bản, trong máy tính, người ta xây dựng nên các dữ liệu có cấu trúc phục vụ cho các mục đích sử dụng khác nhau. Tùy theo cách "ghép" chúng ta có mảng, tập hợp, xâu, bản ghi,...

31

2.2.2. Đơn vị thông tin

- Đơn vị nhỏ nhất để biểu diễn thông tin gọi là **bit**. Một bit tương ứng với một sự kiện có 1 trong 2 trạng thái.
- Ví dụ: Một mạch đèn có 2 trạng thái là:
 - Tắt (Off) khi mạch điện qua công tắc là hở
 - Mở (On) khi mạch điện qua công tắc là đóng
- Số học nhị phân sử dụng hai ký số 0 và 1 để biểu diễn các số. Vì khả năng sử dụng hai số 0 và 1 là như nhau nên một chỉ thị chỉ gồm một chữ số nhị phân có thể xem như là đơn vị chứa thông tin nhỏ nhất.

32

Đơn vị thông tin (tiếp)

- Bit là chữ viết tắt của **BI**nary digi**T**. Trong tin học, người ta thường sử dụng các đơn vị đo thông tin lớn hơn như sau:

Tên gọi	Ký hiệu	Giá trị
Byte	B	8 bit
KiloByte	KB	2^{10} B = 1024 Byte
MegaByte	MB	2^{20} B
GigaByte	GB	2^{30} B
TeraByte	TB	2^{40} B

33

2.3. Biểu diễn số nguyên

- Số nguyên gồm số nguyên **không dấu** và số nguyên **có dấu**.
- Về nguyên tắc đều dùng 1 chuỗi bit để biểu diễn.
- Đối với số nguyên có dấu, người ta sử dụng bit đầu tiên để biểu diễn dấu và bit này gọi là bit dấu.

34

2.3.1. Biểu diễn số nguyên không dấu

- Dạng tổng quát: giả sử dùng n bit để biểu diễn cho một số nguyên không dấu A:

$$a_{n-1}a_{n-2}\dots a_3a_2a_1a_0$$

- Giá trị của A được tính như sau:

$$A = a_{n-1}2^{n-1} + a_{n-2}2^{n-2} + \dots + a_12^1 + a_02^0$$

$$A = \sum_{i=0}^{n-1} a_i 2^i$$

- Dải biểu diễn của A: từ 0 đến $2^n - 1$

35

Ví dụ:

- Biểu diễn các số nguyên không dấu sau đây bằng 8 bit:

$$A = 45 \qquad B = 156$$

Giải:

$$A = 45 = 32 + 8 + 4 + 1 = 2^5 + 2^3 + 2^2 + 2^0$$

$$\rightarrow A = 0010\ 1101$$

$$B = 156 = 128 + 16 + 8 + 4 = 2^7 + 2^4 + 2^3 + 2^2$$

$$\rightarrow B = 1001\ 1100$$

36

Ví dụ (tiếp)

- Cho các số nguyên không dấu X, Y được biểu diễn bằng 8 bit như sau:

$$X = 0010\ 1011$$

$$Y = 1001\ 0110$$

Xác định giá trị của X, Y

Giải:

$$X = 0010\ 1011 = 2^5 + 2^3 + 2^1 + 2^0 = 32 + 8 + 2 + 1 = 43$$

$$Y = 1001\ 0110 = 2^7 + 2^4 + 2^2 + 2^1 = 128 + 16 + 4 + 2 = 150$$

37

Với n = 8 bit

- Dài biểu diễn là [0, 255]
- Trục số học máy tính:

$$0000\ 0000 = 0$$

$$0000\ 0001 = 1$$

$$0000\ 0010 = 2$$

$$0000\ 0011 = 3$$

.....

$$1111\ 1111 = 255$$



38

Biểu diễn số nguyên không dấu

- Với n = 16 bit:
 - dài biểu diễn: [0, 65535]
- Với n = 32 bit:
 - dài biểu diễn: [0, 2³²-1]

39

2.3.2. Biểu diễn số nguyên có dấu

- Khái niệm về số bù
 - Số bù 9 và số bù 10 (hệ thập phân)
 - Giả sử có 1 số nguyên có dấu A được biểu diễn bởi n chữ số thập phân.
 - Số bù 9 của A: $(10^n - 1) - A$
 - Số bù 10 của A: $10^n - A$
 - Số bù 10 = số bù 9 + 1

40

Biểu diễn số nguyên có dấu

- Số bù 1 và số bù 2 (hệ nhị phân)
 - Giả sử có 1 số nguyên nhị phân A được biểu diễn = n bit nhị phân
 - Số bù 1 của A: $(2^n - 1) - A$
 - Số bù 2 của A: $2^n - A$
 - Số bù 2 = số bù 1 + 1

41

Số bù 1 và bù 2 (tiếp)

- Ví dụ: n = 4 bit, A = 0110

$$\begin{array}{r} 1111 \\ - 0110 \\ \hline \end{array}$$

Số bù 1: **1001**

Nhận xét: số bù 1 là đảo các bit 0 ↔ 1

$$\begin{array}{r} 10000 \\ - 0110 \\ \hline \end{array}$$

Số bù 2: **1010**

Nhận xét: A + số bù 2 của nó, bỏ bit ngoài cùng đi, ta được 0000

= số bù 1 + 1

42

Biểu diễn số nguyên có dấu bằng số bù 2

- Dùng n bit để biểu diễn số nguyên có dấu:

$$a_{n-1}a_{n-2}\dots a_2a_1a_0$$

- Với số dương:

- bit $a_{n-1} = 0$
- các bit còn lại biểu diễn độ lớn của số dương đó
- Dạng tổng quát của số dương: $0a_{n-2}\dots a_2a_1a_0$
- Giá trị của số dương:

$$A = \sum_{i=0}^{n-2} a_i \times 2^i$$

- Dài biểu diễn: $[0, 2^{n-1}-1]$

43

Biểu diễn số nguyên có dấu bằng số bù 2

- Với số âm: được biểu diễn bằng số bù 2 của số dương tương ứng

- bit $a_{n-1} = 1$
- Dạng tổng quát của số âm: $1a_{n-2}\dots a_2a_1a_0$
- Giá trị của số âm:

$$A = -2^{n-1} + \sum_{i=0}^{n-2} a_i \times 2^i$$

- Dài biểu diễn: $[-2^{n-1}, -1]$

44

Biểu diễn số nguyên có dấu bằng số bù 2

- Kết hợp lại, ta có dài biểu diễn của số nguyên có dấu n bit là:

- $[-2^{n-1}, 2^{n-1} - 1]$

- Công thức tổng quát:

$$A = -a_{n-1} \times 2^{n-1} + \sum_{i=0}^{n-2} a_i \times 2^i$$

45

Một số ví dụ về số nguyên có dấu

- Xác định giá trị của các số nguyên có dấu 8 bit sau đây:

$$A = 0101\ 0110$$

$$B = 1101\ 0010$$

Giải:

$$A = 2^6 + 2^4 + 2^2 + 2^1 = 64 + 16 + 4 + 2 = +86$$

$$B = -2^7 + 2^6 + 2^4 + 2^1 = -128 + 64 + 16 + 2 = -46$$

46

Bài tập

- Biểu diễn các số nguyên sau với n = 8 bit:

- X = +58
- Y = -80

- Xác định giá trị của số nguyên có dấu 8 bit: Z = 1100 1001

47

Trường hợp cụ thể

- Trường hợp 8 bit: biểu diễn các giá trị từ -128 đến +127

$$0000\ 0000 = 0$$

$$0000\ 0001 = +1$$

$$\dots\dots\dots$$

$$0111\ 1111 = +127$$

$$1000\ 0000 = -128$$

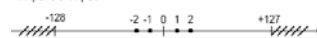
$$1000\ 0001 = -127$$

$$\dots\dots\dots$$

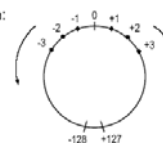
$$1111\ 1110 = -2$$

$$1111\ 1111 = -1$$

Trục số học:



Trục số học máy tính:



48

Trường hợp cụ thể

- Với $n = 16$ bit, dải biểu diễn:
 - $[-32768, +32767]$
- Với $n = 32$ bit: -2^{31} đến $2^{31} - 1$
- Với $n = 64$ bit: -2^{63} đến $2^{63} - 1$
- Chuyển đổi từ byte thành word:
 - đối với số dương thêm 8 bit 0 bên trái
 - $+19 = 0001\ 0011$ (8 bit)
 - $+19 = 0000\ 0000\ 0001\ 0011$ (16 bit)
 - đối với số âm thêm 8 bit 1 bên trái
 - $-19 = 1110\ 1101$ (8 bit)
 - $-19 = 1111\ 1111\ 1110\ 1101$ (16 bit)

49

Binary Code Decimal Code

- Dùng 4 bit để mã hóa từng chữ số thập phân từ 0 đến 9

0 → 0000
 1 → 0001 8 → 1000
 9 → 1001

- Có 6 tổ hợp không dùng: 1010, 1011, 1100, 1101, 1110, 1111

50

Binary Code Decimal Code

- $35 \rightarrow 0011\ 0101_{\text{BCD}}$
- $61 \rightarrow 0110\ 0001_{\text{BCD}}$
- $1087 \rightarrow 0001\ 0000\ 1000\ 0111_{\text{BCD}}$
- Cứ 1 chữ số thập phân đơn lẻ được mã hóa bằng 4 bit

51

Binary Code Decimal Code

- Phép cộng số BCD:

$$\begin{array}{r} 35 \rightarrow 0011\ 0101_{\text{BCD}} \\ + 61 \rightarrow +0110\ 0001_{\text{BCD}} \\ \hline 96 \leftarrow 1001\ 0110_{\text{BCD}} \end{array}$$

Kết quả đúng, không phải hiệu chỉnh

$$\begin{array}{r} 87 \rightarrow 1000\ 0111_{\text{BCD}} \\ + 96 \rightarrow +1001\ 0110_{\text{BCD}} \\ \hline 183 \leftarrow 1\ 0001\ 1101_{\text{BCD}} \end{array}$$

Kết quả sai, phải hiệu chỉnh

52

Binary Code Decimal Code

- Hiệu chỉnh:
 - Nhận xét: $7 + 6$ hay $8 + 9$ đều vượt 9 nên có nhớ.
 - Hiệu chỉnh bằng cách cộng thêm 6 ở những vị trí có nhớ (>9)
- $$\begin{array}{r} 1\ 0001\ 1101 \\ + 0110\ 0110 \leftarrow \text{hiệu chỉnh} \\ \hline 0001\ 1000\ 0011_{\text{BCD}} \rightarrow \text{kết quả đúng} \end{array}$$

53

Các kiểu lưu trữ số BCD

- BCD không gói (Unpacked BCD): mỗi số BCD 4 bit được lưu trữ trong 4 bit thấp của mỗi byte. Ví dụ: Số 35 được lưu trữ:

0011	0101
------	------

- BCD gói (packed BCD): hai số BCD được lưu trữ trong một byte. Ví dụ: Số 35 được lưu trữ:

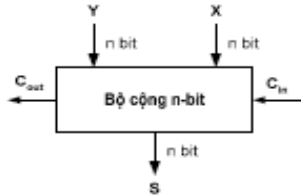
0011	0101
------	------

54

2.4. Các phép toán số học với số nguyên

Phép cộng số nguyên không dấu

Bộ cộng n-bit



55

2.4. Các phép toán số học với số nguyên

Phép cộng số nguyên không dấu

- Tiến hành cộng lần lượt từng bit từ phải qua trái.
- Khi cộng hai số nguyên không dấu n bits ta thu được một số nguyên không dấu cũng n bits.
- Nếu tổng của hai số đó lớn hơn 2^{n-1} thì khi đó sẽ tràn số ($C_{out} = 1$) và kết quả sẽ là sai.
- Để tránh hiện tượng này, ta dùng nhiều bit hơn

56

Ví dụ phép cộng số nguyên không dấu

- Với trường hợp 8 bit, nếu tổng nhỏ hơn 255 thì kết quả đúng

$$\begin{array}{r} 57 = 00111001 \\ + 34 = 00100010 \\ \hline 91 = 01011011 \\ \\ 209 = 11010001 \\ + 73 = 01001001 \\ \hline 282 = 100011010 \end{array}$$

Bit tràn ra ngoài \Rightarrow kết quả = 26 là sai.

57

Phép đảo dấu

- Phép đảo dấu thực chất là lấy bù 2

+37 = 0010 0101	-37 = 1101 1011
bù 1: 1101 1010	bù 1: 0010 0100
<u> +1</u>	<u> +1</u>
bù 2: 1101 1011 = -37	bù 2: 0010 0101 = +37

58

Cộng hai số nguyên có dấu

Khi cộng 2 số nguyên có dấu n bit, không quan tâm đến bit Cout, và kết quả nhận được là n bit:

- Cộng 2 số **khác dấu** kết quả **luôn đúng**
- Cộng 2 số **cùng dấu**:
 - nếu dấu kết quả cùng dấu với các số hạng thì **kết quả là đúng**.
 - nếu kết quả có dấu ngược lại, khi đó có **tràn xảy ra** (Overflow) và **kết quả bị sai**
- Tràn xảy ra khi tổng nằm ngoài dải biểu diễn $[-(2^{n-1}), +(2^{n-1} - 1)]$

59

Cộng hai số nguyên có dấu- ví dụ:

$$\begin{array}{r} (+70) = 0100 0110 \\ +(+42) = 0010 1010 \\ \hline +112 = 0111 0000 = +112 \\ \\ (+97) = 0110 0001 \\ +(-52) = 1100 1100 \quad (\text{vì } +52 = 0011 0100) \\ \hline +45 = 1 0010 1101 = +45 \end{array}$$

60

Cộng hai số nguyên có dấu- ví dụ:

$$\begin{aligned}
 (+75) &= 0100\ 1011 \\
 +(+82) &= 0101\ 0010 \\
 \hline
 +157 &= 1001\ 1101 = -99 \\
 \text{tổng vượt } +127 &\rightarrow \text{chuyển sang bên âm} \\
 (-104) &= 1001\ 1000 \quad (\text{vì } +104 = 0110\ 1000) \\
 +(-43) &= 1101\ 0101 \quad (\text{vì } +43 = 0010\ 1011) \\
 \hline
 -147 &= 1\ 0110\ 1101 = +109 \rightarrow \text{sai}
 \end{aligned}$$

không
quan tâm

âm + âm \rightarrow dương

61

Nguyên tắc thực hiện phép trừ

- Phép trừ hai số nguyên: $X - Y = X + (-Y)$
- Nguyên tắc: lấy bù 2 của số trừ Y để được $-Y$, sau đó cộng với số bị trừ X

62

Nhân số nguyên không dấu

$$\begin{array}{r}
 1011 \quad \text{Số bị nhân (11)} \\
 \times 1101 \quad \text{Số nhân (13)} \\
 \hline
 1011 \\
 0000 \\
 1011 \\
 1011 \\
 \hline
 10001111 \quad \text{Tích (143)}
 \end{array}$$

Các tích riêng phần

63

Nhân số nguyên không dấu

- Các tích riêng phần được xác định như sau:
 - nếu bit của số nhân = 0 thì tích riêng phần = 0
 - nếu bit của số nhân = 1 thì tích riêng phần = số bị nhân
 - tích riêng phần tiếp theo được dịch trái so với tích riêng phần trước đó
- Tích = tổng các tích riêng phần
- Nhân 2 số nguyên n bit, tích có độ dài là 2n bit (không bao giờ tràn)

64

Nhân hai số nguyên có dấu

- Sử dụng thuật giải nhân hai số nguyên không dấu
- Bước 1: chuyển đổi số bị nhân và số nhân thành số dương tương ứng
- Bước 2: nhân 2 số dương bằng thuật giải đã học, được tích của 2 số dương
- Bước 3: hiệu chỉnh dấu của tích như sau:
 - nếu 2 thừa số ban đầu cùng dấu thì không cần hiệu chỉnh
 - nếu 2 thừa số ban đầu là khác dấu thì ta lấy bù 2 của tích ở kết quả bước 2

65

Chia số nguyên không dấu

$$\begin{array}{r}
 \text{Số bị chia} \quad 10010011 \\
 \begin{array}{r}
 1011 \\
 001110 \\
 1011 \\
 001111 \\
 1011 \\
 100
 \end{array} \\
 \hline
 \text{Số chia} \quad 1011 \\
 \text{Thương} \quad 00001101 \\
 \\
 \text{Phần dư} \quad 100
 \end{array}$$

66

Chia số nguyên có dấu

- Bước 1: Chuyển đổi số bị chia và số chia về thành số dương tương ứng.
- Bước 2: Sử dụng thuật giải chia số nguyên không dấu để chia hai số dương, kết quả nhận được là thương Q và phần dư R đều là dương
- Bước 3: Hiệu chỉnh dấu của kết quả như sau:
(Lưu ý: phép đảo dấu thực chất là phép lấy bù hai)

Số bị chia	Số chia	Thương	Số dư
dương	dương	giữ nguyên	giữ nguyên
dương	âm	đảo dấu	giữ nguyên
âm	dương	đảo dấu	đảo dấu
âm	âm	giữ nguyên	đảo dấu

67

2.5. Tính toán logic với số nhị phân

		AND	OR	XOR
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

68

2.5. Tính toán logic với số nhị phân

	NOT
0	1
1	0

69

2.5. Tính toán logic với số nhị phân

- Thực hiện các phép toán logic với 2 số nhị phân:
 - Kết quả là 1 số nhị phân khi thực hiện các phép toán logic với từng cặp bit của 2 số nhị phân đó
 - Các phép toán này chỉ tác động lên từng cặp bit mà không ảnh hưởng đến bit khác.

70

2.5. Tính toán logic với số nhị phân

VĐ: A = 1010 1010 và B = 0000 1111

	AND	OR	XOR	NOT
1010 1010				01010101
0000 1111				11110000
	00001010	10101111	10100101	

Nhận xét: +Phép AND dùng để xoá một số bit và giữ nguyên 1 số bit còn lại.
+Phép OR dùng để thiết lập 1 số bit và giữ nguyên 1 số bit khác.

71

2.6. Biểu diễn ký tự

Nguyên tắc chung:

- Các ký tự cũng cần được chuyển đổi thành chuỗi bit nhị phân gọi là **mã ký tự**.
- Số bit dùng cho mỗi ký tự theo các mã khác nhau là khác nhau.

Vd : Bộ mã ASCII dùng 8 bit cho 1 ký tự.

Bộ mã Unicode dùng 16 bit.

72

Bộ mã ASCII (American Standard Code for Information Interchange)

- Do ANSI (American National Standard Institute) thiết kế
- <http://www.asciitable.com/>
- ASCII là bộ mã được dùng để *trao đổi thông tin chuẩn của Mỹ*. Lúc đầu chỉ dùng 7 bit (128 ký tự) sau đó mở rộng cho 8 bit và có thể biểu diễn 256 ký tự khác nhau trong máy tính
- Bộ mã 8 bit → mã hóa được cho $2^8 = 256$ ký tự, có mã từ $00_{16} \div FF_{16}$, bao gồm:
 - 128 ký tự chuẩn có mã từ $00_{16} \div 7F_{16}$
 - 128 ký tự mở rộng có mã từ $80_{16} \div FF_{16}$

73

ASCII Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	SPC	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

Physical Device Controls: Format Effectors

Bộ mã ASCII (tiếp)

- 95 ký tự hiển thị được: có mã từ $20_{16} \div 7E_{16}$
 - 26 chữ cái hoa Latin 'A' ÷ 'Z' có mã từ $41_{16} \div 5A_{16}$
 - 26 chữ cái thường Latin 'a' ÷ 'z' có mã từ $61_{16} \div 7A_{16}$
 - 10 chữ số thập phân '0' ÷ '9' có mã từ $30_{16} \div 39_{16}$

75

Bộ mã ASCII (tiếp)

- 95 ký tự hiển thị được:
 - Các dấu câu: . , ? ! : ; ...
 - Các dấu phép toán: + - * / ...
 - Một số ký tự thông dụng: #, \$, &, @, ...
 - Dấu cách (mã là 20_{16})
- 33 mã điều khiển: mã từ $00_{16} \div 1F_{16}$ và $7F_{16}$ dùng để mã hóa cho các chức năng điều khiển

76

Điều khiển định dạng

BS	Backspace – Lùi lại một vị trí. Ký tự điều khiển con trỏ lùi lại một vị trí.
HT	Horizontal Tab – Ký tự điều khiển con trỏ dịch đi một khoảng định trước
LF	Line Feed – Ký tự điều khiển con trỏ xuống dòng
VT	Vertical Tab – Ký tự điều khiển con trỏ dịch đi một số dòng
FF	Form Feed – Ký tự điều khiển con trỏ chuyển xuống đầu trang tiếp theo.
CR	Carriage Return – Ký tự điều khiển con trỏ về đầu dòng hiện hành.

77

Các ký tự mở rộng của bảng mã ASCII

- Được định nghĩa bởi:
 - Nhà chế tạo máy tính
 - Người phát triển phần mềm
- Ví dụ:
 - Bộ mã ký tự mở rộng của IBM: được dùng trên máy tính IBM-PC.
 - Bộ mã ký tự mở rộng của Apple: được dùng trên máy tính Macintosh.
 - Các nhà phát triển phần mềm tiếng Việt cũng đã thay đổi phần này để mã hoá cho các ký tự riêng của chữ Việt, ví dụ như bộ mã TCVN 5712.

78

Bộ mã Unicode

- Do các hãng máy tính hàng đầu thiết kế
- Là bộ mã 16-bit, Vậy số ký tự có thể biểu diễn (mã hoá) là 2^{16}
- Được thiết kế cho đa ngôn ngữ, trong đó có tiếng Việt

79

2.7. Biểu diễn số thực

2.7.1. Nguyên tắc chung

- Để biểu diễn số thực, trong máy tính người ta thường dùng ký pháp dấu phẩy động (Floating Point Number).
- Tổng quát: một số thực X được biểu diễn theo kiểu số dấu phẩy động như sau:
 - $X = M * R^E$
 - M là phần định trị (Mantissa)
 - R là cơ số (Radix)
 - E là phần mũ (Exponent)

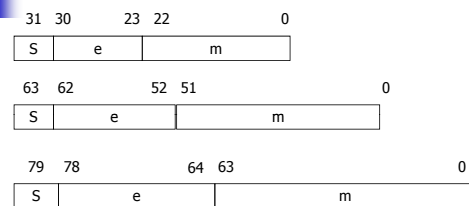
80

2.7.2. Chuẩn IEEE754/85

- Cơ số $R = 2$
- Các dạng:
 - 32 – bit (4 byte float trong C)
 - 48 – bit (real trong Pascal)
 - 64 – bit (8 byte)
 - 80 – bit (10 byte)

81

Các dạng biểu diễn chính



- trường S nằm bên trái nhất biểu diễn dấu
- e : mũ
- m : định trị

82

Dạng 32 – bit

- S là bit dấu
 - $S = 0$: số dương
 - $S = 1$: số âm
- e (8 bit) là mã excess – 127 của phần mũ E :
 - $E = e - 127$
 - khi $e = 0$ thì phần mũ = -127, khi $e = 127$ thì phần mũ = 0
 - $e_{max} = 255$ (8 bit)
 - giá trị 127 gọi là độ lệch (bias)
- m (23 bit) là phần lẻ của phần định trị M : $M=1.m$
- Công thức xác định giá trị của số thực:
$$X = (-1)^S * 1.m * 2^{e-127}$$

83

Dạng 32 – bit

Các quy ước đặc biệt

- Các bit của $e = 0$, các bit của $m = 0$ thì $X = \pm 0$
 $x000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000 \rightarrow X = \pm 0$
- Các bit của $e = 1$, các bit của $m = 0$ thì $X = \pm \infty$
 $x111\ 1111\ 1000\ 0000\ 0000\ 0000\ 0000\ 0000 \rightarrow X = \pm \infty$
- Các bit của $e = 1$, còn m có ít nhất 1 bit = 1 thì nó không biểu diễn cho số nào cả (NaN – Not A Number)

84

Dạng 32 – bit

- Dải biểu diễn giá trị
 - 2^{-127} đến 2^{+127}
 - 10^{-38} đến 10^{+38}

85

Dạng 32 – bit. Ví dụ:

- Xác định giá trị của số thực được biểu diễn bằng 32 bit như sau:
 - 1100 0001 0101 0110 0000 0000 0000 0000
- S = 1 → số âm
- e = 1000 0010₂ = 130 → E = 130 – 127 = 3
- Vậy, X = -1.10101100 * 2³ = -1101.011 = -13.375

86

Dạng 32 – bit. Ví dụ (tiếp):

- 0011 1111 1000 0000 0000 0000 0000 0000
- Kết quả = +1.0

87

Dạng 64 – bit

- S là bit dấu
- e (11 bit): mã excess-1023 của phần mũ E → E = e – 1023
- m (52 bit): phần lẻ của phần định trị M
- Giá trị số thực:

$$X = (-1)^S * 1.m * 2^{e-1023}$$
- Dải giá trị biểu diễn: 10⁻³⁰⁸ đến 10⁺³⁰⁸

88

Dạng 80 – bit

- S là bit dấu
- e (15 bit): mã excess-16383 của phần mũ E → E = e – 16383
- m (64 bit): phần lẻ của phần định trị M
- Giá trị số thực:

$$X = (-1)^S * 1.m * 2^{e-16383}$$
- Dải giá trị biểu diễn: 10⁻⁴⁹³² đến 10⁺⁴⁹³²

89

Thực hiện phép toán số dấu phẩy động

- X1 = M1 * R^{E1}
- X2 = M2 * R^{E2}
- Ta có:
 - X1 * X2 = (M1 * M2) * R^{E1+E2}
 - X1 / X2 = (M1 / M2) * R^{E1 - E2}
 - X1 ± X2 = (M1 * R^{E1-E2} ± M2) * R^{E2}, với E2 ≥ E1

90

Các khả năng tràn số

- Tràn trên số mũ (Exponent Overflow): mũ dương vượt ra khỏi giá trị cực đại của số mũ dương có thể ($\rightarrow \infty$)
- Tràn dưới số mũ (Exponent Underflow): mũ âm vượt ra khỏi giá trị cực đại của số mũ âm có thể ($\rightarrow 0$)
- Tràn trên phần định trị (Mantissa Overflow): cộng hai phần định trị có cùng dấu, kết quả bị nhớ ra ngoài bit cao nhất.
- Tràn dưới phần định trị (Mantissa Underflow): Khi hiệu chỉnh phần định trị, các số bị mất ở bên phải phần định trị.

91

Phép cộng và phép trừ

- Kiểm tra các số hạng có bằng 0 hay không.
- Hiệu chỉnh phần định trị.
- Cộng hoặc trừ phần định trị.
- Chuẩn hóa kết quả.

92

Hỏi - đáp



93

Lời hay ý đẹp

“Hạnh phúc lớn nhất của một nhà bác học là thấy học trò của mình vượt thầy”

Lopée

“Nước trong quá thì không có cá, người xét nét quá thì không có bạn bè”

Cổ ngôn Trung Hoa

94

Chương 3: Hệ thống máy tính

Nguyễn Hồng Phương
Email: phuongnh-fit@mail.hut.edu.vn
Website: <http://is.hut.edu.vn/~phuongnh>
Bộ môn Hệ thống thông tin
Viện Công nghệ thông tin và Truyền thông
Đại học Bách Khoa Hà Nội

1

Nội dung chương 3

- 3.1. Giới thiệu
- 3.2. Chức năng và các thành phần của máy tính
- 3.3. Liên kết hệ thống
- 3.4. Hoạt động của máy tính
- 3.5. Phần mềm máy tính

2

3.1. Giới thiệu

- Hình dạng và cấu trúc máy tính: đa dạng
- Máy tính điện tử được xem là hệ xử lý thông tin gồm 2 phần chính

Phần cứng (Hardware)

Các cấu kiện, linh kiện điện, điện tử trong một hệ máy

Thể xác

Phần mềm (Software)

Bộ chương trình gồm các chỉ thị điện tử ra lệnh cho máy tính thực hiện nhiệm vụ nào đó theo yêu cầu của người sử dụng

Linh hồn

3

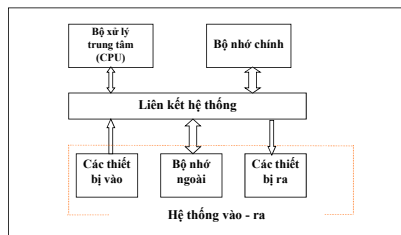
3.2. Chức năng và các thành phần của máy tính

- Chức năng cơ bản của hệ thống máy tính:
 - Xử lý dữ liệu
 - Lưu trữ dữ liệu
 - Trao đổi dữ liệu
 - Điều khiển

4

Các thành phần của máy tính

- Bộ xử lý trung tâm (Central Processing Unit)
- Bộ nhớ (Memory)
- Hệ thống vào ra (Input/Output System)
- Liên kết hệ thống (System Interconnection)



5

Minh họa máy tính



6

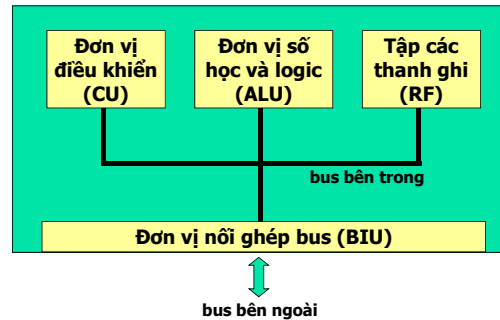
Bộ xử lý trung tâm (CPU)

- Chức năng
 - điều khiển hoạt động của máy tính
 - xử lý dữ liệu
- Nguyên tắc hoạt động cơ bản
 - CPU hoạt động theo chương trình nằm trong bộ nhớ chính



7

Cấu trúc cơ bản của CPU



8

Các thành phần cơ bản của CPU

- Đơn vị điều khiển (Control Unit – CU): điều khiển hoạt động của máy tính theo chương trình đã định sẵn.
- Đơn vị số học và logic (Arithmetic and Logic Unit - ALU): thực hiện các phép toán số học và các phép toán logic trên các dữ liệu cụ thể.
- Tập thanh ghi (Register File - RF): lưu giữ các thông tin tạm thời phục vụ cho hoạt động của CPU.
- Đơn vị nối ghép bus (Bus Interface Unit - BIU): kết nối và trao đổi thông tin giữa bus bên trong (internal bus) và bus bên ngoài (external bus).

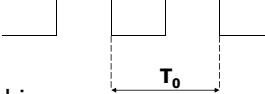
9

Tốc độ của bộ xử lý

- Tốc độ của bộ xử lý:
 - Số lệnh được thực hiện trong 1 giây
 - MIPS (Million of Instructions per Second)
 - Khó đánh giá chính xác
- Tần số xung nhịp của bộ xử lý:
 - Bộ xử lý hoạt động theo một xung nhịp (clock) có tần số xác định.
 - Tốc độ của bộ xử lý được đánh giá gián tiếp thông qua tần số của xung nhịp.

10

Tốc độ của bộ xử lý

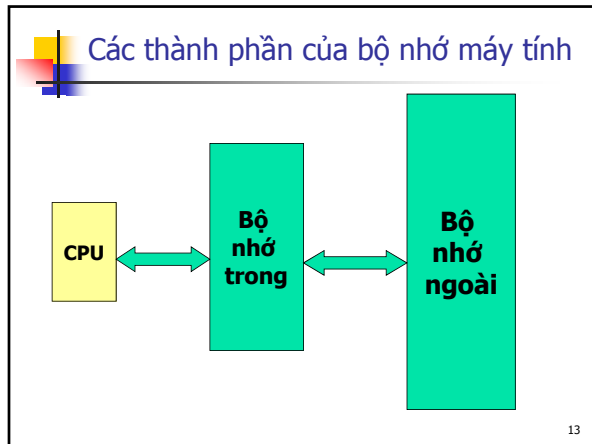
- Dạng xung nhịp: 
- T_0 là chu kỳ xung nhịp
- Mỗi thao tác của bộ xử lý cần kT_0 , $k \in \mathbb{N}$
- Tần số xung nhịp $f_0 = 1/T_0$
- Ví dụ: máy tính dùng bộ xử lý Pentium IV 2GHz
 - $f_0 = 2\text{GHz} = 2 \times 10^9 \text{ Hz}$
 - $T_0 = 1/f_0 = 1/(2 \times 10^9) = 0.5 \text{ (ns)}$

11

Bộ nhớ máy tính

- Chức năng: lưu trữ chương trình và dữ liệu.
- Các thao tác cơ bản với bộ nhớ:
 - Thao tác đọc (Read)
 - Thao tác ghi (Write)
- Các thành phần chính
 - Bộ nhớ trong (Internal Memory)
 - Bộ nhớ ngoài (External Memory)

12



Bộ nhớ trong

- Chức năng và đặc điểm
 - Chứa các thông tin mà CPU có thể trao đổi trực tiếp.
 - Tốc độ rất nhanh, dung lượng không lớn.
 - Sử dụng bộ nhớ bán dẫn: ROM, RAM
 - ROM (Read Only Memory): là bộ nhớ chỉ đọc thông tin, dùng để lưu trữ các chương trình hệ thống, chương trình điều khiển xuất nhập cơ sở (ROM-BIOS: ROM Basic Input/Output System). **Không mất thông tin ngay cả khi không có điện.**
 - RAM (Random Access Memory): là bộ nhớ truy xuất ngẫu nhiên, dùng để truy xuất dữ liệu và chương trình trong quá trình thao tác và tính toán. **Thông tin sẽ mất khi mất điện.**

14

Bộ nhớ trong

- Các loại bộ nhớ trong:
 - Bộ nhớ chính
 - Bộ nhớ cache (bộ nhớ đệm nhanh)

15

Bộ nhớ chính (Main Memory)

- Chứa các chương trình và dữ liệu đang được CPU sử dụng.
- Tổ chức thành các ngăn nhớ được đánh địa chỉ.
- Ngăn nhớ thường được tổ chức theo byte.
- Nội dung của ngăn nhớ có thể thay đổi, song địa chỉ vật lý của ngăn nhớ luôn cố định.

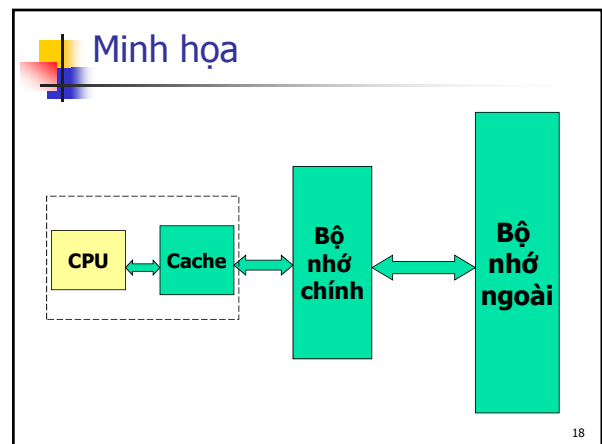
Nội dung	Địa chỉ
10101100	0000
01001100	0001
11100111	0010
01010011	0011
00001010	0100
11001010	0101
01110000	0110
10110110	0111

16

Bộ nhớ đệm nhanh (cache memory)

- Bộ nhớ có tốc độ nhanh được đặt đệm giữa CPU và bộ nhớ chính nhằm tăng tốc độ CPU truy cập bộ nhớ.
- Dung lượng nhỏ hơn bộ nhớ chính.
- Tốc độ nhanh hơn.
- Cache thường được chia thành một số mức
- Cache có thể được tích hợp trên chip vi xử lý.
- Cache có thể có hoặc không.

17



Bộ nhớ ngoài (External Memory)

- Chức năng và đặc điểm
 - Lưu giữ tài nguyên phần mềm của máy tính.
 - Được kết nối với hệ thống dưới dạng các thiết bị vào ra.
 - Dung lượng lớn.
 - Tốc độ chậm
- Các loại bộ nhớ ngoài:
 - Bộ nhớ từ: đĩa cứng, đĩa mềm
 - Bộ nhớ quang: đĩa CD, DVD
 - Bộ nhớ bán dẫn: Flash disk, memory card

19

Minh họa

Floppy disk Compact disk Compact Flash Card USB Flash Drive

Hard disk

20

Hệ thống vào ra

- Chức năng: trao đổi thông tin giữa máy tính với thế giới bên ngoài.
- Các thao tác cơ bản:
 - Vào dữ liệu (Input)
 - Ra dữ liệu (Output)
- Các thành phần chính:
 - Các thiết bị ngoại vi (Peripheral Devices)
 - Các mô-đun vào-ra (IO Modules)

21

Cấu trúc cơ bản của hệ thống vào-ra

nội ghép với CPU và bộ nhớ chính

Cổng vào-ra

Cổng vào-ra

Mô-đun vào-ra

Cổng vào-ra

Thiết bị ngoại vi

Thiết bị ngoại vi

Thiết bị ngoại vi

22

Các thiết bị ngoại vi

- Chức năng: chuyển đổi dữ liệu giữa bên trong và bên ngoài máy tính.
- Các loại thiết bị ngoại vi cơ bản:
 - Thiết bị vào: bàn phím, chuột, máy quét,...
 - Thiết bị ra: màn hình, máy in, máy chiếu,...
 - Thiết bị nhớ: các ổ đĩa,...
 - Thiết bị truyền thông: MODEM,....

23

Thiết bị vào

Bàn phím (Keyboard)

Chuột (mouse)

Nhận dạng vân tay

Máy quét (Scanner)

Mic (Microphone)

24

Thiết bị ra

Màn hình (display, monitor) Máy in (Printer) Máy in kiêm phôtô

Loa (Speaker) Tai nghe (Headphone) Máy chiếu (Projector)

25

Thiết bị truyền thông

Card mạng (Network Interface Card) MODEM (Modulator-DEModulator)

MODEM là thiết bị biến đổi tín hiệu mạng tương tự thành thông tin số mã hóa và giải mã tín hiệu truyền nhận được

26

Mô-đun vào-ra

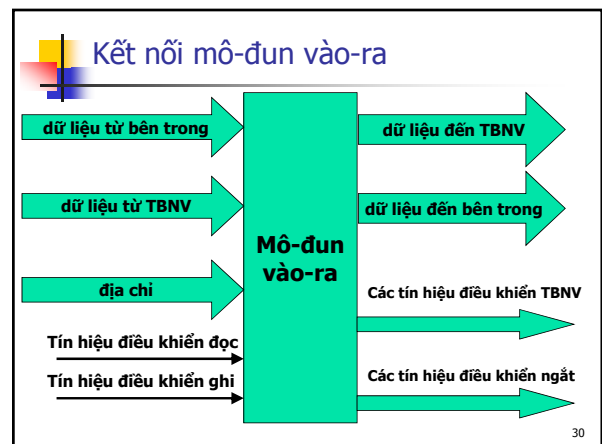
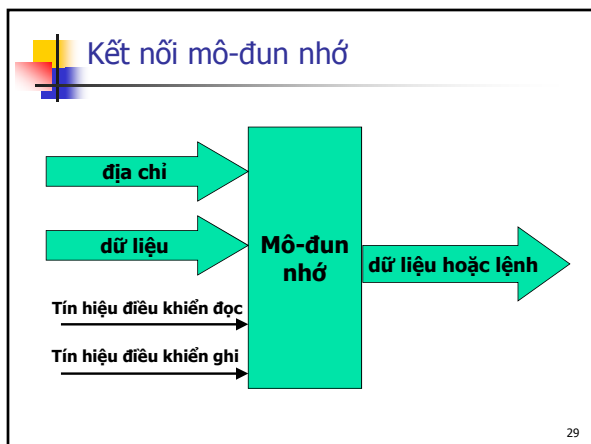
- Chức năng: nối ghép các thiết bị ngoại vi với máy tính.
- Mỗi mô-đun vào-ra có một hoặc một vài cổng vào-ra (I/O Port).
- Mỗi cổng vào-ra được đánh một địa chỉ xác định.
- Các thiết bị ngoại vi được kết nối và trao đổi dữ liệu với máy tính thông qua các cổng vào-ra.

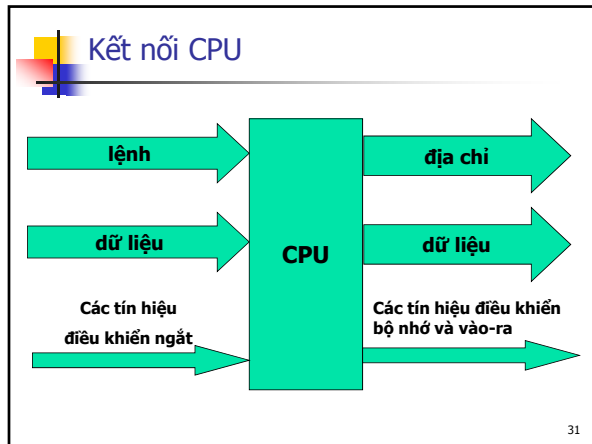
27

3.3. Liên kết hệ thống

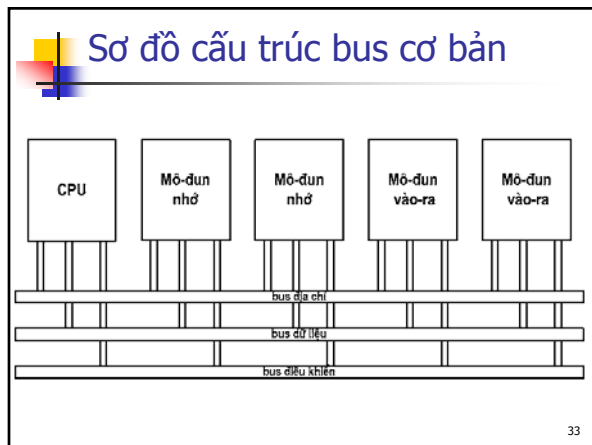
- 3.3.1. Luồng thông tin trong máy tính
 - Các mô-đun trong máy tính:
 - CPU
 - Mô-đun nhớ
 - Mô-đun vào-ra
 - cần được kết nối với nhau

28





- ### 3.3.2. Cấu trúc bus cơ bản
- Bus: tập hợp các đường kết nối dùng để vận chuyển thông tin giữa các mô-đun của máy tính với nhau.
 - Các bus chức năng:
 - Bus địa chỉ
 - Bus dữ liệu
 - Bus điều khiển
 - Độ rộng bus: là số đường dây của bus có thể truyền các bit thông tin đồng thời (chỉ dùng cho bus địa chỉ và bus dữ liệu).
- 32



- ### Bus địa chỉ
- Chức năng: vận chuyển địa chỉ để xác định ngăn nhớ hay cổng vào-ra.
 - Độ rộng bus địa chỉ: xác định dung lượng bộ nhớ cực đại của hệ thống.
 - Nếu độ rộng của bus địa chỉ là N bit:

$$A_{N-1}, A_{N-2}, \dots, A_2, A_1, A_0$$
 → có thể đánh địa chỉ tối đa cho 2^N ngăn nhớ
 - Ví dụ: bộ xử lý Pentium có bus địa chỉ 32 bit → không gian địa chỉ là 2^{32} byte = 4GB (đánh địa chỉ theo byte)
- 34

- ### Bus dữ liệu
- Chức năng:
 - vận chuyển lệnh từ bộ nhớ đến CPU
 - vận chuyển dữ liệu giữa CPU, các mô-đun nhớ và các mô-đun vào-ra với nhau
 - Độ rộng bus dữ liệu: xác định số bit dữ liệu có thể được trao đổi đồng thời.
 - M bit: $D_{M-1}, D_{M-2}, \dots, D_2, D_1, D_0$
 - M thường là 8, 16, 32, 64, 128 bit.
 - Ví dụ: các bộ xử lý Pentium có bus dữ liệu 64 bit.
- 35

- ### Bus điều khiển
- Chức năng: vận chuyển các tín hiệu điều khiển
 - Các loại tín hiệu điều khiển:
 - Các tín hiệu phát ra từ CPU để điều khiển mô-đun nhớ và mô-đun vào-ra.
 - Các tín hiệu từ mô-đun nhớ hay mô-đun vào-ra gửi đến yêu cầu CPU.
- 36

Đặc điểm của cấu trúc đơn bus

- Bus hệ thống chỉ phục vụ được một yêu cầu trao đổi dữ liệu tại một thời điểm.
- Bus hệ thống phải có tốc độ bằng tốc độ bus của mô-đun nhanh nhất trong hệ thống
- Bus hệ thống phụ thuộc vào cấu trúc bus (các tín hiệu) của bộ xử lý → các mô-đun nhớ và mô-đun vào-ra cũng phụ thuộc vào bộ xử lý.
- Vì vậy cần phải phân cấp bus → đa bus

37

Phân cấp bus trong máy tính

- Phân cấp bus cho các thành phần
 - Bus của bộ xử lý
 - Bus của bộ nhớ chính
 - Các bus vào-ra
- Phân cấp bus khác nhau về tốc độ
- Bus bộ nhớ chính và các bus vào ra không phụ thuộc vào bộ xử lý cụ thể.

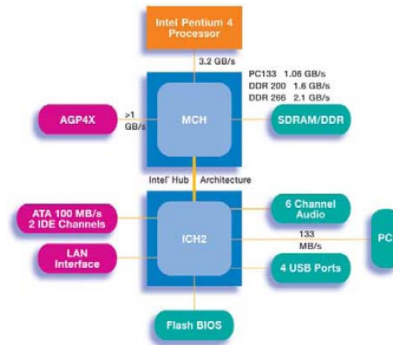
38

Các bus điển hình trong PC

- Bus của bộ xử lý (Front Side Bus-FSB): có tốc độ nhanh nhất
- Bus của bộ nhớ chính: nối ghép với các mô-đun RAM
- AGP bus (Accelerated Graphic Port): nối ghép card màn hình tăng tốc
- PCI bus (Peripheral Component Interconnect): nối ghép các thiết bị ngoại vi có tốc độ trao đổi dữ liệu nhanh.
- USB (Universal Serial Bus): bus nối tiếp đa năng.
- IDE (Integrated Device Electronics): bus kết nối với ổ đĩa cứng hoặc ổ đĩa CD, DVD

39

Máy tính Pentium 4 dùng chipset 845



40

Các kiểu bus

- Bus dành riêng (Dedicated):
 - Các đường địa chỉ và dữ liệu tách rời
 - Ưu điểm: điều khiển đơn giản
 - Nhược điểm: có nhiều đường kết nối
- Bus dồn kênh (Multiplexed):
 - Các đường dùng chung cho địa chỉ và dữ liệu
 - Có đường điều khiển để phân biệt có địa chỉ hay có dữ liệu
 - Ưu điểm: có ít đường dây
 - Nhược điểm: điều khiển phức tạp hơn, hiệu năng hạn chế

41

Phân xử bus

- Có nhiều mô-đun điều khiển bus như CPU và bộ điều khiển vào-ra
- Chỉ cho phép một mô-đun điều khiển bus tại một thời điểm
- Phân xử bus có thể:
 - tập trung: có 1 bộ điều khiển bus (Bus Controller / Arbiter) hoặc là 1 phần của CPU hay mạch tách rời.
 - phân tán: mỗi mô-đun có thể chiếm bus và có đường điều khiển đến tất cả các mô-đun khác.

42

3.4. Hoạt động của máy tính

- 3.4.1. Thực hiện chương trình
- 3.4.2. Ngắt
- 3.4.3. Hoạt động vào-ra

43

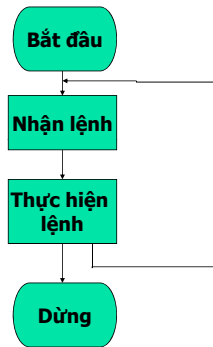
3.4.1. Thực hiện chương trình

- Là hoạt động cơ bản của máy tính
- Máy tính lặp đi lặp lại hai bước:
 - Nhận lệnh
 - Thực hiện lệnh

} chu trình lệnh
- Thực hiện chương trình bị dừng nếu thực hiện lệnh bị lỗi hoặc gặp lệnh dừng

44

Chu trình lệnh



45

Nhận lệnh

- Bắt đầu mỗi chu trình lệnh, CPU nhận lệnh từ bộ nhớ chính.
- Bộ đếm chương trình PC (Program Counter) của CPU giữ địa chỉ của lệnh sẽ được nhận.
- CPU nhận lệnh từ ngăn nhớ được trỏ bởi PC.
- Lệnh được nạp vào thanh ghi lệnh IR (Instruction Register).
- Sau khi lệnh được nhận vào, nội dung PC tự động tăng để trỏ sang lệnh kế tiếp.

46

Thực hiện lệnh

- Bộ xử lý giải mã lệnh đã được nhận và phát tín hiệu điều khiển thực hiện thao tác mà lệnh yêu cầu.
- Các kiểu thao tác của lệnh:
 - Trao đổi dữ liệu giữa CPU và bộ nhớ chính
 - Trao đổi dữ liệu giữa CPU và mô-đun vào-ra
 - Xử lý dữ liệu: thực hiện các phép toán số học hoặc phép toán logic với các dữ liệu.
 - Điều khiển rẽ nhánh
 - Kết hợp các thao tác trên.

47

3.4.2. Ngắt (Interrupt)

- Khái niệm chung về ngắt: Ngắt là cơ chế cho phép CPU tạm dừng chương trình đang thực hiện để chuyển sang thực hiện một chương trình khác, gọi là chương trình con phục vụ ngắt.
- Các loại ngắt:
 - Ngắt do lỗi khi thực hiện chương trình, ví dụ: tràn số, chia cho 0.
 - Ngắt do lỗi phần cứng, ví dụ: lỗi RAM
 - Ngắt do mô-đun vào-ra phát tín hiệu ngắt đến CPU yêu cầu trao đổi dữ liệu.

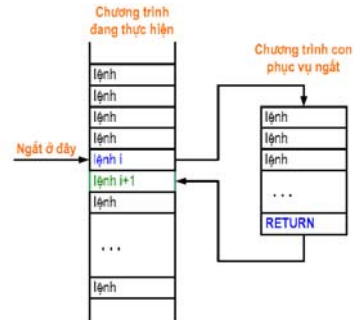
48

Hoạt động ngắt

- Sau khi hoàn thành mỗi một lệnh, bộ xử lý kiểm tra tín hiệu ngắt.
- Nếu không có ngắt → bộ xử lý nhận lệnh tiếp theo của chương trình hiện tại.
- Nếu có tín hiệu ngắt:
 - Tạm dừng chương trình đang thực hiện
 - Cắt ngữ cảnh (các thông tin liên quan đến chương trình bị ngắt)
 - Thiết lập PC trỏ đến chương trình con phục vụ ngắt
 - Chuyển sang thực hiện chương trình con phục vụ ngắt.
 - Cuối chương trình con phục vụ ngắt, khôi phục ngữ cảnh và tiếp tục chương trình đang bị tạm dừng.

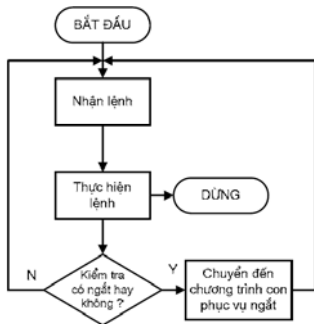
49

Hoạt động ngắt (tiếp)



50

Chu trình lệnh với ngắt



51

Xử lý với nhiều tín hiệu ngắt

- Xử lý ngắt tuần tự
 - Khi một ngắt đang được thực hiện, các ngắt khác sẽ bị cấm
 - Bộ xử lý sẽ bỏ qua các ngắt tiếp theo trong khi đang xử lý một ngắt.
 - Các ngắt vẫn đang đợi và được kiểm tra sau khi ngắt đầu tiên được xử lý xong
 - Các ngắt được thực hiện tuần tự
- Xử lý ngắt ưu tiên
 - Các ngắt được định nghĩa mức ưu tiên khác nhau
 - Ngắt có mức ưu tiên thấp hơn có thể bị ngắt bởi ngắt ưu tiên cao hơn → xảy ra ngắt lồng

52

3.4.3. Hoạt động vào-ra

- Hoạt động vào-ra: là hoạt động trao đổi dữ liệu giữa thiết bị ngoại vi với bên trong máy tính.
- Các kiểu hoạt động vào-ra:
 - CPU trao đổi dữ liệu với mô-đun vào-ra
 - Mô-đun vào-ra trao đổi dữ liệu trực tiếp với bộ nhớ chính.

53

3.5. Phần mềm máy tính (Software Computer)

- Thế nào là phần mềm máy tính?
 - "Phần mềm là thuật ngữ chung cho các bộ sưu tập có tổ chức dữ liệu và lệnh của máy tính, thường được phân thành hai loại chính: phần mềm hệ thống (system software) cung cấp các chức năng xác định cơ bản của máy tính và phần mềm ứng dụng (application software) được sử dụng bởi người dùng để hoàn thành những nhiệm vụ xác định."
<http://www.openprojects.org/software-definition.htm>
 - "Phần mềm là thuật ngữ tổng quát cho rất nhiều loại chương trình khác nhau được sử dụng để thao tác với máy tính và các thiết bị liên quan."
http://searchsoa.techtarget.com/sDefinition/0,sid26_gci213024_00.html
 - Phân biệt phần mềm (software) với phần cứng (hardware)

54

3.5.1. Dữ liệu và giải thuật

- Mỗi bài toán phải giải quyết gồm 2 phần:
 - phần dữ liệu
 - phần xử lý
- Phần **dữ liệu** liên quan đến thông tin của bài toán:
 - đầu vào: dữ liệu được cung cấp để xử lý
 - đầu ra: kết quả xử lý
- Phần **xử lý**: những thao tác phải được máy tính tiến hành nhằm đáp ứng yêu cầu của người dùng.

55

3.5.2. Chương trình và ngôn ngữ lập trình

- Thuật toán mới chỉ ra cách giải quyết một bài toán theo kiểu tư duy của con người. Để máy có thể hiểu và tiến hành xử lý được ta phải biến các bước thao tác thành các chỉ thị (statement) và biểu diễn trong dạng mà máy tính hiểu được. Quá trình này gọi là **lập trình**. Giải thuật được biểu diễn dưới dạng một tập các chỉ thị của một ngôn ngữ nào đó gọi là **chương trình**. Ngôn ngữ dùng để lập trình gọi là **ngôn ngữ lập trình** – ngôn ngữ dùng để trao đổi với máy tính, máy tính hiểu và thực thi nhiệm vụ đã chỉ ra.
- Tương tự với dữ liệu, máy tính không thể xử lý dữ liệu một cách hình thức như trong giải tích mà nó phải là những con số hay những giá trị cụ thể.

56

Chương trình

Chương trình = Cấu trúc dữ liệu + Giải thuật

Program = Data Structure + Algorithm

N. Wirth

57

Ngôn ngữ lập trình

- Có nhiều loại ngôn ngữ lập trình. Sự khác nhau giữa các loại liên quan đến mức độ phụ thuộc của chúng vào kiến trúc và hoạt động máy tính, phụ thuộc vào lớp/lĩnh vực ứng dụng. Có nhiều cách phân loại khác nhau và do đó các ngôn ngữ lập trình được phân thành các nhóm khác nhau. Người ta phân các ngôn ngữ theo một cách chung nhất thành 3 nhóm:
 - Ngôn ngữ máy
 - Hợp ngữ
 - Ngôn ngữ bậc cao

58

Ngôn ngữ máy

- Mỗi loại máy tính đều có ngôn ngữ máy riêng. Đó chính là loại ngôn ngữ duy nhất để viết chương trình mà máy tính hiểu trực tiếp và thực hiện được. Các chỉ thị (lệnh) của ngôn ngữ này viết bằng mã nhị phân hay mã hex-xa. Nó gắn chặt với kiến trúc phần cứng của máy và do vậy nó khai thác được các đặc điểm phần cứng. Tuy nhiên, nó lại không hoàn toàn thuận lợi cho người lập trình do tính khó nhớ của mã, tính thiếu cấu trúc,... Vì thế, để viết một ứng dụng bằng ngôn ngữ máy thì quả là việc không dễ, nhất là phải tiến hành các thay đổi, chỉnh sửa hay phát triển thêm về sau.

59

Hợp ngữ

- Hợp ngữ cho phép người lập trình sử dụng một số từ tiếng Anh viết tắt để thể hiện các câu lệnh thực hiện. Thí dụ để cộng nội dung của 2 thanh ghi AX và BX rồi ghi kết quả vào AX, ta có thể dùng câu lệnh hợp ngữ sau:
ADD AX, BX
- Một chương trình hợp ngữ phải được dịch ra ngôn ngữ máy nhờ chương trình hợp dịch trước khi máy tính có thể thực hiện.

60

Ngôn ngữ bậc cao

- FORTRAN, COBOL, Pascal, C/C++, VB, VC++, Delphi, Java, .NET,...
- Các chương trình viết trong ngôn ngữ này, trước khi để máy có thể thực thi cần phải chuyển đổi sang ngôn ngữ máy. Quá trình chuyển đổi đó gọi là quá trình dịch.

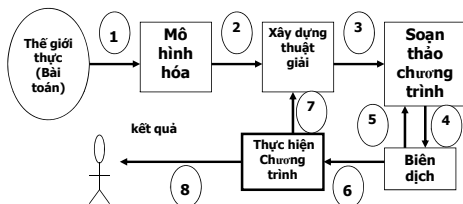
61

Hai phương thức dịch

- **Thông dịch (Interpreter):** Bộ thông dịch, đọc từng lệnh của chương trình nguồn, phân tích cú pháp của câu lệnh đó và nếu đúng thì thực hiện. Quá trình bắt đầu từ lệnh đầu tiên của chương trình đến lệnh cuối cùng nếu không có lỗi. Bộ thông dịch này giống như vai trò của 1 thông dịch viên (translator).
- **Biên dịch (Compiler):** Khác với thông dịch, trình biên dịch dịch toàn bộ chương trình nguồn sang ngôn ngữ đích. Với chương trình đích này, máy đã có thể hiểu được và biết cách thực thi. Quá trình biên dịch sẽ tạo ra chương trình đích chỉ khi các lệnh trong chương trình nguồn không có lỗi.

62

Quy trình giải quyết một bài toán trên máy tính



63

Quy trình giải quyết...(tiếp)

- **B1 Xác định bài toán:** Thuật ngữ mới cho bước này là xác định yêu cầu người dùng, người mong muốn có phần mềm để sử dụng.
- **B2 Phân tích bài toán:** Tìm hiểu nhiệm vụ (chức năng) mà phần mềm cần xây dựng phải có và các dữ liệu cần thiết. Qua đó xây dựng các giải pháp khả thi. Nói một cách ngắn gọn, bước này tìm hiểu hệ thống là gì? Và làm gì?
- **B3 Thiết kế hệ thống:** thực hiện thiết kế kiến trúc hệ thống, thiết kế các mô đun chương trình, thiết kế giao tiếp, thiết kế an toàn,... Như vậy, nhiệm vụ thiết kế mô đun chính là xây dựng giải thuật cho mô đun đó và cách diễn tả giải thuật.

64

Quy trình giải quyết...(tiếp)

- **B4 Xây dựng chương trình:** Viết code cho các mô đun theo ngôn ngữ lập trình đã xác định.
- **B5 Quay lại soạn thảo:** khi quá trình dịch phát hiện lỗi cú pháp trong chương trình nguồn
- **B6 Kiểm thử chương trình:** nhằm kiểm tra tính đúng đắn của từng mô đun và cả hệ thống trước khi bàn giao cho khách hàng.
- **B7 Xem lại giải thuật:** khi kết quả thực hiện không đúng (lỗi logic).
- **B8 Triển khai:** bước này gồm cả nhiệm vụ viết tài liệu phần mềm, hướng dẫn sử dụng và bảo trì phần mềm. Đây cũng là mục đích của phần mềm được yêu cầu và nhằm kéo dài vòng đời phần mềm (Software Life Cycle).

65

3.5.3. Phân loại phần mềm máy tính

- Theo quan điểm sử dụng chung:
 - **Phần mềm hệ thống:** Là phần mềm điều khiển hoạt động bên trong của máy tính và cung cấp môi trường giao tiếp giữa người dùng và máy tính nhằm khai thác hiệu quả phần cứng phục vụ cho nhu cầu sử dụng. Loại phần mềm này đòi hỏi tính ổn định, tính an toàn cao. Chẳng hạn các hệ điều hành máy đơn hay hệ điều hành mạng, các tiện ích hệ thống,...
 - **Phần mềm ứng dụng:** Là phần mềm dùng để giải quyết các vấn đề phục vụ cho các hoạt động khác nhau của con người như quản lý, kế toán, soạn thảo văn bản, trò chơi... Nhu cầu về phần mềm ứng dụng ngày càng tăng và đa dạng.

66

Phân loại phần mềm máy tính (tiếp)

- Theo đặc thù ứng dụng và môi trường:
 - Phần mềm thời gian thực (Real-time SW)
 - Phần mềm nghiệp vụ (Business SW)
 - Phần mềm tính toán KH&KT (Eng.&Scie. SW)
 - Phần mềm nhúng (Embedded SW)
 - Phần mềm trên Web (Web-based SW)
 - Phần mềm trí tuệ nhân tạo (IA SW)
 -

67

Virus máy tính ?

- Nó là cái gì ? Là một chương trình máy tính (do con người viết ra) có khả năng tự nhân bản và lây nhiễm và gây hại cho máy tính, không được sự cho phép của người dùng.
- Giống như virus sinh học, virus máy tính có khả năng nhân bản, lây lan nhanh chóng, có khả năng biến đổi thành các dạng khác, và nói chung là có hại.
- Virus máy tính thường phá hủy dữ liệu, làm sai lệch thông tin, ăn cắp thông tin cá nhân phục vụ những ý đồ xấu.
- Lây lan qua: đĩa mềm, CD, ổ USB, thư điện tử,...
- Thường nghe nói: malware, adware, worms, Trojan Horse

68

Phòng và diệt virus ?

- Sử dụng các phần mềm cảnh báo và diệt virus, phần mềm gián điệp như: Norton Anti Virus, Kaspersky, Bit Defender, BKAV,...
- Cảnh giác với các thư lạ, những thông tin mời mọc hấp dẫn trên mạng,...
- Cảnh giác với các ổ đĩa chứa dữ liệu không rõ ràng.
- Nói chung, vẫn khó tránh. Trong trường hợp bị lây nhiễm mà không tự khắc phục được, dữ liệu lại quan trọng thì nên tìm đến các chuyên gia.

69

Hỏi - đáp



70

Lời hay ý đẹp

"Đường đi không khó vì ngăn sông cách núi, mà khó vì lòng người ngại núi e sông"

Nguyễn Bá Học

"Nếu anh thấy một gia đình hạnh phúc, anh nên tin rằng ở trong gia đình có một người đàn bà biết quên mình"

René Bazin

71

Hướng dẫn tạo và sử dụng email miễn phí trên mạng

Nguyễn Hồng Phương

Email: phuongnh-fit@mail.hut.edu.vn

Website: <http://is.hut.edu.vn/~phuongnh>

Bộ môn Hệ thống thông tin
Viện Công nghệ thông tin và Truyền thông
Đại học Bách Khoa Hà Nội

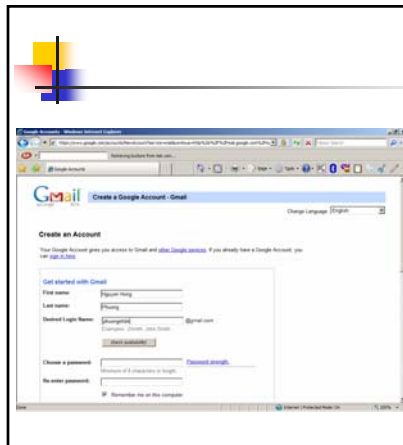
1

Tạo gmail (của hãng Google)

- Dùng trình duyệt IE, FireFox,...
- Gõ vào địa chỉ URL: <http://mail.google.com>



2



- Điền các thông tin: First name, Last name, Desired Login Name (vd: phuongnh@gmail.com) và nhấn nút "check availability" để kiểm tra xem tên truy nhập phuongnh đã có chưa.

3

- Nhập password 2 lần và nhập các thông tin khác.



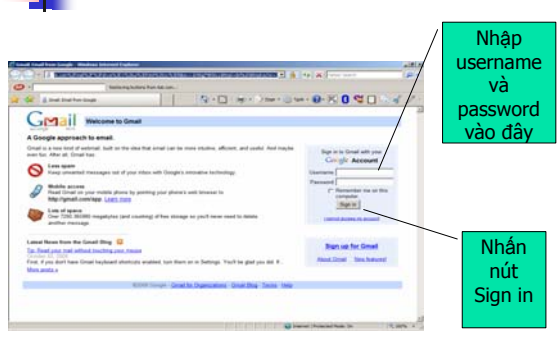
4

Sau đó, kích vào nút I accept. Create my account



5

Sau khi tạo thành công, chúng ta có thể vào web mail



6

Tạo yahoo mail (của hãng Yahoo)

- Vào trang: www.mail.yahoo.com
- Làm tương tự như với gmail đã hướng dẫn ở trên.

7

Web mail vs. POP3

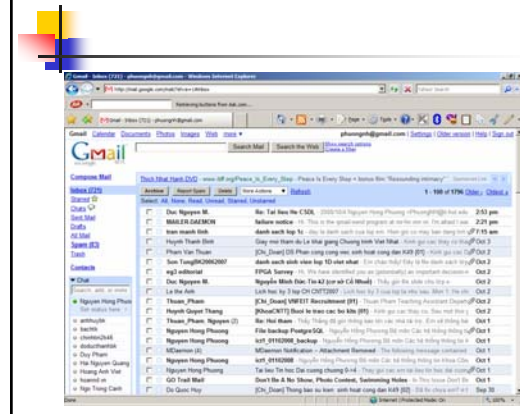
- Web mail vs. POP3

8

Sử dụng gmail

- Tạo thư: Compose
- Xem thư đến: Inbox
- Xem thư đã gửi: Sent mail
- Thư rác: Spam
- Thùng rác: Trash

9



10

Chương 4: Mạng máy tính

Nguyễn Hồng Phương

Email: phuongnh-fit@mail.hut.edu.vn

Website: <http://is.hut.edu.vn/~phuongnh>

Bộ môn Hệ thống thông tin

Viện Công nghệ thông tin và Truyền thông

Đại học Bách Khoa Hà Nội

1

Nội dung chương này

- 4.1. Lịch sử phát triển, khái niệm mạng máy tính
- 4.2. Phân loại mạng máy tính
- 4.3. Các thành phần cơ bản của một mạng máy tính
- 4.4. Mạng Internet

2

4.1. Lịch sử phát triển của mạng máy tính

- Máy tính ra đời từ những năm 1950. Đến đầu những năm 1960 mạng máy tính bắt đầu xuất hiện. Lúc đầu mạng có dạng là một máy tính lớn nối với nhiều trạm cuối (*terminal*). Đến đầu những năm 1970 mạng máy tính là các máy tính độc lập được nối với nhau. Qui mô và mức độ phức tạp của mạng ngày càng tăng.
- Hiện nay mạng máy tính phát triển rất mạnh ở mọi lĩnh vực mọi nơi. Ngày càng hiếm các máy tính đơn lẻ, không nối mạng. Ngay các máy tính cá nhân ở gia đình cũng được kết nối Internet qua đường điện thoại. Mạng trở thành một yếu tố không thể thiếu của công nghệ thông tin nói riêng, cũng như đời sống nói chung.

3

4.1. Khái niệm mạng máy tính

- **Mạng máy tính** hay **mạng** (*computer network, network*) là một tập hợp gồm nhiều máy tính hoặc thiết bị xử lý thông tin được kết nối với nhau qua các đường truyền và có sự trao đổi dữ liệu với nhau.
- Nhờ có mạng máy tính, thông tin từ một máy tính có thể được truyền sang máy tính khác. Có thể ví mạng máy tính như một hệ thống giao thông vận tải mà hàng hoá trên mạng là dữ liệu, máy tính là nhà máy lưu trữ xử lý dữ liệu, hệ thống đường truyền như là hệ thống đường sá giao thông.
- Ví dụ về mạng máy tính: mạng tại TTMT-Khoa CNTT, mạng của Tổng cục thuế, mạng Internet,...

4

4.2. Phân loại mạng máy tính

- Theo mối quan hệ giữa các máy trong mạng
 - **Mạng bình đẳng** (*peer-to-peer*) các máy có quan hệ ngang hàng, một máy có thể yêu cầu một máy khác phục vụ.
 - **Mạng khách/chủ** (*client/server*). Một số máy là server (máy chủ) chuyên phục vụ các máy khác gọi là máy khách (*client*) hay máy trạm (*workstation*) khi có yêu cầu. Các dịch vụ có thể là cung cấp thông tin, tính toán hay các dịch vụ Internet.

5

4.2. Phân loại mạng máy tính (tiếp)

- Theo qui mô địa lý
 - **LAN** (*Local Area Network*) mạng cục bộ ở trong phạm vi nhỏ, ví dụ bán kính 500m, số lượng máy tính không quá nhiều, mạng không quá phức tạp.
 - **WAN** (*Wide Area Network*) mạng diện rộng, các máy tính có thể ở các thành phố khác nhau. Bán kính có thể 100-200 km. Ví dụ mạng của Tổng cục thuế.
 - **GAN** (*Global Area Network*) mạng toàn cầu, máy tính ở nhiều nước khác nhau. Thường mạng toàn cầu là kết hợp của nhiều mạng con. Ví dụ mạng Internet.

6

4.3. Các thành phần cơ bản của mạng máy tính

Các máy tính

Vi mạng (Network Interface Card-NIC)

Đường truyền vật lý: hữu tuyến, vô tuyến

Các thiết bị kết nối mạng: HUB, SWITCH, ROUTER

7

4.3. Các thành phần cơ bản của mạng máy tính

Các thiết bị đầu cuối (terminal) khác: máy photo, máy in, scanner, camera mạng,...

Các phụ kiện mạng: ổ cắm, giắc cắm,...

8

4.3. Các thành phần cơ bản của mạng máy tính

- Hệ điều hành mạng: Hệ điều hành mạng là một phần mềm điều khiển sự hoạt động của mạng.
- Các phần mềm mạng cho máy tính: Hiện nay nói chung các hệ điều hành đều sẵn có khả năng kết nối mạng. Trong trường hợp hệ điều hành của máy tính không có sẵn khả năng kết nối mạng thì các phần mềm này là cần thiết.
- Các ứng dụng trên mạng: Ví dụ như Email, hệ quản trị cơ sở dữ liệu.

9

Kiến trúc mạng máy tính

- **Kiến trúc mạng máy tính** (network architecture) thể hiện cách kết nối máy tính với nhau và qui ước truyền dữ liệu giữa các máy tính như thế nào. Cách nối các máy tính với nhau gọi là **hình trạng (topology)** của mạng. Tập các qui ước truyền thông gọi là **giao thức (protocol)**.
- Có hai kiểu nối mạng chủ yếu là **điểm-điểm** (point to point) và **quảng bá** (broadcast). Trong kiểu điểm-điểm các đường truyền nối các nút thành từng cặp. Như vậy một nút sẽ gửi dữ liệu đến nút lân cận nó (nút được nối trực tiếp với nó). Nút lân cận sẽ chuyển tiếp dữ liệu như vậy cho đến khi dữ liệu đến đích.

10

Kiểu nối mạng điểm - điểm

- Kiểu nối mạng điểm- điểm có ba dạng chính là : hình sao (*star*), chu trình (*loop*), cây (*tree*) và đầy đủ (*complete*).

11

Kiểu nối mạng quảng bá

- Trong kiểu quảng bá các nút nối vào đường truyền chung. Như vậy khi một nút gửi dữ liệu các nút còn lại đều nhận được. Do đó dữ liệu gửi đi cần có địa chỉ đích. Khi một nút nhận được dữ liệu nó sẽ kiểm tra địa chỉ đích xem có phải gửi cho mình không.

12

4.4. Mạng Internet - Khái niệm

- *Internet* là một mạng máy tính có qui mô toàn cầu (GAN), gồm rất nhiều mạng con và máy tính nối với nhau bằng nhiều loại phương tiện truyền. Internet không thuộc sở hữu của ai cả. Chỉ có các uỷ ban điều phối và kỹ thuật giúp điều hành Internet.
- Ban đầu là mạng của Bộ Quốc phòng Mỹ (DoD) dùng để đảm bảo liên lạc giữa các đơn vị quân đội. Sau đó phát triển thành mạng cho các trường đại học và viện nghiên cứu. Cuối cùng mạng có qui mô toàn cầu và trở thành mạng Internet.

13

Các dịch vụ chính của Internet

- Ta có thể dùng Internet để thực hiện nhiều dịch vụ mạng. Các dịch vụ thông dụng nhất trên Internet hiện nay là:
 - Truyền thông tin (FTP, File Transfer Protocol)
 - Truy nhập máy tính từ xa (telnet)
 - Web (WWW) để tìm kiếm và khai thác thông tin trên mạng
 - Thư điện tử (E-mail)
 - Tán gẫu (Chat) trên mạng...

14

Lợi ích của Internet

- Trong thời đại của công nghệ thông tin hiện nay Internet có nhiều lợi ích như truyền tin, phổ biến tin, thu thập tin, trao đổi tin một cách nhanh chóng thuận tiện rẻ tiền hơn so với các phương tiện khác như điện thoại, fax. Internet ảnh hưởng đến toàn bộ thế giới đến mọi ngành, mọi lĩnh vực xã hội. Hiện nay Internet thành một yếu tố quan trọng không thiếu được trong thời đại hiện nay, có mặt ở mọi nơi, mọi lĩnh vực, mọi ngành.

15

Làm sao để có được các dịch vụ Internet

- Để kết nối được Internet ta cần :
 - Máy tính có Modem (Dial-up, ADSL) hoặc card mạng.
 - Có thuê bao kết nối với Internet: qua mạng, qua đường điện thoại, đường thuê riêng của bưu điện. Thông thường hiện nay kết nối qua điện thoại hoặc qua ADSL
 - Có tài khoản Internet ở trên mạng hay ở một nhà cung cấp dịch vụ Internet (Internet Service Provider, ISP), ví dụ như VNN, FPT.
 - Có phần mềm Internet thông dụng như Web browser để xem trang web, ví dụ IE, FireFox , phần mềm để xem thư hay chat như Outlook, Messenger.

16

Cần cảnh giác với:

- Virus
- Các thông tin độc hại trên mạng

17

Hỏi - đáp



18



Lời hay ý đẹp

"Hạnh phúc là đấu tranh"

Karl Mark

Học trong quá khứ, sống trong hiện tại, và chuẩn bị cho tương lai.

Learn from the past, live in the present, and prepare for the future

Dự tính điều xấu nhất, hy vọng điều tốt nhất

Plan for the worst, hope for the best

Hãy xin sẽ được, hãy tìm sẽ gặp, hãy gõ cửa sẽ mở.

Ask you will receive, seek you will find, knock and the door will be opened for you.

Hành động có giá trị hơn nói

Action speaks louder than word

Chương 5: Hệ điều hành

Nguyễn Hồng Phương
Email: phuongnh-fit@mail.hut.edu.vn
Website: <http://is.hut.edu.vn/~phuongnh>
Bộ môn Hệ thống thông tin
Viện Công nghệ thông tin và Truyền thông
Đại học Bách Khoa Hà Nội

1

Nội dung chương này

- 5.1. Các khái niệm cơ bản
 - 5.1.1. Khái niệm hệ điều hành
 - 5.1.2. Tập (file)
 - 5.1.3. Quản lý tập của hệ điều hành
- 5.2. Một số hệ điều hành
 - 5.2.1. Hệ điều hành MS-DOS
 - 5.2.2. Hệ điều hành Windows
 - 5.2.3. Hệ điều hành Linux
- 5.3. Hệ lệnh của hệ điều hành

2

Nội dung chương này

- 5.4. Hệ điều hành Windows
 - 5.4.1. Sự ra đời và phát triển
 - 5.4.2. Khởi động và thoát khỏi Windows
 - 5.4.3. Một số thuật ngữ và thao tác thường sử dụng
 - 5.4.4. Cấu hình Windows (Control Panel)
 - 5.4.5. Windows Explorer
 - 5.4.6. Gọi thực hiện chương trình
 - 5.4.7. Chế độ Command Prompt
 - 5.4.8. Recycle Bin

3

5.1. Các khái niệm cơ bản

- 5.1.1. Khái niệm hệ điều hành
- 5.1.2. Tập (file)
- 5.1.3. Quản lý tập của hệ điều hành

4

5.1.1. Khái niệm hệ điều hành

- Hệ điều hành là hệ thống chương trình đảm bảo quản lý tài nguyên của hệ thống tính toán và cung cấp các dịch vụ cho người sử dụng.
- Thông thường trong các máy tính hiện nay, hệ điều hành được cài đặt trên đĩa.
- Nhiệm vụ cụ thể của hệ điều hành là:
 - Khởi động máy tính, tạo môi trường giao tiếp cho người sử dụng.
 - Tự động điều khiển và kiểm soát hoạt động của các thiết bị (ổ đĩa, bàn phím, màn hình, máy in,...).
 - Quản lý việc cấp phát tài nguyên của máy tính như bộ xử lý trung tâm, bộ nhớ, các thiết bị vào ra...
 - Quản lý các chương trình đang thực hiện trên máy tính.
 - Thực hiện giao tiếp với người sử dụng để nhận lệnh và thực hiện lệnh.

5

5.1.1. Khái niệm hệ điều hành

- Hệ điều hành là phần mềm hệ thống, nên phụ thuộc vào cấu trúc của máy tính. Mỗi loại máy tính có hệ điều hành khác nhau. Ví dụ:
 - Máy tính lớn IBM360 có hệ điều hành là DOS, TOS.
 - Máy tính lớn EC-1022 có hệ điều hành là OC-EC.
 - Máy tính cá nhân PC-IBM có hệ điều hành MS-DOS.
 - Mạng máy tính có các hệ điều hành mạng NETWARE, UNIX, WINDOWS-NT...
 - ...

6

5.1.2. Tệp (file)

- Tệp là tập hợp các dữ liệu có liên quan với nhau và được tổ chức theo 1 cấu trúc nào đó, thường được lưu trữ bên ngoài máy tính.
- Nội dung của tệp có thể là chương trình, dữ liệu, văn bản,...
- Mỗi tệp tin được lưu lên đĩa với một tên riêng biệt. Mỗi hệ điều hành có quy ước đặt tên khác nhau, tên tệp tin thường có 2 phần:
 - phần tên (name): bắt buộc phải có của một tệp tin
 - phần mở rộng (extension): có thể có hoặc không.

7

5.1.2. Tệp (file) (tiếp)

- Phần tên: Bao gồm các ký tự chữ từ A đến Z, các chữ số từ 0 đến 9, các ký tự khác như #, \$, %, ~, ^, @, (,), !, _, khoảng trắng. Phần tên do người tạo ra tệp tin đặt. Với MS-DOS phần tên có tối đa là 8 ký tự, với Windows phần tên có thể đặt tối đa 128 ký tự.
- Phần mở rộng: thường dùng 3 ký tự trong các ký tự nêu trên. Thông thường phần mở rộng do chương trình ứng dụng tạo ra tệp tin tự đặt.
- Giữa phần tên và phần mở rộng có một dấu chấm (.) ngăn cách.

8

Kiểu của file

- Ta có thể căn cứ vào phần mở rộng để xác định kiểu của file:
 - COM, EXE : Các file khả thi chạy trực tiếp được trên hệ điều hành.
 - TXT, DOC, ... : Các file văn bản.
 - PAS, BAS, ... : Các file chương trình PASCAL, DELPHI, BASIC, ...
 - WK1, XLS, ... : Các file chương trình bảng tính LOTUS, EXCEL ...
 - BMP, GIF, JPG, ... : Các file hình ảnh.
 - MP3, DAT, WMA, ... : Các file âm thanh, video.

9

Kí hiệu đại diện (wildcard)

- Để chỉ một nhóm các tệp tin, ta có thể sử dụng hai ký hiệu đại diện:
 - Dấu ? dùng để đại diện cho một ký tự bất kỳ trong tên tệp tin tại vị trí nó xuất hiện.
 - Dấu * dùng để đại diện cho một chuỗi ký tự bất kỳ trong tên tệp tin từ vị trí nó xuất hiện.
- Ví dụ:
 - Bai?.doc Bai1.doc, Bai6.doc, Baiq.doc, ...
 - Bai*.doc Bai.doc, Bai6.doc, Bai12.doc, BaiTap.doc, ...
 - BaiTap.* BaiTap.doc, BaiTap.xls, BaiTap.ppt, BaiTap.dbf, ...
- Lưu ý: Nên đặt tên mang tính gợi nhớ

10

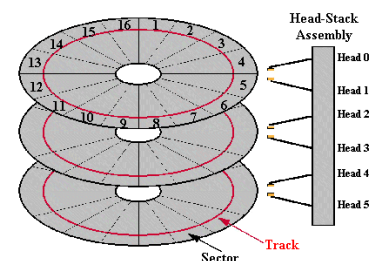
5.1.3. Quản lý tệp của hệ điều hành

- Cấu trúc đĩa từ
 - Hệ thống đĩa từ gồm nhiều mặt (side) gắn số hiệu là 0, 1,... Về mặt logic mỗi mặt đĩa có một đầu ghi/ đọc (header), đôi khi người ta còn đồng nhất 2 khái niệm này.
 - Mỗi mặt chia thành các rãnh (track - các đường tròn đồng tâm). Các rãnh được đánh số từ ngoài vào trong bắt đầu từ 0.
 - Mỗi rãnh chia thành các cung (Sector), mỗi sector thông thường có dung lượng 512 byte.
 - Một từ trụ (cylinder) gồm các rãnh có cùng bán kính nằm trên các mặt đĩa khác nhau.

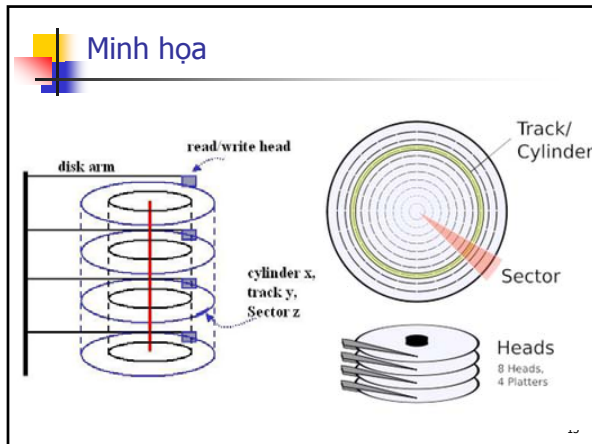
11

Minh họa

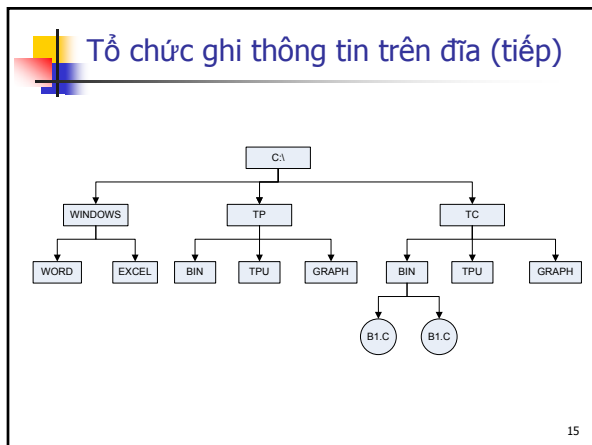
Drive Physical and Logical Organization



12



- ### Tổ chức ghi thông tin trên đĩa
- Thông tin lưu trữ trên đĩa dưới dạng các tệp. Mỗi tệp chiếm 1 hoặc nhiều sectors tùy dung lượng tệp.
 - Để thuận lợi cho việc quản lý tệp, hệ điều hành cho phép chia đĩa thành các vùng, mỗi vùng chia thành các vùng con,... Mỗi vùng có 1 vùng con riêng để lưu trữ thông tin về vùng đó, vùng con này được gọi là thư mục (Directory). Tệp được lưu trữ ở các vùng, vì vậy ta có thể thấy tổ chức lưu trữ này có dạng cây (Tree). Ví dụ:



- ### Thư mục
- Thư mục là nơi lưu giữ các tệp tin theo một chủ đề nào đó theo ý người sử dụng. Đây là biện pháp giúp ta quản lý được tệp tin, dễ dàng tìm kiếm chúng khi cần truy xuất. Các tệp tin có liên quan với nhau có thể được xếp trong cùng một thư mục. Sau đây là biểu tượng của thư mục hay còn gọi là Folder trong Windows
-

- ### Thư mục (tiếp)
- Trên mỗi đĩa có một thư mục chung gọi là thư mục gốc. Thư mục gốc không có tên riêng và được ký hiệu là \ (dấu xỏ phải: backslash). Dưới mỗi thư mục gốc có các tệp tin trực thuộc và các thư mục con. Trong các thư mục con cũng có các tệp tin trực thuộc và thư mục con của nó. Thư mục chứa thư mục con gọi là thư mục cha.
 - Thư mục đang làm việc gọi là thư mục hiện hành.
 - Tên của thư mục tuân thủ theo cách đặt tên của tệp tin.

- ### Cách xác định tên đầy đủ của tệp
- Tên tệp đầy đủ gồm nơi lưu trữ tệp - đường dẫn từ gốc đến tệp (Path) và tên tệp. Đường dẫn được chỉ ra nhánh cây thư mục chứa tệp, trong đó sử dụng ký hiệu "\" ngăn cách tên các thư mục.
 - Ví dụ :
 - C:\TC\BIN\B1.C

File hệ thống

- Hệ điều hành được phân chia thành các phần, phù hợp với các chức năng riêng của công việc. Những phần này được lưu trên đĩa dưới dạng các tệp (File).
- Ví dụ: Hệ điều hành MS-DOS gồm tập các tệp, trong đó có 3 tệp cơ bản:
 - MSDOS.SYS - tệp.
 - IO.SYS - tệp điều khiển vào ra.
 - COMMAND.COM - tệp lệnh.

19

5.2. Một số hệ điều hành

- Hiện nay có nhiều hệ điều hành khác nhau như MS-DOS, UNIX, LINUX, Windows 95, Windows 98, Windows 2000, Windows XP, Windows 2003, và Windows VISTA là một sản phẩm mới của MicroSoft. Mỗi hệ điều hành có các đặc trưng khác nhau, tuy vậy trong mỗi hệ điều hành có thể tích hợp nhiều hình thái giao tiếp người- máy khác nhau: dòng lệnh, bảng chọn, biểu tượng,...

20

5.2. Một số hệ điều hành (tiếp)

- 5.2.1. Hệ điều hành MS-DOS
- 5.2.2. Hệ điều hành Windows
- 5.2.3. Hệ điều hành Linux

21

5.2.1. Hệ điều hành MS-DOS

- Hình thức giao tiếp: văn bản - text
- Thực hiện các chức năng bằng câu lệnh



22

5.2.2. Hệ điều hành Windows

- Hình thức giao tiếp: đồ họa (bảng chọn, biểu tượng)
- Thực hiện chức năng thông qua giao diện đồ họa hoặc phím tắt.



23

5.2.3. Hệ điều hành Linux

- Hình thức giao tiếp: dòng lệnh, đồ họa
- Thực hiện các chức năng: câu lệnh, giao diện đồ họa



24

5.3. Hệ lệnh của hệ điều hành

- Thao tác với tệp: Sao chép, di chuyển, xoá, đổi tên, xem nội dung tệp
- Thao tác với thư mục: tạo, xoá, sao chép
- Thao tác với đĩa: tạo khuôn (Format), sao chép đĩa

25

5.4. Hệ điều hành Windows

- 5.4.1. Sự ra đời và phát triển
- 5.4.2. Khởi động và thoát khỏi Windows
- 5.4.3. Một số thuật ngữ và thao tác thường sử dụng
- 5.4.4. Cấu hình Windows (Control Panel)
- 5.4.5. Windows Explorer

26

5.4.1. Sự ra đời và phát triển

- Windows là một bộ chương trình do hãng Microsoft sản xuất. Từ version 3.0 ra đời vào tháng 5 năm 1990 đến nay, Microsoft đã không ngừng cải tiến làm cho môi trường này ngày càng được hoàn thiện.

27

5.4.1. Sự ra đời và phát triển (tiếp)

- Windows 95
- Windows 98, Windows Me
- Windows NT 4.0, Windows 2000, Windows XP, Windows 2003
- Windows Vista

28

5.4.2. Khởi động và thoát khỏi Windows XP

- Khởi động Windows XP:
 - Tự khởi động khi bật máy
 - Người dùng phải đăng nhập (login): nhập username và password vào
 - Mỗi người dùng có một tập hợp thông tin thiết lập riêng gọi là user profile
- Thoát khỏi Windows XP:
 - Đóng tất cả các cửa sổ, chương trình đang mở
 - Nhấn Alt+F4 hoặc chọn Start/ Turn off computer

29

5.4.3. Một số thuật ngữ và thao tác thường sử dụng

- Biểu tượng (icon):
 - là các hình vẽ nhỏ đặc trưng cho một đối tượng nào đó của Windows hoặc của các ứng dụng chạy trong môi trường Windows.
 - Phía dưới biểu tượng là tên biểu tượng. Tên này mang một ý nghĩa nhất định, thông thường nó diễn giải cho chức năng được gán cho biểu tượng (ví dụ nó mang tên của 1 trình ứng dụng).

30

Biểu tượng

Đây là biểu tượng

31

Cửa sổ (Windows)

- Cửa sổ là khung giao tiếp đồ họa của 1 ứng dụng hoặc 1 lệnh.
- Bố cục của 1 cửa sổ : gồm thanh tiêu đề, thanh thực đơn, 1 số thành phần khác phụ thuộc vào loại cửa sổ,...
- Các hộp giao tiếp
- Các thao tác trên một cửa sổ: di chuyển cửa sổ, thay đổi kích thước cửa sổ, phóng to, thu nhỏ, phục hồi kích thước cửa sổ, chuyển đổi giữa các cửa sổ, đóng cửa sổ...

32

Cửa sổ (Windows)

33

Hộp hội thoại (Dialogue box)

34

Sử dụng chuột trong windows

- Chuột là thiết bị không thể thiếu khi làm việc trong môi trường Windows XP. Con trỏ chuột (mouse pointer) cho biết vị trí tác động của chuột trên màn hình. Hình dáng của con trỏ chuột trên màn hình thay đổi theo chức năng và chế độ làm việc của ứng dụng. Khi làm việc với thiết bị chuột bạn thường sử dụng các thao tác cơ bản sau :
 - **Point:** trỏ chuột trên mặt phẳng mà không nhấn nút nào cả.
 - **Click:** nhấn nhanh và thả nút chuột trái. Dùng để lựa chọn thông số, đối tượng hoặc câu lệnh.
 - **Double Click (D Click) :** nhấn nhanh nút chuột trái hai lần liên tiếp. Dùng để khởi động một chương trình ứng dụng hoặc mở thư mục/ tập tin.

35

Sử dụng chuột trong windows (tiếp)

- **Drag** (kéo thả): nhấn và giữ nút chuột trái khi di chuyển đến nơi khác và buông ra. Dùng để chọn một khối văn bản, để di chuyển một đối tượng trên màn hình hoặc mở rộng kích thước của cửa sổ...
- **Right Click (R_Click):** nhấn nhanh và thả nút chuột phải. Dùng mở menu tương ứng với đối tượng để chọn các lệnh thao tác trên đối tượng đó.
- **Chú ý:**
 - Đa số chuột hiện nay có bánh xe trượt hoặc nút đẩy ở giữa dùng để cuộn màn hình làm việc được nhanh hơn và thuận tiện hơn.
 - Trong Windows các thao tác được thực hiện mặc nhiên với nút chuột trái, vì vậy để tránh lặp lại, khi nói **Click** (nhấn chuột) hoặc **D_Click** (nhấn đúp chuột) thì được ngầm hiểu đó là nút chuột trái. Khi nào cần thao tác với nút chuột phải sẽ mô tả rõ ràng.

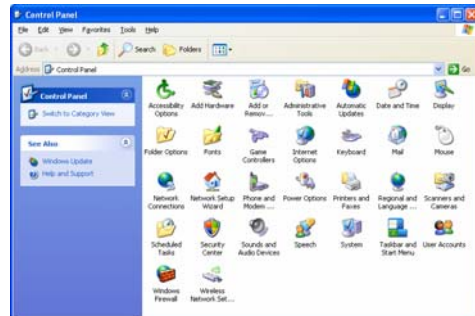
36

5.4.4. Cấu hình Windows (Control Panel)

- Giới thiệu về Control Panel
 - Control Panel là một chương trình cho phép người sử dụng xem và chỉnh sửa các tham số của hệ thống máy tính như dạng hiện của dữ liệu ngày tháng, dữ liệu số, thiết lập hoặc thay đổi cấu hình cho phù hợp với công việc hoặc sở thích của người dùng, cài đặt phần cứng, phần mềm.
 - Khởi động: Start / Settings / Control Panel

37

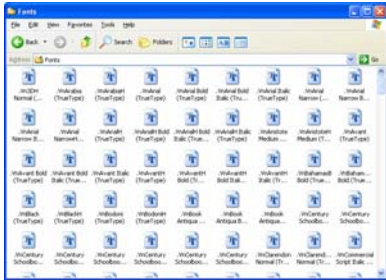
Cửa sổ làm việc của Control Panel



38

Cài đặt và loại bỏ font chữ

- Để cài đặt thêm những Font chữ khác hoặc loại bỏ các Font chữ, ta chọn chương trình **Fonts**



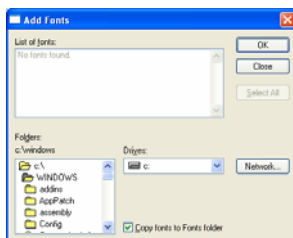
39

Cài đặt và loại bỏ font chữ (tiếp)

- **Loại bỏ font chữ.** Từ cửa sổ Fonts
 - Chọn những Font cần loại bỏ
 - Chọn File/ Delete (hoặc nhấn phím Delete).
- **Thêm font chữ mới** Từ cửa sổ Fonts, chọn lệnh File/Install New Font, xuất hiện hộp thoại Add Fonts. Trong hộp thoại này **chỉ ra nơi chứa các Font nguồn muốn thêm bằng cách chọn tên ổ đĩa** chứa các tập tin Font chữ, sau đó chọn các tên Font và Click OK.

40

Thêm font



41

Thay đổi dạng hiện màn hình

- Chọn lệnh **Start/ Settings/ Control Panel/ Display** hoặc **Right Click trên màn hình** nền (Desktop), chọn Properties. Xuất hiện cửa sổ Display Properties với các thành phần như sau:
- **Desktop:** Chọn ảnh nền cho Desktop bằng cách Click chọn các ảnh nền có sẵn hoặc Click vào nút Browse để chọn tập tin ảnh không có trong danh sách những ảnh có sẵn.
- **Screen Saver:** xác lập màn hình nghỉ
- **Settings:** Thay đổi chế độ màu và độ phân giải của màn hình. - Chế độ màu càng cao thì hình ảnh càng đẹp và rõ nét. Các chế độ màu: 64.000 màu (16 bits), 16 triệu màu (24 bits). Chế độ màu trên mỗi máy tính có thể khác nhau tùy thuộc vào dung lượng bộ nhớ của card màn hình. Độ phân giải càng lớn thì màn hình càng hiển thị được nhiều thông tin.

42

Thay đổi dạng hiện màn hình (tiếp)



43

Cài đặt và loại bỏ chương trình

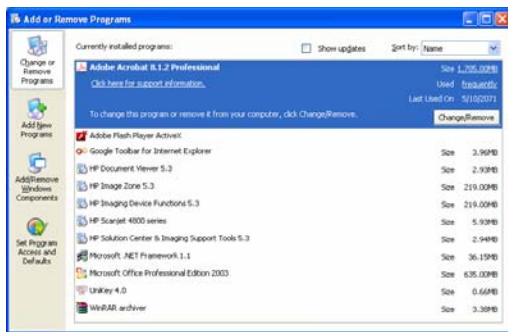
- Để cài đặt các chương trình mới hoặc loại bỏ các chương trình không còn sử dụng bạn nhấn đúp chuột vào biểu tượng **Add or Remove Programs** trong cửa sổ Control Panel xuất hiện hộp thoại và thao tác theo chỉ dẫn



Add or Remove Programs

44

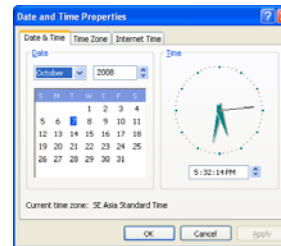
Cài đặt và loại bỏ chương trình (tiếp)



45

Cấu hình ngày giờ cho hệ thống

- Bạn có thể thay đổi ngày giờ của hệ thống bằng cách **D** Click lên biểu tượng đồng hồ trên thanh Taskbar hoặc vào Control Panel, chọn nhóm **Date/Time - Date & Time**: thay đổi ngày, tháng, năm, giờ, phút, giây.



46

Thay đổi thuộc tính của chuột



- Lệnh Start/ Settings/ Control Panel, rồi chọn biểu tượng Mouse

- Lớp Buttons:** thay đổi phím trái và phím chuột phải (thuận tay trái hay phải) và tốc độ nhấp đúp chuột.
- Lớp Pointers:** cho phép chọn hình dạng trỏ chuột trong các trạng thái làm việc.

47

Thay đổi thuộc tính của bàn phím



- Lệnh Start/ Settings/ Control Panel, rồi chọn biểu tượng Keyboard

- Repeat delay:** thay đổi thời gian trễ cho phím.
- Repeat rate:** thay đổi tốc độ lặp lại khi nhấn phím

48

Thay đổi thuộc tính vùng (Regional Settings)

- Lệnh Start / Settings / Control Panel / Regional and Language Options
- **Lớp Regional Options:** Thay đổi thuộc tính vùng địa lý, sau đó sẽ kéo theo sự thay đổi các thuộc tính của Windows. Click chọn **Customize**, cửa sổ Customize để thay đổi qui ước về dạng số, tiền tệ, thời gian, ngày tháng.



49

Số, tiền tệ, ngày

- **Number:** Thay đổi định dạng số, cho phép định dạng việc hiển thị
 - Decimal symbol: Thay đổi ký hiệu phân cách hàng thập phân.
 - No. of digits after decimal: Thay đổi số các số lẻ ở phần thập phân.
 - Digit grouping symbol: Thay đổi ký hiệu phân nhóm hàng ngàn.
 - Digit grouping: Thay đổi số ký số trong một nhóm (3 số / 4 số/ ...)
 - Negative sign symbol: Thay đổi ký hiệu của số âm.
 - Negative number format: Thay đổi dạng thể hiện của số âm.
 - Display leading zeros: **0.7** hay **.7**.
 - Measurement system: Chọn hệ thống đo lường như cm, inch, ...
 - List separator: Chọn dấu phân cách giữa các mục trong danh sách
- **Currency:** Thay đổi định dạng tiền tệ (\$, VND, ...)
- **Time:** Thay đổi định dạng giờ theo chế độ 12 giờ hay 24 giờ
- **Date:** Thay đổi định dạng ngày tháng (Date), cho phép chọn cách thể hiện ngày



50

Cài đặt / loại bỏ máy in

- **Cài đặt thêm máy in:**
 - Với một số máy in thông dụng Windows đã tích hợp sẵn chương trình điều khiển (driver) của các máy in, tuy nhiên cũng có những máy in mà trong Windows chưa có chương trình điều khiển. Muốn sử dụng những máy in này ta cần phải gọi thực hiện chương trình **Printers and Faxes** trong Control Panel.
- Các bước cài đặt máy in:
 - Chọn lệnh Start/ Settings/ Printers and Faxes
 - Click chọn Add a Printer, xuất hiện hộp thoại Add
 - Làm theo các bước hướng dẫn của hệ thống

51

Cài đặt / loại bỏ máy in (tiếp)

- **Loại bỏ máy in đã cài đặt**
 - Chọn lệnh Start/ Settings/ Printers and Faxes
 - Click chuột chọn máy in muốn loại bỏ
 - Nhấn phím Delete, sau đó chọn Yes

52

5.4.5. Windows Explorer

- Là một chương trình được hỗ trợ từ phiên bản Windows 95 cho phép người sử dụng thao tác với các tài nguyên có trong máy tính như tập tin, thư mục, ổ đĩa và những tài nguyên khác có trong máy của bạn cũng như các máy tính trong hệ thống mạng (nếu máy tính của bạn có nối mạng).
- Với Windows Explorer, các thao tác như sao chép, xóa, đổi tên thư mục và tập tin,... được thực hiện một cách thuận tiện và dễ dàng

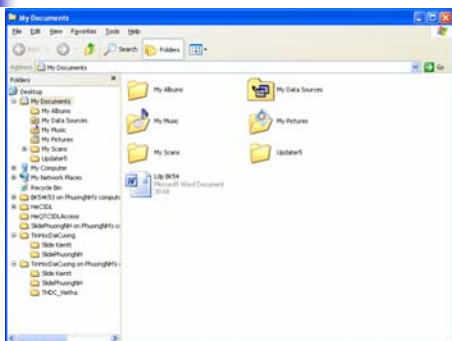
53

Khởi động chương trình Windows Explorer

- Thực hiện một trong những cách sau:
 - Chọn lệnh Start/ Programs/ Accessories/ Windows Explorer
 - R_Click lên Start, sau đó chọn Explore
 - R_Click lên biểu tượng My Computer, sau đó chọn Explore ...

54

Cửa sổ làm việc của Windows Explorer



55

Cửa sổ làm việc của Windows Explorer

- **Cửa sổ trái** (Folder) là cấu trúc cây thư mục. Nó trình bày cấu trúc thư mục của các đĩa cứng và các tài nguyên kèm theo máy tính, bao gồm ổ đĩa mềm, ổ đĩa cứng, ổ đĩa CD...
 - Những đối tượng có dấu cộng (+) ở phía trước cho biết đối tượng đó còn chứa những đối tượng khác trong nó nhưng không được hiển thị. Nếu Click vào dấu + thì Windows Explorer sẽ hiển thị các đối tượng chứa trong đối tượng đó. Khi đó, dấu + sẽ đổi thành dấu -, và nếu Click vào dấu - thì đối tượng sẽ được thu gọn trở lại.
- **Cửa sổ phải** liệt kê nội dung của đối tượng được chọn tương ứng bên cửa sổ trái

56

Thanh địa chỉ (Address Bar)

- Cho phép nhập đường dẫn thư mục/ tập tin cần tới hoặc để xác định đường dẫn hiện hành

57

Công cụ trên thanh toolbar

	- Back: Chuyển về thư mục trước đó.
	- Up: Chuyển lên thư mục cha.
	- Forward: Chuyển tới thư mục vừa quay về (Back).
	- Search: Tìm kiếm tập tin/ thư mục.
	- Folder: Cho phép ẩn/ hiện cửa sổ Folder bên trái.
	- Views: Các chế độ hiển thị các đối tượng (tập tin/ thư mục/ ổ đĩa)

58

Thao tác với thư mục và tệp

- Mở tập tin, thư mục
- Chọn tập tin, thư mục
- Tạo thư mục
- Sao chép tập tin, thư mục
- Di chuyển tập tin, thư mục
- Xóa tập tin, thư mục
- Phục hồi tập tin, thư mục
- Đổi tên tập tin, thư mục
- Thay đổi thuộc tính tập tin, thư mục

59

Mở tập tin, thư mục

- Có ba cách thực hiện :
 - Cách 1: D_Click lên biểu tượng của tập tin/ thư mục.
 - Cách 2: R_Click lên biểu tượng của tập tin/ thư mục và chọn mục Open.
 - Cách 3: Chọn tập tin/ thư mục và nhấn phím Enter.
- Nếu tập tin thuộc loại tập tin văn bản thì chương trình ứng dụng kết hợp sẽ được khởi động và tài liệu sẽ được nạp vào.
- Trong trường hợp chương trình ứng dụng không được cài đặt trong máy tính thì Windows sẽ mở hộp thoại Open With và cho chọn chương trình kết hợp. Nếu tập tin thuộc dạng chương trình ứng dụng thì chương trình tương ứng sẽ được khởi động

60

Chọn tập tin, thư mục

- Chọn một tập tin/ thư mục: Click lên biểu tượng tập tin/ thư mục.
- Chọn một nhóm tập tin/ thư mục: có thể thực hiện theo 2 cách:
 - Các đối tượng cần chọn là một danh sách gồm các đối tượng liên tục: Click lên đối tượng đầu danh sách để chọn, sau đó nhấn giữ phím Shift và Click lên đối tượng ở cuối danh sách.
 - Các đối tượng cần chọn nằm rời rạc nhau: nhấn giữ phím Ctrl và Click chọn các đối tượng tương ứng.

61

Tạo thư mục

- Chọn nơi chứa thư mục cần tạo (thư mục / ổ đĩa ở cửa sổ bên trái).
- Chọn menu **File/ New/ Folder** hoặc **R_Click/ New/ Folder**.
- Nhập tên thư mục mới, sau đó gõ Enter để kết thúc

62

Sao chép tập tin và thư mục

- Chọn các thư mục và tập tin cần sao chép. Sau đó có thể thực hiện theo một trong hai cách sau:
 - Cách 1: Nhấn giữ phím Ctrl và Drag đối tượng đã chọn đến nơi cần chép.
 - Cách 2: Nhấn tổ hợp phím Ctrl + C (hoặc Edit/ Copy hoặc R_Click và chọn Copy) để chép vào Clipboard, sau đó chọn nơi cần chép đến và nhấn tổ hợp phím Ctrl + V (hoặc Edit/ Paste hoặc R_Click và chọn Paste)

63

Di chuyển tập tin và thư mục

- Chọn các thư mục và tập tin cần di chuyển. Sau đó có thể thực hiện theo một trong hai cách sau:
 - Cách 1: Drag đối tượng đã chọn đến nơi cần di chuyển.
 - Cách 2: Nhấn tổ hợp phím Ctrl + X (hoặc Edit/ Cut hoặc R_Click và chọn Cut) để chép vào Clipboard, sau đó chọn nơi cần di chuyển đến và nhấn tổ hợp phím Ctrl + V (hoặc Edit/ Paste hoặc R_Click và chọn Paste).

64

Xóa tập tin và thư mục

- Chọn các thư mục và tập tin cần xóa.
- Chọn File/ Delete
- hoặc: Nhấn phím Delete
- hoặc: R_Click và chọn mục Delete.
- Xác nhận có thực sự muốn xoá hay không (Yes/ No)

65

Phục hồi tập tin và thư mục

- Các đối tượng bị xóa sẽ được đưa vào Recycle Bin. Nếu muốn phục hồi các đối tượng đã xóa, bạn thực hiện các thao tác sau đây:
 - D_Click lên biểu tượng Recycle Bin
 - Chọn tên đối tượng cần phục hồi.
 - Thực hiện lệnh **File/ Restore** hoặc R_Click và chọn mục **Restore**.
 - **Chú ý:** Nếu muốn xóa hẳn các đối tượng, ta thực hiện thao tác xóa một lần nữa đối với các đối tượng ở trong Recycle Bin. Nếu muốn xóa hẳn tất cả các đối tượng trong Recycle Bin, R_Click lên mục Recycle Bin và chọn mục Empty Recycle Bin.

66

Đổi tên tập tin và thư mục

- Chọn đối tượng muốn đổi tên
- Thực hiện lệnh **File/ Rename** hoặc nhấn phím F2 hoặc R_Click trên đối tượng và chọn mục Rename.
- Nhập tên mới, sau đó gõ Enter để kết thúc.
- **Chú ý:** với tập tin đang sử dụng thì các thao tác di chuyển, xoá, đổi tên không thể thực hiện được.

67

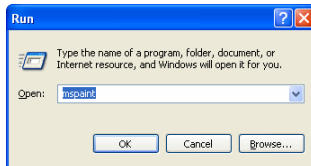
Thay đổi thuộc tính tập tin và thư mục

- Nhấn chuột phải lên đối tượng muốn thay đổi thuộc tính và chọn mục **Properties**
- Thay đổi các thuộc tính.
- Chọn **Apply** để xác nhận thay đổi, ngược lại thì nhấn **Cancel**.

68

5.4.6. Gọi thực hiện chương trình

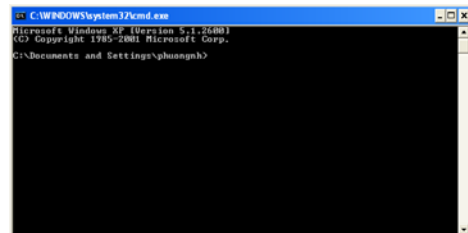
- Nếu là chương trình thực thi thì ta:
 - Kích đúp vào biểu tượng của nó
 - Hoặc Start / Run rồi gõ tên chương trình vào



69

5.4.7. Chế độ Command Prompt

- Start / Run, gõ vào cmd rồi nhấn OK



70

5.4.8. Recycle Bin

- Kích đúp vào biểu tượng Recycle bin trên màn hình desktop

Làm sạch thùng rác

Khôi phục các thư đã xóa



71



72



Lời hay ý đẹp

**"Đồ lỗi cho người khác vì những sai lầm trong cuộc đời của mình
Cũng giống như bạn đi sai đường trên một chiếc xe lửa và đổ lỗi
cho chuyên xe đó."**

<http://www.conduongthanhcong.com/>

Chương 6: Các hệ thống ứng dụng

Nguyễn Hồng Phương
Email: phuongnh-fit@mail.hut.edu.vn
Website: <http://is.hut.edu.vn/~phuongnh>
Bộ môn Hệ thống thông tin
Viện Công nghệ thông tin và Truyền thông
Đại học Bách Khoa Hà Nội

1

Nội dung chương này

- 6.1. Hệ thống thông tin quản lý
- 6.2. Hệ thống tin bảng tính
- 6.3. Hệ quản trị cơ sở dữ liệu
- 6.4. Các hệ thống thông minh

2

6.1. Hệ thống thông tin quản lý

- Khái niệm
 - *Hệ thống thông tin quản lý* là hệ thống bao gồm phần cứng, phần mềm, con người, quy trình thu thập, phân tích, xử lý, đánh giá và phân phối, chia sẻ những thông tin cần thiết một cách kịp thời và chính xác dựa trên nhu cầu của tổ chức.
 - HTTTQL thủ công: sử dụng giấy, bút, không sử dụng máy tính
 - Năm thành phần cơ bản: (1) cơ sở hạ tầng (phần cứng và hệ thống truyền thông), (2) phần mềm, (3) cơ sở dữ liệu, (4) quy trình và (5) nhân sự.

3

6.1. Hệ thống thông tin quản lý

- Chức năng
 - Nhập dữ liệu
 - Xử lý thông tin
 - Xuất dữ liệu
 - Lưu trữ thông tin
 - Thông tin phản hồi

4

6.1. Hệ thống thông tin quản lý

- Các dạng thông tin
 - Theo quan điểm cá nhân
 - Theo quan điểm tổ chức
- Đặc tính của thông tin
 - Chính xác
 - Đầy đủ
 - Thống nhất
 - Thích hợp và dễ hiểu
 - Kịp thời

5

6.1. Hệ thống thông tin quản lý

- Xây dựng và phát triển HTTT: phương pháp chu kỳ hệ thống SDLC (Systems Development Life Cycle)
 - Lập kế hoạch
 - Phân tích
 - Thiết kế
 - Cài đặt
 - Kiểm định
 - Vận hành
 - Bảo trì

6

6.2. Hệ thống tin bảng tính

- Máy tính: Hỗ trợ việc tính toán, nhất là kế toán và phân tích thống kê.
- Phần mềm thông dụng: Phần mềm bảng tính (PMBT) spreadsheet software
- PMBT: giúp tính toán các số liệu, từ đó cho phép xây dựng và làm việc với những tình huống mô phỏng thế giới thực.

7

Hệ thống tin bảng tính

- **Bảng tính - phần mềm của dự toán**
 - Tạo thay đổi lớn trong hoạt động kinh doanh
 - Giúp thao tác với con số, phương thức khó làm bằng tay
 - Rút ngắn khoảng cách thời gian thực hiện
 - Giúp khám phá mối liên hệ giữa các con số => cơ sở dự đoán tương lai

8

Hệ thống tin bảng tính (2)

■ Bảng tính: những ô lưới linh động

- Dạng ô lưới gồm: Các hàng đánh số từ 1 và Các cột đánh số từ chữ A.
- Ô là giao của 1 hàng và 1 cột. Ví dụ ô A1 là giao của hàng 1 và cột A.
- Mỗi ô có thể chứa dữ liệu dạng số, chuỗi kí tự hoặc công thức hiển thị liên hệ giữa các con số.
- Giá trị số là vật liệu thô để tính toán

9

Hệ thống tin bảng tính (3)

STT	Họ và tên	Ngày sinh	Điện thoại	Email	Toán	Lý	Hóa	Trung bình
1	Lê Đức Anh	10/10/1987	0987654321	anh101@gmail.com	8	7	8	7.67
2	Phan Đức Phúc	12/09/1986	0981348910	phucpdt@gmail.com	9	8	9	8.67
3	Lê Anh Vũ Hà	23/08/1987	0947844888	anh17@yahoo.com	6	8	8	7.33
4	Nguyễn Trương Sơn	30/04/1986	0964021663	sontruongson1@gmail.com	7	8	9	8
5	Nguyễn Thị Thanh	06/02/1987	0982148611	thanh15@gmail.com	8	7	9	7.33
6	Nguyễn Thị Thùy Mai	22/12/1986	0168297498	mai_cmt1@yahoo.com	5	8	5	6
7	Ngô Văn Mạnh	23/12/1986	0975194713	manh_vantran@gmail.com	6	9	6	7
8	Trần Tuấn Anh	26/03/1987	0123821400	truan111@gmail.com	8	6	7	7
9	Lưu Đức Lý	02/09/1987	0985608020	liuvntrun13@gmail.com	9	8	8	8.33
10	Đỗ Lê Minh	01/06/1987	0936100692	minhdo1@gmail.com	10	8	9	9
11	Đinh Hoàng Đức	01/01/1986	0972147011	huongduc@gmail.com	7	7	7	6.33
12	Lê Sĩ Đức	10/12/1987	0983402387	lehsiduc@gmail.com	8	8	6	7.33

10

Hệ thống tin bảng tính (4)

Toán	Lý	Hóa	Trung bình
8	7	8	7.67
9	8	9	8.67
6	8	8	7.33

11

Hệ thống tin bảng tính (5)

- Các chức năng cơ bản của PMBT:
 - **Tự động lập các giá trị, tiêu đề và công thức:** Giúp đơn giản hóa việc nhập các dữ liệu lặp.
 - **Tự động tính lại:** Khi có một sự thay đổi tại 1 ô thì toàn bộ bảng tính sẽ được tính toán lại.
 - **Các hàm thư viện:** thực hiện các công việc tính toán đã định sẵn. Giúp tiết kiệm thời gian và giảm nguy cơ phát sinh lỗi.

12

Hệ thống tin bảng tính (6)

- Các chức năng cơ bản của PMBT:
 - **Macro**: Giúp “thu” lại các thao tác lặp đi lặp lại và định nghĩa nó là 1 macro. Khi cần thực hiện các thao tác đó thì chỉ việc gọi macro tương ứng.
 - **Bảng tính mẫu**: Chỉ bao gồm các tiêu đề và công thức nhưng không chứa dữ liệu. Giúp tiết kiệm thời gian và công sức.
 - **Liên kết**: Cho phép tạo liên kết động giữa các bảng tính.
 - **Cơ sở dữ liệu**: Cho phép thao tác: lưu trữ và truy cập thông tin, tìm kiếm, báo cáo,...

13

Hệ thống tin bảng tính (7)

- Những đặc điểm nổi bật khác:
 - Công cụ giải phương trình, những bài toán tối ưu.
 - Lotus hỗ trợ Multimedia, Excel sử dụng trí tuệ nhân tạo...
 - Vẽ đồ thị: từ các con số chuyển thành đồ thị để biểu đạt thông tin: đồ thị tròn, đồ thị đường, đồ thị cột...

14

Hệ thống tin bảng tính (8)

- Kinh nghiệm sử dụng:
 - Hãy hình dung bảng tính trước khi bạn đưa ra các giá trị và công thức vào
 - Kiểm tra nhiều lần mỗi công thức và giá trị
 - Làm bảng tính trở nên dễ đọc.
 - Kiểm tra kết quả bằng những cách khác
 - Xây dựng các hàm kiểm tra chéo
 - Đổi giá trị đầu vào và quan sát kết quả
 - Hãy tận dụng những hàm có sẵn
 - PMBT hỗ trợ quyết định chứ không thay quyết định.

15

6.3. Hệ quản trị cơ sở dữ liệu

- Khái niệm hệ quản trị cơ sở dữ liệu
- Các tính năng của hệ quản trị cơ sở dữ liệu

16

Khái niệm cơ sở dữ liệu

- Là một tập hợp các dữ liệu
 - Biểu diễn một vài khía cạnh nào đó của thế giới thực
 - Có liên hệ logic thống nhất
 - Được thiết kế và bao gồm những dữ liệu phục vụ một mục đích nào đó.
- Là một bộ sưu tập các dữ liệu tác nghiệp được lưu trữ lại và được các hệ ứng dụng của một xí nghiệp cụ thể nào đó sử dụng.
- Là một hệ thống các thông tin có cấu trúc, lưu trữ trên các thiết bị lưu trữ thông tin.
- Ví dụ:
 - Trang niên giám điện thoại
 - Danh sách sinh viên.
 - Hệ thống tài khoản ngân hàng.

17

Ưu điểm khi sử dụng CSDL:

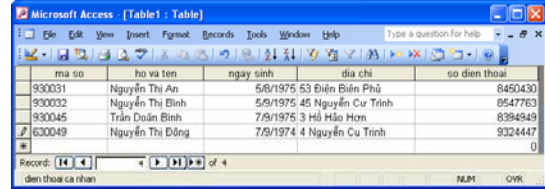
- việc lưu trữ một lượng thông tin khổng lồ trở nên dễ dàng.
- Giúp nhanh chóng và mềm dẻo trong việc tra cứu thông tin.
- Giúp dễ dàng sắp xếp và tổ chức thông tin
- Giúp in và phân phối thông tin theo nhiều cách.

18

Bên trong cơ sở dữ liệu:

- Một CSDL được hình thành từ các file chứa một tập thông tin có liên quan.
- Một file CSDL bao gồm:
 - Nhiều bản ghi (record): 1 bản ghi là thông tin liên quan đến 1 người, 1 sản phẩm hoặc 1 sự kiện nào đó.
 - Nhiều trường (field): Mỗi 1 đoạn thông tin riêng rẽ trong 1 record là 1 trường.
 - Ví dụ: 1 record trong csdl thư viện có các field cho tác giả, tựa đề sách, nhà XB, địa chỉ, ...
 - Mỗi trường được xác định bằng kiểu cụ thể: có các kiểu ngày, chữ, số,...

19



ma so	ho va ten	ngay sinh	dia chi	so dien thoai
930031	Nguyễn Thị An	5/6/1975	53 Điện Biên Phủ	8450430
930032	Nguyễn Thị Bình	5/9/1975	45 Nguyễn Cư Trinh	8547763
930045	Trần Đoàn Bình	7/9/1975	3 Hồ Hảo Hớn	8394949
630049	Nguyễn Thị Đông	7/9/1974	4 Nguyễn Cư Trinh	9324447

20

Hệ quản trị cơ sở dữ liệu

- Là một hệ thống phần mềm cho phép
 - Định nghĩa, tạo lập: xác định kiểu, cấu trúc, ràng buộc dữ liệu, lưu trữ dữ liệu trên các thiết bị nhớ.
 - Thao tác: truy vấn, cập nhật, kết xuất, ... các CSDL cho các ứng dụng khác nhau
- Ví dụ: MS SQL Server, DB2, MS Access, Oracle, FoxPro, ...



21

Các tính năng của hệ quản trị CSDL

- Quản lý dữ liệu tồn tại lâu dài
- Truy xuất dữ liệu một cách hiệu quả
- Hỗ trợ ít nhất một mô hình dữ liệu
- Đảm bảo tính độc lập dữ liệu, toàn vẹn dữ liệu
- Hỗ trợ các ngôn ngữ cấp cao nhất định cho phép người sử dụng định nghĩa cấu trúc của dữ liệu, truy nhập và thao tác dữ liệu
- Quản trị giao dịch
- Điều khiển truy nhập
- Sao lưu và phục hồi dữ liệu



22

6.4. Các hệ thống thông minh

- Hệ dựa trên tri thức, hệ chuyên gia
- Tác tử thông minh
- Công nghệ cảm biến, công nghệ số: ngôi nhà thông minh, máy móc, trang thiết bị thông minh.

23



24



Lời hay ý đẹp

"Không có điều vĩ đại nào được thực hiện mà không có những con người vĩ đại. Và họ chỉ vĩ đại nếu họ quyết tâm trở thành người vĩ đại"

<http://www.conduongthanhhong.com/>

Phần II: Giải quyết bài toán

Nguyễn Hồng Phương
Email: phuongnh-fit@mail.hut.edu.vn
Website: <http://is.hut.edu.vn/~phuongnh>
Bộ môn Hệ thống thông tin
Viện Công nghệ thông tin và Truyền thông
Đại học Bách Khoa Hà Nội

1

Nội dung phần này

- Chương 1: Giải quyết bài toán bằng máy tính
 - Khái niệm về bài toán
 - Quá trình giải quyết bài toán bằng máy tính
 - Các phương pháp giải quyết bài toán bằng máy tính
 - Phân loại bài toán
- Chương 2: Thuật toán
 - Định nghĩa thuật toán
 - Biểu diễn thuật toán
 - Một số thuật toán thông dụng
 - Thuật toán đệ quy
 - Thuật giải heuristic

2

Chương 1: Giải quyết bài toán bằng máy tính

Nguyễn Hồng Phương
Email: phuongnh-fit@mail.hut.edu.vn
Website: <http://is.hut.edu.vn/~phuongnh>
Bộ môn Hệ thống thông tin
Viện Công nghệ thông tin và Truyền thông
Đại học Bách Khoa Hà Nội

3

Nội dung chương này

- 1.1. Khái niệm về bài toán
- 1.2. Các bước giải quyết bài toán bằng máy tính
- 1.3. Các phương pháp giải quyết vấn đề bằng máy tính
- 1.4. Phân loại bài toán

4

1.1. Khái niệm về vấn đề và bài toán

- Vấn đề rộng hơn bài toán?
- Pitago chia vấn đề ra:
 - Theorema là vấn đề cần được khẳng định đúng-sai
 - Problema là vấn đề cần tìm giải pháp để đạt được một mục tiêu xác định từ những điều kiện ban đầu.
- Diễn đạt bằng sơ đồ: $A \rightarrow B$
 - A là giả thiết, điều kiện ban đầu
 - B là kết luận, mục tiêu cần đạt
 - \rightarrow là suy luận, giải pháp cần xác định

5

1.2. Các bước giải quyết bài toán bằng máy tính

- Bước 1: Xác định vấn đề-bài toán
- Bước 2: Lựa chọn phương pháp giải
- Bước 3: Xây dựng thuật toán hoặc thuật giải
- Bước 4: Cài đặt chương trình
- Bước 5: Hiệu chỉnh chương trình
- Bước 6: Thực hiện chương trình

6

1.3. Các phương pháp giải quyết vấn đề bằng máy tính

- Giải quyết vấn đề theo hướng xác định trực tiếp lời giải
 - xác định trực tiếp lời giải qua thủ tục tính toán hoặc thủ tục bao gồm một số hữu hạn các thao tác sơ cấp.
- Giải quyết vấn đề theo hướng tìm kiếm lời giải
 - nguyên lý "thử và sai"
 - các phương pháp
 - liệt kê hay vét cạn
 - thử ngẫu nhiên
 - quay lui
 - chia để trị

7

1.4. Phân loại bài toán

- Bài toán đa thức
- Bài toán không đa thức
- NP Problems

8



9

Lời hay ý đẹp

Chân thật và thẳng thắn là bước đầu cần thiết làm nảy nở và củng cố mối quan hệ bạn bè

Lênin

10

Chương 2: Thuật toán

Nguyễn Hồng Phương
Email: phuongnh-fit@mail.hut.edu.vn
Website: <http://is.hut.edu.vn/~phuongnh>
Bộ môn Hệ thống thông tin
Viện Công nghệ thông tin và Truyền thông
Đại học Bách Khoa Hà Nội

1

Nội dung chương này

- 2.1. Định nghĩa thuật toán
- 2.2. Biểu diễn thuật toán
- 2.3. Một số thuật toán thông dụng
- 2.4. Thuật toán đệ quy
- 2.5. Thuật giải heuristic

2

2.1. Định nghĩa thuật toán

- Là một khái niệm cơ sở của toán học và tin học.
- Bao gồm một dãy hữu hạn các lệnh/chỉ thị rõ ràng và có thể thi hành được để hướng dẫn thực hiện một hành động nhằm đạt được mục tiêu đề ra.
- Thuật toán là sự thể hiện của một phương pháp để giải quyết một vấn đề.

3

Ví dụ 1: Thuật toán tìm phần tử lớn nhất của một dãy hữu hạn các số nguyên

- Các bước:
 - Đặt giá trị lớn nhất tạm thời là số nguyên đầu tiên.
 - So sánh số nguyên kế tiếp trong dãy với giá trị lớn nhất tạm thời, nếu số nguyên này lớn hơn giá trị lớn nhất tạm thời thì đặt giá trị lớn nhất tạm thời bằng số nguyên này.
 - Lặp lại bước 2 nếu còn số nguyên trong dãy chưa được xét.
 - Dừng nếu không còn số nguyên nào trong dãy chưa được xét. Giá trị lớn nhất tạm thời lúc này chính là giá trị lớn nhất trong dãy số.

4

Ví dụ 2: Thuật toán giải phương trình bậc hai: $ax^2 + bx + c = 0$ ($a \neq 0$)

- Nhập 3 hệ số a, b, c
- Tính giá trị $\Delta = b^2 - 4*a*c$
- Xét dấu Δ . Nếu $\Delta > 0$ thì thực hiện các thao tác sau đây:
 - Tính các nghiệm theo các công thức:
 - $x_1 = (-b - \sqrt{\Delta}) / (2*a)$
 - $x_2 = (-b + \sqrt{\Delta}) / (2*a)$
 - Xuất kết quả: phương trình có hai nghiệm x_1 và x_2 .
- Nếu $\Delta = 0$ thì xuất kết quả: phương trình có nghiệm kép là $-b / (2*a)$
- Nếu $\Delta < 0$ thì xuất kết quả: phương trình vô nghiệm
- Dừng thuật toán

5

Các đặc trưng của thuật toán

- Nhập (input):** có các giá trị nhập từ một tập hợp nhất định.
- Xuất (output):** từ mỗi giá trị của tập hợp nhập, tạo ra giá trị xuất thuộc một tập hợp nhất định.
- Tính xác định (definiteness):** các bước chính xác, rõ ràng.
- Tính hữu hạn (finiteness):** cho ra kết quả sau một số hữu hạn bước.
- Tính hiệu quả:** được đánh giá dựa trên một số tiêu chuẩn (khối lượng tính toán, không gian, thời gian sử dụng).
- Tính tổng quát:** áp dụng được cho tất cả các bài toán có dạng như mong muốn

6

2.2. Biểu diễn thuật toán

- Sử dụng các ngôn ngữ:
 - Ngôn ngữ tự nhiên
 - Ngôn ngữ lưu đồ (sơ đồ khối)
 - Ngôn ngữ tựa ngôn ngữ lập trình (mã giả)
 - Ngôn ngữ lập trình

7

Ngôn ngữ lưu đồ

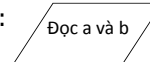
- Các thành phần:
 - Nút giới hạn: được biểu diễn bởi hình ôvan có ghi chữ bên trong, gồm có nút đầu và nút cuối:



- Nút thao tác: là một hình chữ nhật có ghi các lệnh cần thực hiện:



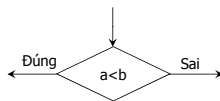
- Nút nhập/xuất dữ liệu:



8

Ngôn ngữ lưu đồ (2)

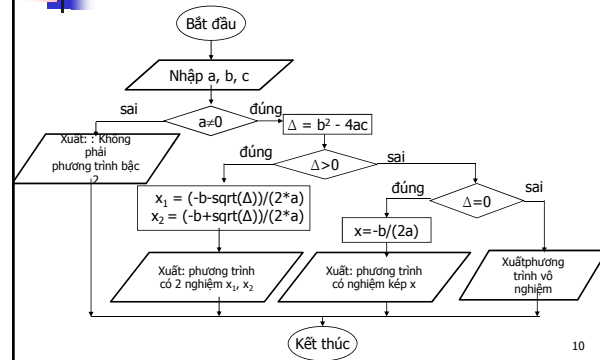
- Nút điều kiện: là một hình thoi có ghi điều kiện cần kiểm tra, thường có 1 cung đi vào và 2 cung đi ra (tương ứng với 2 trường hợp đúng/sai)



- Cung: là đường nối từ nút này đến nút khác của lưu đồ

9

Ví dụ: lưu đồ biểu diễn thuật toán giải phương trình bậc 2



10

Mã giả

- Sử dụng mệnh đề có cấu trúc chuẩn hóa và vẫn dùng ngôn ngữ tự nhiên.
- Sử dụng các ký hiệu toán học, các biến, cấu trúc kiểu thủ tục.
- Hành động gán:
 - $i \leftarrow i+1$
- Tiện lợi, đơn giản, vẫn dễ hiểu.

11

Mã giả (2)

- Các cấu trúc thường gặp:
 - Cấu trúc chọn:
 - if (điều kiện) then (hành động) end if
 - if (điều kiện) then (hành động 1) else (hành động 2) end if
 - Cấu trúc lặp:
 - while (điều kiện) do (hành động) end while
 - repeat (hành động) until (điều kiện)
 - for (biến)=(giá trị đầu) to (giá trị cuối) do (hành động) end for
 - for (biến)=(giá trị cuối) downto (giá trị đầu) do (hành động) end for
 - Cấu trúc nhảy:
 - goto nhãn x;

12

Ví dụ: thuật toán giải phương trình bậc 2

- **Nhập:** các hệ số a, b, c
- **Xuất:** kết luận về nghiệm của phương trình bậc hai
- **Thuật toán:**
 - if a = 0 then
 Xuất: Không phải phương trình bậc hai, Dừng
 - end if
 - delta $b^2-4*a*c$
 - if delta > 0 then
 - x1 $(-b-\text{sqrt}(\Delta))/(2*a)$
 - x2 $(-b+\text{sqrt}(\Delta))/(2*a)$
 - Dừng
 - else if delta = 0 then x12 $-b/(2*a)$, Xuất: nghiệm kép x12
 - else Xuất: phương trình vô nghiệm
 - end if

13

2.3. Một số thuật toán thông dụng

- Thuật toán kiểm tra số nguyên tố
- Thuật toán tìm USCLN, BSCNN của 2 số nguyên
- Thuật toán tìm phần tử lớn nhất trong một dãy
- Thuật toán sắp xếp
- Thuật toán tìm kiếm

14

Tìm phần tử lớn nhất trong một dãy hữu hạn số

- **Nhập:** dãy số a[1], a[2], a[3],... a[n]
- **Xuất:** max là giá trị lớn nhất trong dãy số đã cho
- **Thuật toán:**

```
max a[1]
for i = 2 to n do
    if max < a[i] then
        max a[i]
    end if
end for
```

Xuất: max là giá trị lớn nhất trong dãy số

15

2.4. Thuật toán đệ quy

- Có một số trường hợp, cách giải có thể vi phạm các tính chất của thuật toán nhưng lại khá đơn giản và được chấp nhận.
- Bài toán có thể được phân tích và đưa tới việc giải một bài toán cùng loại nhưng cấp độ thấp hơn.
- Ví dụ:
 - Định nghĩa giai thừa
 - $0! = 1$
 - $n! = n*(n-1)!$ với $n > 0$
 - Định nghĩa dãy số Fibonacci: 1, 1, 2, 3, 5, 8, 13, ...
 - $f_1 = 1,$
 - $f_2 = 1,$
 - $f_n = f_{n-1} + f_{n-2}$

16

Thuật toán đệ quy (2)

- Thuật toán đệ quy tính giai thừa của 1 số tự nhiên:
 - Input: số tự nhiên n
 - Output: F(n) bằng n!
 - Thuật giải:
 1. F:=1
 2. if $n > 0$ then F:=F(n-1)*n
 3. Output F

17

Thuật toán đệ quy (3)

- Thuật toán đệ quy tính số hạng thứ n của dãy số Fibonacci:
 - Input: số tự nhiên n
 - Output: F(n) bằng số hạng thứ n của dãy
 - Thuật giải:
 1. if $n=1$ or $n=2$ then F:=1
 2. if $n > 2$ then F:=F(n-1)+F(n-2)
 3. Output F

18

Thuật toán đệ quy (4)

- Đặc điểm của thuật toán đệ quy:
 - Có 1 trường hợp cơ sở/trường hợp dừng
 - Có phần đệ quy bên trong thuật toán (nó gọi đến chính nó)
 - Có sự biến đổi tiến tới trường hợp cơ sở.

19

Bài tập

- Viết thuật toán tìm USCLN của hai số tự nhiên
- Viết thuật toán tìm BSCNN của hai số tự nhiên
- Viết thuật toán tìm phần tử lớn nhất trong một dãy số hữu hạn
- Viết thuật toán sắp xếp
- Viết thuật toán tìm kiếm

20

2.5. Thuật giải heuristic

- Thường tìm được lời giải tốt (những chưa chắc đã tốt nhất)
- Dễ dàng và nhanh chóng hơn so với giải thuật tối ưu
- Thể hiện một cách hành động khá tự nhiên, gần gũi với suy nghĩ và hành động của con người.

21

Thuật giải heuristic (2)

- Các nguyên lý
 - Nguyên lý vét cạn thông minh: trong bài toán tìm kiếm khi không gian tìm kiếm lớn => giới hạn không gian tìm kiếm hoặc thực hiện dò tìm đặc biệt dựa vào đặc thù của bài toán để nhanh chóng tìm ra mục tiêu
 - Nguyên lý tham lam: lấy tiêu chuẩn tối ưu toàn cục làm tiêu chuẩn chọn lựa hành động cục bộ của từng bước trong quá trình tìm kiếm lời giải
 - Nguyên lý thử tự: thực hiện hành động theo thứ tự hợp lý của không gian khảo sát nhằm nhanh chóng đạt được một lời giải tốt.

22

Lời hay ý đẹp

Cuộc đời chỉ là một chuỗi những cơ hội ngẫu nhiên. Cái khó là nắm bắt cơ hội mà hành động. Đứng bao giờ để vượt mặt. Không phải ngày nào bạn cũng gặp cơ hội may đâu "

George Besnard-Shaw

23

Phần 3 - Lập trình bằng ngôn ngữ C

Nguyễn Hồng Phương
Email: phuongnh-fit@mail.hut.edu.vn
Website: <http://is.hut.edu.vn/~phuongnh>
Bộ môn Các hệ thống thông tin
Viện Công nghệ thông tin và Truyền thông
Đại học Bách Khoa Hà Nội

1

Mục tiêu

- Giúp sinh viên bước đầu học lập trình thông qua phương tiện là ngôn ngữ lập trình C

2

Nội dung phần 3

- Chương 1: Giới thiệu tổng quan về ngôn ngữ lập trình C
- Chương 2: Kiểu dữ liệu và biểu thức trong C
- Chương 3: Các cấu trúc lập trình trong C
- Chương 4: Con trỏ và mảng
- Chương 5: Xâu ký tự
- Chương 6: Hàm
- Chương 7: Cấu trúc
- Chương 8: Tập

3



4

Lời hay ý đẹp

"The love we gave is the only love we keep"
(Tình yêu cho đi là thứ tình yêu duy nhất ta còn giữ)

5

Chương 1: Tổng quan về ngôn ngữ C

Nguyễn Hồng Phương
Email: phuongnh-fit@mail.hut.edu.vn
Website: <http://is.hut.edu.vn/~phuongnh>
Bộ môn Hệ thống thông tin
Viện Công nghệ thông tin và Truyền thông
Đại học Bách Khoa Hà Nội

1

Nội dung chương này

- 1.1. Lịch sử phát triển ngôn ngữ lập trình C
- 1.2. Các phần tử cơ bản của ngôn ngữ C
- 1.3. Cấu trúc cơ bản của một chương trình C
- 1.4. Biên dịch chương trình viết bằng C
- 1.5. Bài tập

2

1.1. Lịch sử phát triển ngôn ngữ lập trình C

- Ngôn ngữ lập trình C (NNLT C) ra đời tại phòng thí nghiệm BELL của tập đoàn AT&T (Hoa Kỳ)
- Do Brian W. Kernighan và Dennis Ritchie phát triển vào đầu 1970, hoàn thành 1972
- C dựa trên nền các ngôn ngữ BCPL (*Basic Combined Programming Language*) và ngôn ngữ B.
- Tên là ngôn ngữ C như là sự tiếp nối ngôn ngữ B.

3

1.1. Lịch sử phát triển ngôn ngữ lập trình C

- Đặc điểm của NNLT C:
 - Là một ngôn ngữ lập trình hệ thống mạnh, khả chuyển, có tính linh hoạt cao.
 - Có thể mạnh trong xử lý các dạng dữ liệu số, văn bản, cơ sở dữ liệu.
 - Thường được sử dụng để viết:
 - Các chương trình hệ thống như hệ điều hành (VD Unix: 90% viết bằng C, 10% viết bằng hợp ngữ).
 - Các chương trình ứng dụng chuyên nghiệp có can thiệp tới dữ liệu ở mức thấp như xử lý văn bản, xử lý ảnh...

4

1.1. Lịch sử phát triển ngôn ngữ lập trình C

- 1978: C được giới thiệu trong phiên bản đầu của cuốn sách "*The C programming language*"
- Sau đó, C được bổ sung thêm những tính năng và khả năng mới → Đồng thời tồn tại nhiều phiên bản nhưng không tương thích nhau.
- Năm 1989, Viện tiêu chuẩn quốc gia Hoa Kỳ (*American National Standards Institute - ANSI*) đã công bố phiên bản chuẩn hóa của ngôn ngữ C: *ANSI C* hay *C chuẩn* hay *C89*

5

1.1. Lịch sử phát triển ngôn ngữ lập trình C

- Tất cả các phiên bản của ngôn ngữ C hiện nay đều tuân theo các mô tả đã được nêu ra trong ANSI C, sự khác biệt nếu có thì chủ yếu ở các thư viện bổ sung.
- Hiện nay cũng có nhiều phiên bản của ngôn ngữ C khác nhau, gắn liền với một bộ chương trình dịch cụ thể của ngôn ngữ C:
 - Turbo C++** và Borland C++ của Borland Inc.
 - MSC và VC của Microsoft Corp.
 - GCC của GNU project.

6

1.2. Các phần tử cơ bản của ngôn ngữ C

7

1.2.1. Tập ký tự

- Chương trình C được tạo ra từ các phần tử cơ bản là tập ký tự .
- Các ký tự tổ hợp với nhau tạo thành các từ
- Các từ liên kết với nhau theo một quy tắc xác định để tạo thành các câu lệnh
- Từ các câu lệnh → tổ chức thành chương trình.

8

1.2.1. Tập ký tự (tiếp)

26 chữ cái hoa:	A B C ... X Y Z
26 chữ cái thường:	a b c ... x y z
10 chữ số:	0 1 2 3 4 5 6 7 8 9
Các kí hiệu toán:	+ - * / = < >
Các dấu ngăn cách:	. ; , : space tab
Các dấu ngoặc:	() [] { }
Các kí hiệu đặc biệt:	_ ? \$ & # ^ \ ! ' " ~

9

1.2.2. Từ khóa (keyword)

- Là những từ có sẵn của ngôn ngữ và được sử dụng dành riêng cho những mục đích xác định.
- Các từ khóa trong C được sử dụng để
 - Đặt tên cho các kiểu dữ liệu: **int, float, double, char, struct, union...**
 - Mô tả các lệnh, các cấu trúc điều khiển: **for, do, while, switch, case, if, else, break, continue...**

10

1.2.2. Từ khóa (keyword) (tiếp)

Một số từ khóa hay dùng trong Turbo C++

break	case	char	const	continue	default
do	double	else	enum	float	for
goto	if	int	interrupt	long	return
short	signed	sizeof	static	struct	switch
typedef	union	unsigned	void	while	

11

1.2.3. Định danh / tên (identifier)

- Là một dãy các ký tự dùng để gọi tên các đối tượng trong chương trình.
 - Các đối tượng trong chương trình gồm có biến, hằng, hàm, kiểu dữ liệu... ta sẽ làm quen ở những mục tiếp theo.
- Có thể được đặt tên:
 - Bởi ngôn ngữ lập trình (đó chính là các từ khóa)
 - Hoặc do người lập trình đặt.

12

1.2.3. Định danh / tên (identifier) (tiếp)

- Quy tắc đặt tên:
 - Chỉ được gồm có: **chữ cái**, **chữ số** và dấu gạch dưới “_” (*underscore*).
 - Bắt đầu của định danh phải là chữ cái hoặc dấu gạch dưới, không được bắt đầu định danh bằng chữ số.
 - Định danh do người lập trình đặt không được trùng với từ khóa.

13

1.2.3. Định danh / tên (identifier) (tiếp)

- Ví dụ định danh/tên hợp lệ:
i, x, y, a, b, _function, _MY_CONSTANT, PI, gia_tri_1
- Ví dụ về định danh/tên không hợp lệ:

1_a, 3d, 55x	bắt đầu bằng chữ số
so luong, ti le	có kí tự không hợp lệ (dấu cách - <i>space</i>) trong tên
int, char	trùng với từ khóa của ngôn ngữ C

14

1.2.3. Định danh / tên (identifier) (tiếp)

- Cách thức đặt định danh/tên:
 - Hằng số: chữ hoa
 - Các biến, hàm hay cấu trúc: Bằng chữ thường.
 - Nếu tên gồm nhiều từ thì ta nên phân cách các từ bằng dấu gạch dưới.
- Ví dụ:

Định danh	Loại đối tượng
HANG_SO_1, _CONSTANT_2	hằng
a, b, i, j, count	biến
nhap_du_lieu, tim_kiem, xu_ly	hàm
sinh_vien, mat_hang	cấu trúc

15

1.2.4. Các kiểu dữ liệu

- Là một tập hợp các giá trị mà một dữ liệu thuộc kiểu dữ liệu đó có thể nhận được.
- Trên một kiểu dữ liệu ta xác định một số phép toán đối với các dữ liệu thuộc kiểu dữ liệu đó.
- Ví dụ: Trong ngôn ngữ C có kiểu dữ liệu **int**. Một dữ liệu thuộc kiểu dữ liệu **int** thì:
 - Là một số nguyên (*integer*)
 - Có thể nhận giá trị từ -2^{15} đến $2^{15} - 1$.

16

1.2.4. Các kiểu dữ liệu (tiếp)

- Trên kiểu dữ liệu **int** ngôn ngữ C định nghĩa các phép toán số học đối với số nguyên như sau:
 - Đảo dấu: -
 - Cộng: +
 - Trừ: -
 - Nhân: *
 - Chia lấy phần nguyên: /
 - Chia lấy phần dư: %
 - So sánh bằng: ==
 - So sánh lớn hơn: >
 - So sánh nhỏ hơn: <

17

1.2.5. Hằng số (constant)

- Là đại lượng có giá trị không đổi trong chương trình.
- Để giúp chương trình dịch nhận biết hằng ta cần nắm được cách biểu diễn hằng trong một chương trình C.

18

Biểu diễn hằng số nguyên

- **Dạng thập phân:**
 - Giá trị số dưới hệ đếm cơ số 10 thông thường
 - Ví dụ: 2007, 396
- **Dạng thập lục phân:**
 - Giá trị số dưới dạng hệ đếm cơ số 16 và thêm tiền tố **0x**
 - Ví dụ: 0x7D7, 0x18C.
- **Dạng bát phân:**
 - Giá trị số dưới dạng hệ đếm cơ số 8 và thêm tiền tố **0**
 - Ví dụ: 03727, 0614.

19

Biểu diễn hằng số thực

- Dưới dạng số thực dấu phẩy tĩnh:
Ví dụ: 3.14159 , 123.456
- Dưới dạng số thực dấu phẩy động:
Ví dụ: 31.4159 E -1
12.3456 E +1
1.23456 E +2

20

Biểu diễn hằng ký tự

- Bằng ký hiệu của ký tự đó đặt giữa 2 dấu nháy đơn ("").
- Bằng số thứ tự của ký tự đó trong bảng mã ASCII (và lưu ý số thứ tự của một ký tự trong bảng mã ASCII là một số nguyên nên có một số cách biểu diễn).

21

Biểu diễn hằng ký tự - Ví dụ:

Kí tự cần biểu diễn	Cách 1	Cách 2
Chữ cái A	'A'	65 hoặc 0101 hoặc 0x41
Dấu nháy đơn '	'\''	39 hoặc 047 hoặc 0x27
Dấu nháy kép "	'\"'	34 hoặc 042 hoặc 0x22
Dấu gạch chéo ngược \	'\\'	92 hoặc 0134 hoặc 0x5c
Kí tự xuống dòng	'\n'	
Kí tự NUL	'\0'	0 hoặc 00 hoặc 0x0

22

Biểu diễn hằng xâu ký tự

- Một hằng là xâu kí tự được biểu diễn bởi dãy các kí tự thành phần có trong xâu đó và được đặt trong cặp dấu nháy kép ("").
- Ví dụ: "Đại học Bách Khoa", "Tin học đại cương", "Nguyễn Hồng Phương",...

23

1.2.6. Biến (variable)

- Là đại lượng mà giá trị có thể thay đổi trong chương trình.
- Hằng và biến được sử dụng để lưu trữ dữ liệu, và phải thuộc một kiểu dữ liệu nào đó.
- Tên biến và hằng được đặt theo quy tắc đặt tên cho định danh.

24

1.2.7. Hàm (function)

- Còn được gọi là chương trình con
- Những đoạn chương trình lặp đi lặp lại nhiều lần ở những chỗ khác nhau → Viết thành hàm để khi cần chỉ cần gọi ra chứ không phải viết lại toàn bộ.
- Giải quyết một bài toán lớn thì chương trình của ta có thể rất lớn và dài → Chia thành các công việc nhỏ hơn được viết thành các hàm.

25

Một số hàm toán học hay dùng trong C

Một số hàm toán học hay được sử dụng trong C

Hàm	Ý nghĩa	Kí hiệu toán học	Ví dụ
sqrt(x)	Căn bậc 2 của x	\sqrt{x}	sqrt(16.0) bằng 4.0
pow(x, y)	x mũ y	x^y	pow(2, 3) bằng 8
exp(x)	e mũ x	e^x	exp(1.0) bằng 2.718282
log(x)	logarithm tự nhiên (cơ số e) của x	$\ln x$	log(2.718282) bằng 1.0

26

Một số hàm toán học hay dùng trong C (tiếp)

log10(x)	logarithm cơ số 10 của x	$\log x$	log10(100) bằng 2
sin(x)	sin của x	$\sin x$	sin(0.0) bằng 0.0
cos(x)	cosin của x	$\cos x$	cos(0.0) bằng 1.0
tan(x)	tang của x	$\tan x$	tan(0.0) bằng 0.0
ceil(x)	phần nguyên giả của x, tức là số nguyên nhỏ nhất không nhỏ hơn x	$\lceil x \rceil$	ceil(2.5) bằng 3 ceil(-2.5) bằng -2
floor(x)	phần nguyên non của x, tức là số nguyên lớn nhất không lớn hơn x	$\lfloor x \rfloor$	floor(2.5) bằng 2 floor(-2.5) bằng -3

27

1.2.8. Câu lệnh (statement)

- Diễn tả một hoặc một nhóm các thao tác trong giải thuật.
- Chương trình được tạo thành từ dãy các câu lệnh.
- Cuối mỗi câu lệnh đều có dấu chấm phẩy (;) để đánh dấu kết thúc câu lệnh.

28

1.2.8. Câu lệnh (tiếp)

- Câu lệnh được chia thành 2 nhóm chính:
 - Nhóm các câu lệnh đơn:
 - Không chứa câu lệnh khác.
 - Ví dụ: phép gán, phép cộng, phép trừ...
 - Nhóm các câu lệnh phức:
 - Chứa câu lệnh khác trong nó.
 - Ví dụ: lệnh khối, các cấu trúc lệnh rẽ nhánh, cấu trúc lệnh lặp...
 - Lệnh khối là một số các lệnh đơn được nhóm lại với nhau và đặt trong cặp dấu ngoặc nhọn { }

29

1.2.9. Chú thích (Comment)

- Lời mô tả, giải thích vắn tắt cho một câu lệnh, một đoạn chương trình hoặc cả chương trình
- Chỉ có tác dụng giúp chương trình viết ra dễ đọc và dễ hiểu hơn
- Trình biên dịch sẽ tự động bỏ qua không dịch phần nội dung nằm trong phạm vi của vùng chú thích đó.
- 2 cách chú thích
 - Trên 1 dòng: //
 - Trên nhiều dòng: /* */

30

1.2.9. Chú thích (tiếp)

- Cách 1:
 - Vùng bắt đầu từ // đến cuối dòng là vùng chú thích.
 - Ví dụ:
`a = 5; b = 3; // Khai tạo giá trị cho các biến này`
- Cách 2:
 - Toàn bộ vùng bắt đầu nằm trong cặp kí hiệu /* */ là vùng chú thích.
 - Ví dụ:
`/* Đoạn chương trình sau khai báo 2 biến nguyên và khai tạo giá trị cho 2 biến nguyên này */
int a, b;
a = 5; b = 3;`

31

1.3. Cấu trúc cơ bản của một chương trình C

- Gồm 6 phần có thứ tự như sau:

Phần 1: Khai báo tệp tiêu đề: #include
Phần 2: Định nghĩa kiểu dữ liệu mới: typedef ...
Phần 3: Khai báo các hàm nguyên mẫu
Phần 4: Khai báo các biến toàn cục
Phần 5: Hàm main()
Phần 6: Nội dung các hàm đã khai báo

32

1.3. Cấu trúc cơ bản... (tiếp)

- Phần 1: Khai báo tệp tiêu đề:
 - Thông báo cho chương trình dịch biết là chương trình có sử dụng những thư viện nào.
 - Ví dụ:
`#include <stdio.h> // thao tác vào ra
#include <conio.h> // hàm của DOS`
- Phần 2: Định nghĩa các kiểu dữ liệu mới
 - Định nghĩa các kiểu dữ liệu mới (nếu cần) dùng cho cả chương trình.

33

1.3. Cấu trúc cơ bản... (tiếp)

- Phần 3: Khai báo các hàm nguyên mẫu:
 - Giúp cho chương trình dịch biết được những thông tin cơ bản của các hàm sử dụng trong chương trình.
- Phần 4: Khai báo các biến toàn cục
 - Ví dụ:
`int a, b;
int tong, hieu, tich;`

34

1.3. Cấu trúc cơ bản... (tiếp)

- Phần 5: Hàm **main()**
 - Khi thực hiện, chương trình sẽ bắt đầu bằng việc thực hiện các lệnh trong hàm **main()**.
 - Trong hàm **main()** có thể có lệnh gọi tới các hàm khác.
- Phần 6: Nội dung của các hàm đã khai báo
 - Cài đặt (viết mã) cho các hàm đã khai báo nguyên mẫu ở phần 3.

35

Ví dụ một chương trình C đơn giản

```
/* Chương trình sau sẽ nhập vào từ bàn phím 2 số nguyên và hiển thị ra màn hình tổng, hiệu tích của 2 số nguyên vừa nhập vào */  
#include <stdio.h>  
#include <conio.h>  
void main()  
{  
    // Khai báo các biến trong chương trình  
    int a, b;  
    int tong, hieu, tich;
```

36

Ví dụ một chương trình C đơn giản (tiếp)

```
// Nhập vào từ bàn phím 2 số nguyên
printf("\nNhập vào số nguyên thứ nhất: ");
scanf("%d",&a);
printf("\nNhập vào số nguyên thứ hai: ");
scanf("%d",&b);
// Tính tổng, hiệu, tích của 2 số vừa nhập
tong = a + b;
hieu = a - b;
tich = a*b;
```

37

Ví dụ một chương trình C đơn giản (tiếp)

```
// Hiển thị các giá trị ra màn hình
printf("\n Tổng của 2 số vừa nhập là
%d", tong);
printf("\n Hiệu của 2 số vừa nhập là
%d", hieu);
printf("\n Tích của 2 số vừa nhập là
%d", tich);
// Đợi người dùng ấn phím bất kỳ để kết thúc
getch();
}
```

38

1.4. Biên dịch chương trình viết bằng ngôn ngữ C

- Biên dịch chương trình viết bằng ngôn ngữ C (xem tài liệu phát)
 - Dùng trình biên dịch Turbo C++ 3.0
 - Cài đặt Turbo C++ 3.0
 - Viết chương trình
 - Sửa đường dẫn tới các thư viện (nếu cần)
 - Biên dịch
 - Chạy chương trình

39

1.5. Bài tập

- Bài tập 1:
Trong các định danh sau, định danh nào là không hợp lệ:
 - MAX_SINH_VIEN
 - CHIEU_CAO
 - ho va ten
 - 1_bien_nao_do
 - so_thuc_1

40

1.5. Bài tập (tiếp)

- Bài tập 2:
Hãy cho biết giá trị của các hằng nguyên sau trong chương trình:
0345, 0x168, 06356, 0xAF04

41

1.5. Bài tập (tiếp)

- Bài tập 3:
Cho biết biểu diễn dưới dạng số thực dấu phẩy tinh của các hằng số thực sau:
 - 535.235 E+3
 - 256.89 E-1
 - 10.103 E-5

42

1.5. Bài tập (tiếp)

- Bài tập 4: Chạy thử hai chương trình sau xem có chương trình nào có lỗi không? Nếu có lỗi thì hãy xem trình biên dịch báo là lỗi gì?
 - Chương trình 1:

```
void main() { }
```
 - Chương trình 2:

```
#include <stdio.h>
#include <conio.h>
void fct() { }
```

43



44

Lời hay ý đẹp

"Sau từ Yêu, Giúp đỡ là một từ đẹp nhất"

45

Chương 2: Kiểu dữ liệu và biểu thức trong C

Nguyễn Hồng Phương
Email: phuongnh-fit@mail.hut.edu.vn
Website: <http://is.hut.edu.vn/~phuongnh>
Bộ môn Hệ thống thông tin
Viện Công nghệ thông tin và Truyền thông
Đại học Bách Khoa Hà Nội

1

Nội dung chương này

- 2.1. Các kiểu dữ liệu chuẩn trong C
- 2.2. Khai báo và sử dụng biến, hằng
- 2.3. Các lệnh vào ra dữ liệu với các biến
- 2.4. Các lệnh vào ra khác
- 2.5. Các phép toán trong C
- 2.6. Biểu thức trong C
- 2.7. Một số toán tử đặc trưng

2

2.1. Các kiểu dữ liệu chuẩn trong C

Kiểu dữ liệu	Ý nghĩa	Kích thước	Miền biểu diễn
unsigned char	Kí tự	1 byte	0 255
char	Kí tự	1 byte	-128 +127
unsigned int	Số nguyên không dấu	2 byte	0 65,535
int	Số nguyên có dấu	2 byte	-32768 +32767

3

2.1. Các kiểu dữ liệu chuẩn trong C (tiếp)

Kiểu dữ liệu	Ý nghĩa	Kích thước	Miền biểu diễn
unsigned long	Số nguyên không dấu	4 bytes	0 ÷ 4,294,967,295
long	Số nguyên có dấu	4 bytes	-2,147,483,648 ÷ 2,147,483,647
float	Số thực dấu phẩy động, độ chính xác đơn	4 bytes	± 3.4E-38 ÷ ± 3.4E+38
double	Số thực dấu phẩy động, độ chính xác kép	8 bytes	± 1.7E-308 ÷ ± 1.7E+308
long double	Số thực dấu phẩy động	10 bytes	± 3.4E-4932 ÷ ± 1.1E+4932

4

2.2. Khai báo và sử dụng biến, hằng

- 2.2.1. Khai báo và sử dụng biến
- 2.2.2. Khai báo và sử dụng hằng

5

2.2.1. Khai báo và sử dụng biến

- Cú pháp khai báo:
`kiểu_dữ_liệu tên_biến;`
Hoặc:
`kiểu_dữ_liệu tên_biến1, ..., tên_biếnN;`
- Ví dụ: Khai báo một biến `x` thuộc kiểu số nguyên 2 byte có dấu (`int`), biến `y, z, t` thuộc kiểu thực 4 byte (`float`) như sau:

```
int x;  
float y, z, t;
```

6

Khai báo và khởi tạo giá trị cho biến

- Cú pháp:
kiểu_dữ_liệu tên_biến = giá_trị_ban_đầu;
Hoặc:
kiểu_dữ_liệu biến₁=giá_trị₁, biến_N=giá_trị_N;
- Ví dụ:

```
int a = 3; // sau lệnh này biến a sẽ có giá trị bằng 3
float x = 5.0, y = 2.6; // sau lệnh này x có giá
// trị 5.0, y có giá trị 2.6
```

7

2.2.2. Khai báo hằng

- Cách 1: Dùng từ khóa **#define**:
 - Cú pháp:
#define tên_hằng giá_trị
 - Ví dụ:
#define MAX_SINH_VIEN 50
#define CNTT "Cong nghe thong tin"
#define DIEM_CHUAN 23.5

8

2.2.2. Khai báo hằng (tiếp)

- Cách 2: Dùng từ khóa **const** :
 - Cú pháp:
const kiểu_dữ_liệu tên_hằng = giá_trị;
 - Ví dụ:

```
const int MAX_SINH_VIEN = 50;
const char CNTT[20] = "Cong nghe thong tin";
const float DIEM_CHUAN = 23.5;
```

9

2.3. Các lệnh vào ra dữ liệu

- Cung cấp 2 hàm vào ra cơ bản:
 - printf()
 - scanf().
- Muốn sử dụng 2 hàm printf() và scanf() ta cần khai báo tệp tiêu đề **stdio.h**:
#include <stdio.h>
Hoặc
#include "stdio.h"

10

2.3.1. Hàm printf

- Mục đích:
 - Hiển thị ra màn hình các loại dữ liệu cơ bản như: Số, kí tự và chuỗi kí tự
 - Một số hiệu ứng hiển thị đặc biệt như xuống dòng, sang trang,...
- Cú pháp:
printf(xâu_định_dạng, [danh_sách_tham_số]);
 - **xâu_định_dạng**: Là chuỗi dùng để qui định cách thức hiển thị dữ liệu ra màn hình máy tính.
 - **danh_sách_tham_số**: Danh sách các biến sẽ được hiển thị giá trị lên màn hình theo cách thức được qui định trong **xâu_định_dạng**.

11

2.3.1. Hàm printf (tiếp)

- Trong **xâu_định_dạng** chứa:
 - Các kí tự thông thường: Được hiển thị ra màn hình.
 - Các nhóm kí tự định dạng: Xác định quy cách hiển thị các tham số trong phần **danh_sách_tham_số**.
 - Các kí tự điều khiển: Dùng để tạo các hiệu ứng hiển thị đặc biệt như xuống dòng ('\n') hay sang trang ('\f')...

12

Hàm printf - ví dụ:

- Chương trình sau

```
#include <conio.h>
#include <stdio.h>
void main()
{
    int a = 5;
    float x = 1.234;
    printf("Hien thi mot so nguyen %d va mot so thuc
    %f", a, x);
    getch();
}
```

- Sẽ cho ra kết quả:

Hien thi mot so nguyen 5 va mot so thuc 1.234000

13

Hàm printf

- Trong ví dụ trên:

- "Hien thi mot so nguyen %d và mot so thuc %f" là **xâu_định_dạng**
- a,x là **danh_sách_tham_số**
- %d dùng để báo cho máy biết rằng cần phải hiển thị tham số kiểu nguyên (biến a)
- %f dùng để báo cho máy cần hiển thị tham số tương ứng (biến x) theo định dạng số thực

14

Hàm printf

- Nhóm kí tự định dạng thứ *k* trong **xâu_định_dạng** dùng để xác định quy cách hiển thị tham số thứ *k* trong **danh_sách_tham_số**.
- Số lượng tham số trong **Danh_sách_tham_số** bằng số lượng nhóm các kí tự định dạng trong **xâu_định_dạng**. Trong ví dụ trên là 2.
- Mỗi nhóm kí tự định dạng chỉ dùng cho một kiểu dữ liệu
Ví dụ: %d dùng cho **kiểu nguyên**
%f dùng cho **kiểu thực**
- Nếu giữa nhóm kí tự định dạng và tham số tương ứng không phù hợp với nhau thì sẽ hiển thị ra kết quả không như ý.

15

Nhóm ký tự định dạng

- Một số nhóm kí tự định dạng phổ biến:

Nhóm kí tự định dạng	Áp dụng cho kiểu dữ liệu	Ghi chú
%d	int, long, char	Hiện thị tham số tương ứng dưới dạng số nguyên có dấu hệ đếm thập phân
%i	int, long, char	Hiện thị tham số tương ứng dưới dạng số nguyên có dấu hệ đếm thập phân
%o	int, long, char	Hiện thị tham số tương ứng dưới dạng số nguyên không dấu trong hệ đếm cơ số 8.

16

Nhóm ký tự định dạng (tiếp)

%u	int, long, char	Hiện thị tham số tương ứng dưới dạng số nguyên không dấu.
%x	int, long, char	Hiện thị tham số tương ứng dưới dạng số nguyên hệ đếm 16 (không có 0x đứng trước), sử dụng các chữ cái a b c d e f
%X	int, long, char	Hiện thị tham số tương ứng dưới dạng số nguyên hệ đếm 16 (không có 0x đứng trước), sử dụng các chữ cái A B C D E F
%e	float, double	Hiện thị tham số tương ứng dưới dạng số thực dấu phẩy động
%f	float, double	Hiện thị tham số tương ứng dưới dạng số thực dấu phẩy tĩnh

17

Nhóm ký tự định dạng (tiếp)

%g	float, double	Hiện thị tham số tương ứng số thực dưới dạng ngắn gọn hơn trong 2 dạng dấu phẩy tĩnh và dấu phẩy động
%c	int, long, char	Hiện thị tham số tương ứng dưới dạng kí tự
%s	char * (xâu kí tự)	Hiện thị tham số tương ứng dưới dạng xâu kí tự

18

Bổ sung về định dạng

- C cho phép đưa thêm một số thuộc tính định dạng dữ liệu khác vào trong chuỗi định dạng như:
 - Độ rộng để hiển thị (độ rộng tối thiểu)
 - Căn lề trái
 - Căn lề phải.

19

Độ rộng hiển thị

- Đối với **số nguyên** hoặc **ký tự** hoặc **xâu ký tự**:

- Có dạng `%md`, với `m` là số nguyên không âm
- Ví dụ: nếu số `a = 1234`

Lệnh:

```
printf("%5d",a);//danh 5 cho de hien thi a  
printf("\n%5d",34);
```

Cho ra kết quả:

□1234

□□□34

(□: kí hiệu cho dấu cách đơn (*space*))

20

Độ rộng hiển thị (tiếp)

- Đối với **số nguyên** hoặc **ký tự** hoặc **xâu ký tự**:

Ví dụ:

```
printf("\n%3d %15s %3c", 1, "nguyen van a", 'g');  
printf("\n%3d %15s %3c", 2, "tran van b", 'k');
```

- Kết quả:

□□1□□□nguyen van a□□g

□□2□□□□tran van b□□k

21

Độ rộng hiển thị (tiếp)

- Đối với số thực: `m`, `n` là 2 số nguyên không âm

`%m.nf`

Báo rằng cần dành:

- `m` vị trí để hiển thị số thực
- `n` vị trí trong `m` vị trí đó để hiển thị phần thập phân.

22

Độ rộng hiển thị (tiếp)

- Định dạng với dữ liệu là số thực (tiếp):

- Ví dụ:

```
printf("\n%f",12.345);  
printf("\n%.2f",12.345);  
printf("\n%8.2f",12.345);
```

- Kết quả:

12.345000

12.35

□□□12.35

23

Độ rộng hiển thị (tiếp)

- Khi số chỗ cần để hiển thị nội dung dữ liệu lớn hơn trong định dạng:

Tự động cung cấp thêm chỗ mới để hiển thị chứ không cắt bớt nội dung của dữ liệu.

- Ví dụ: `a=1000`

```
printf("So a la: %1d", a);
```

- Kết quả:

So a la: 1000

24

Căn lề phải, căn lề trái

■ Căn lề phải:

Khi hiển thị dữ liệu, mặc định C căn lề phải

■ Căn lề trái:

Nếu muốn căn lề trái khi hiển thị dữ liệu ta chỉ cần thêm dấu trừ - vào ngay sau dấu %.

25

Căn lề phải, căn lề trái (tiếp)

■ Ví dụ:

```
printf("\n%-3d %-15s %4.2f %-3c",
      9, "nguyen van a", 8.5, 'g');
printf("\n%-3d %-15s %4.2f %-3c",
      10, "nguyen ha", 6.75, 'k');
```

■ Kết quả:

```
9  nguyen van a   8.50g
10 nguyen ha     6.75k
```

26

2.3.2. Hàm scanf

■ Mục đích:

- Hàm scanf() dùng để nhập dữ liệu từ bàn phím

■ Cú pháp:

```
scanf(xâu_định_dạng, [danh_sách_địa_chỉ]);
```

■ Ví dụ:

```
scanf("%d%f", &a, &b);
```

- Địa chỉ của một biến được viết bằng cách đặt dấu & trước tên biến.

■ Ví dụ:

Các biến có tên là **a**, **x**, **ten_bien**

Thì địa chỉ của chúng lần lượt sẽ là: **&a**, **&x**, **&ten_bien**

27

Hàm scanf (tiếp)

■ Xâu_định_dạng:

- Gồm các ký tự được quy định cho từng loại dữ liệu được nhập vào.
- Ví dụ: Với dữ liệu định nhập vào là kiểu nguyên thì xâu định dạng là: %d

■ Danh_sách_địa_chỉ:

- Bao gồm các địa chỉ của các biến, các địa chỉ này được phân tách nhau bởi dấu phẩy (,)
- Danh_sách_địa_chỉ phải phù hợp với các nhóm ký tự định dạng trong xâu_định_dạng về:
 - Số lượng
 - Kiểu dữ liệu
 - Thứ tự

28

Nhóm ký tự định dạng

Nhóm ký tự định dạng	Ghi chú
%d	Định khuôn dạng dữ liệu nhập vào dưới dạng số nguyên kiểu int
%o	Định khuôn dạng dữ liệu nhập vào dưới dạng số nguyên kiểu int hệ cơ số 8
%x	Định khuôn dạng dữ liệu nhập vào dưới dạng số nguyên kiểu int hệ cơ số 16

29

Nhóm ký tự định dạng (tiếp)

%c	Định khuôn dạng dữ liệu nhập vào dưới dạng ký tự kiểu char
%s	Định khuôn dạng dữ liệu nhập vào dưới dạng xâu ký tự
%f	Định khuôn dạng dữ liệu nhập vào dưới dạng số thực kiểu float
%ld	Định khuôn dạng dữ liệu nhập vào dưới dạng số nguyên kiểu long
%lf	Định khuôn dạng dữ liệu nhập vào dưới dạng số thực kiểu double

30

Ví dụ:

```
#include <conio.h>
#include <stdio.h>
void main()
{
    // khai bao bien
    int a;
    float x;
    char ch;
    char* str;
    // Nhap du lieu
    printf("Nhap vao mot so
    nguyen");
    scanf("%d",&a);

    printf("\n Nhap vao mot so thuc");
    scanf("%f",&x);
    printf("\n Nhap vao mot ki tu");
    fflush(stdin); scanf("%c",&ch);
    printf("\n Nhap vao mot xau ki
    tu");
    fflush(stdin); scanf("%s",str);
    // Hien thi du lieu vua nhap vao
    printf("\n Nhung du lieu vua nhap
    vao");
    printf("\n So nguyen: %d",a);
    printf("\n So thuc : %.2f",x);
    printf("\n Ki tu: %c",ch);
    printf("\n Xau ki tu: %s",str);
}
```

31

Kết quả:

```
Nhap vao mot so nguyen: 2007
Nhap vao mot so thuc: 18.1625
Nhap vao mot ki tu: b
Nhap vao mot xau ki tu: ngon ngu lap trinh C
Nhung du lieu vua nhap vao
So nguyen: 2007
So thuc: 18.16
Ki tu: b
Xau ki tu: ngon
```

32

Một số quy tắc cần lưu ý

- **Quy tắc 1:** Khi đọc số
 - Hàm `scanf()` quan niệm rằng mọi kí tự số, dấu chấm (.) đều là kí tự hợp lệ.
 - Khi gặp các dấu phân cách như tab, xuống dòng hay dấu cách (space bar) thì `scanf()` sẽ hiểu là kết thúc nhập dữ liệu cho một số
- **Quy tắc 2:** Khi đọc ký tự:
Hàm `scanf()` cho rằng mọi kí tự có trong bộ đệm của thiết bị vào chuẩn đều là hợp lệ, kể cả các kí tự tab, xuống dòng hay dấu cách.

33

Một số quy tắc cần lưu ý (tiếp)

- **Quy tắc 3:** Khi đọc xâu kí tự:
 - Hàm `scanf()` nếu gặp các kí tự dấu trắng, dấu tab hay dấu xuống dòng thì nó sẽ hiểu là kết thúc nhập dữ liệu cho một xâu kí tự.
 - Trước khi nhập dữ liệu kí tự hay xâu kí tự ta nên dùng lệnh `fflush(stdin)` để xóa bộ đệm.

34

2.4. Các lệnh vào ra khác

- Hàm `gets()`:
Dùng để nhập vào từ bàn phím một xâu kí tự bao gồm cả dấu cách, điều mà hàm `scanf()` không làm được.
- Cú pháp:
`gets(xâu_kí_tự);`
- Ví dụ:

```
char* str;
puts("Nhap vao mot xau ki tu:");
fflush(stdin); gets(str);
```

35

Các lệnh vào ra khác (tiếp)

- Hàm `puts()`:
Hiện thị ra màn hình nội dung xâu_kí_tự và sau đó đưa con trỏ xuống dòng mới.
- Cú pháp:
`puts(xâu_kí_tự);`
- Ví dụ:
`puts("Nhap vao xau ki tu:");`
- Tương đương với lệnh: `printf("%s\n","Nhap vao xau ki tu:");`

36

Các lệnh vào ra khác (tiếp)

- Hàm `getch()`: thường dùng để chờ người sử dụng ấn một phím bất kì rồi sẽ kết thúc chương trình.
- Cú pháp

`getch()` ;

- Để sử dụng các hàm `gets()`, `puts()`, `getch()` ta cần khai báo tệp tiêu đề `conio.h`.

37

Các lệnh vào ra khác - ví dụ:

```
#include <conio.h>
#include <stdio.h>
void main()
{
    // khai bao bien
    char* str;
    // Nhap du lieu
    puts("Nhap vao mot xau
    ki tu:");
    fflush(stdin);
    gets(str);

    // Hien thi du lieu vua nhap
    vao
    puts("Xau vua nhap vao: ");
    puts(str);
    puts("An phim bat ki de ket
    thuc...");
    getch();
}
```

38

Các lệnh vào ra khác - kết quả:

Nhap vao mot xau ki tu:
ngon ngu lap trinh C
Xau vua nhap vao:
ngon ngu lap trinh C
An phim bat ki de ket thuc ...

39

2.5. Các phép toán trong C

- Bao gồm:
 - Nhóm các phép toán số học
 - Nhóm các phép toán thao tác trên bit
 - Nhóm các phép toán quan hệ
 - Nhóm các phép toán logic.
 - Ngoài ra C còn cung cấp một số phép toán khác nữa như phép gán, phép lấy địa chỉ...

40

2.5.1. Phép toán số học

Toán tử	Ý nghĩa	Kiểu dữ liệu của toán hạng	Ví dụ
-	Phép đổi dấu	Số thực hoặc số nguyên	<code>int a, b;</code> <code>-12; -a; -25.6;</code>
+	Phép toán cộng	Số thực hoặc số nguyên	<code>float x, y;</code> <code>5 + 8; a + x;</code> <code>3.6 + 2.9;</code>
-	Phép toán trừ	Số thực hoặc số nguyên	<code>3 - 1.6; a - 5;</code>
*	Phép toán nhân	Số thực hoặc số nguyên	<code>a * b; b * y;</code> <code>2.6 * 1.7;</code>
/	Phép toán chia	Số thực hoặc số nguyên	<code>10.0/3.0;</code> (bảng 3.33...) <code>10/3.0;</code> (bảng 3.33...) <code>10.0/3;</code> (bảng 3.33...)
/	Phép chia lấy phần nguyên	Giữa 2 số nguyên	<code>10/3;</code> (bảng 3)
%	Phép chia lấy phần dư	Giữa 2 số nguyên	<code>10%3;</code> (bảng 1)

41

2.5.2. Các phép toán trên bit

Toán tử	Ý nghĩa	Kiểu dữ liệu của toán hạng	Ví dụ
&	Phép VÀ nhị phân	2 số nhị phân	<code>0 & 0</code> (có giá trị 0) <code>0 & 1</code> (có giá trị 0) <code>1 & 0</code> (có giá trị 0) <code>1 & 1</code> (có giá trị 1) <code>101 & 110</code> (có giá trị 100)
	Phép HOẶC nhị phân	2 số nhị phân	<code>0 0</code> (có giá trị 0) <code>0 1</code> (có giá trị 1) <code>1 0</code> (có giá trị 1) <code>1 1</code> (có giá trị 1) <code>101 110</code> (có giá trị 111)

42

Các phép toán trên bit (tiếp)

^	Phép HOẶC CÓ LOẠI TRỪ nhị phân	Số nhị phân	$0 \wedge 0$	(có giá trị 0)
			$0 \wedge 1$	(có giá trị 1)
			$1 \wedge 0$	(có giá trị 1)
			$1 \wedge 1$	(có giá trị 0)
			$101 \wedge 110$	(có giá trị 011)
<<	Phép DỊCH TRÁI nhị phân	Số nhị phân	$a \ll n$	(có giá trị $a \cdot 2^n$)
			$101 \ll 2$	(có giá trị 10100)
>>	Phép DỊCH PHẢI nhị phân	Số nhị phân	$a \gg n$	(có giá trị $a / 2^n$)
			$101 \gg 2$	(có giá trị 1)
~	Phép ĐẢO BIT nhị phân (lấy Bù 1)	Số nhị phân	~ 0	(có giá trị 1)
			~ 1	(có giá trị 0)
			~ 110	(có giá trị 001)

43

2.5.3. Các phép toán quan hệ

Toán tử	Ý nghĩa	Ví dụ
>	So sánh lớn hơn giữa 2 số nguyên hoặc thực.	$2 > 3$ (có giá trị 0) $6 > 4$ (có giá trị 1) $a > b$
>=	So sánh lớn hơn hoặc bằng giữa 2 số nguyên hoặc thực.	$6 >= 4$ (có giá trị 1) $x >= a$
<	So sánh nhỏ hơn giữa 2 số nguyên hoặc thực.	$5 < 3$ (có giá trị 0),
<=	So sánh nhỏ hơn hoặc bằng giữa 2 số nguyên hoặc thực.	$5 <= 5$ (có giá trị 1) $2 <= 9$ (có giá trị 1)
=	So sánh bằng nhau giữa 2 số nguyên hoặc thực.	$3 = 4$ (có giá trị 0) $a = b$
!=	So sánh không bằng (so sánh khác) giữa 2 số nguyên hoặc thực.	$5 != 6$ (có giá trị 1) $6 != 6$ (có giá trị 0)

44

2.5.4. Phép toán logic

Toán tử	Ý nghĩa	Kiểu dữ liệu của toán hạng	Ví dụ
&&	Phép VÀ LOGIC. Biểu thức VÀ LOGIC bằng 1 khi và chỉ khi cả 2 toán hạng đều bằng 1	Hai biểu thức logic	$3 < 5 \ \&\& \ 4 < 6$ (có giá trị 1) $2 < 1 \ \&\& \ 2 < 3$ (có giá trị 0) $a > b \ \&\& \ c < d$
	Phép HOẶC LOGIC. Biểu thức HOẶC LOGIC bằng 0 khi và chỉ khi cả 2 toán hạng bằng 0.	Hai biểu thức logic	$6 0$ (có giá trị 1) $3 < 2 3 < 3$ (có giá trị 0) $x >= a x == 0$
!	Phép PHỤ ĐINH LOGIC một ngôi. Biểu thức PHỤ ĐINH LOGIC có giá trị bằng 1 nếu toán hạng bằng 0 và có giá trị bằng 0 nếu toán hạng bằng 1	Biểu thức logic	$!3$ (có giá trị 0) $!(2 > 5)$ (có giá trị 1)

45

2.5.5. Phép gán

- Cú pháp
tên_biến = biểu_thức;
- Lấy giá trị của *biểu_thức* gán cho *tên_biến*
- Ví dụ:
`int a, b, c;`
`a = 3;`
`b = a + 5;`
`c = a * b;`

46

Phép gán (tiếp)

- Biểu thức gán là biểu thức nên nó cũng có giá trị.
- Giá trị của biểu thức gán bằng giá trị của *biểu_thức*:
→ Có thể gán giá trị của biểu thức gán cho một biến khác hoặc sử dụng như một biểu thức bình thường
- Ví dụ:
`int a, b, c;`
`a = b = 2007;`
`c = (a = 20) * (b = 30);`

47

Phép gán (tiếp)

- Phép toán gán thu gọn:
 - `x = x + y;` giống như
 - `x += y;`
- Dạng lệnh gán thu gọn này còn áp dụng được với các phép toán khác: `+`, `-`, `*`, `/`, `%`, `>>`, `<<`, `&`, `|`, `^`

48

2.5.6. Thứ tự ưu tiên các phép toán

Mức	Các toán tử	Trật tự kết hợp
1	() [] . -> ++ (hậu tố) -- (hậu tố)	----->
2	! ~ ++ (tiền tố) -- (tiền tố) - *	----->
	& sizeof	----->
3	* / %	----->
4	+ -	----->
5	<< >>	----->
6	< <= > >=	----->
7	== !=	----->
8	&	----->
9	^	----->
10	!	----->
11	&&	----->
12		----->
13	?:	----->
14	= += -=	----->

Nguyên tắc xác định trật tự thực hiện các phép toán

- Biểu thức con trong ngoặc được tính toán trước các phép toán khác
- Phép toán một ngôi đứng bên trái toán hạng được kết hợp với toán hạng đi liền nó.
- Nếu toán hạng đứng cạnh hai toán tử thì có 2 khả năng là
 - Nếu hai toán tử có độ ưu tiên khác nhau thì toán tử nào có độ ưu tiên cao hơn sẽ kết hợp với toán hạng
 - Nếu hai toán tử cùng độ ưu tiên thì dựa vào trật tự kết hợp của các toán tử để xác định toán tử được kết hợp với toán hạng.

50

2.6. Biểu thức trong C

- 2.6.1. Các loại biểu thức
- 2.6.2. Sử dụng biểu thức

51

2.6.1. Các loại biểu thức

- Biểu thức số học:
 - Là biểu thức mà giá trị của nó là cái đại lượng số học (số nguyên, số thực).
 - Các toán tử là các phép toán số học (cộng, trừ, nhân, chia...), các toán hạng là các đại lượng số học (số, biến, hằng).
 - Ví dụ:
 - a, b, c là các biến thuộc một kiểu dữ liệu số nào đó.
 - $3 * 3.7$
 - $8 + 6/3$
 - $a + b - c...$

52

Các loại biểu thức (tiếp)

- Biểu thức logic:
 - Là biểu thức mà giá trị của nó là các giá trị logic, tức là một trong hai giá trị: Đúng (*TRUE*) hoặc Sai (*FALSE*).
 - Giá trị nguyên khác 0: Đúng (*TRUE*),
 - Giá trị 0: Sai (*FALSE*).
 - Các phép toán logic gồm có
 - AND: VÀ logic, kí hiệu là `&&`
 - OR: HOẶC logic, kí hiệu là `||`
 - NOT: PHỦ ĐỊNH, kí hiệu là `!`

53

Các loại biểu thức (tiếp)

- Biểu thức quan hệ:
 - Là những biểu thức trong đó có sử dụng các toán tử quan hệ so sánh như lớn hơn, nhỏ hơn, bằng nhau, khác nhau...
 - Chỉ có thể nhận giá trị là một trong 2 giá trị Đúng (*TRUE*) hoặc Sai (*FALSE*)
 - Biểu thức quan hệ là một trường hợp riêng của biểu thức logic.

54

Các loại biểu thức (tiếp)

- Ví dụ về biểu thức quan hệ:
 - `5 > 7` // có giá trị logic là sai, FALSE
 - `9 != 10` // có giá trị logic là đúng, TRUE
 - `2 >= 2` // có giá trị logic là đúng, TRUE
 - `a > b` // giả sử a, b là 2 biến kiểu int
 - `a+1 > a` // có giá trị đúng, TRUE

55

Các loại biểu thức (tiếp)

- Ví dụ về biểu thức logic:
 - `(5 > 7) && (9 != 10)` // có giá trị logic là sai, FALSE
 - `0 || 1` // có giá trị logic là đúng, TRUE
 - `(5 > 7) || (9 != 10)` // có giá trị logic là đúng, TRUE
 - `0` // có giá trị logic là sai, FALSE
 - `10` // phù định của 0, có giá trị logic là đúng, TRUE
 - `3` // có giá trị logic là đúng, TRUE
 - `!3` // phủ định của 3, có giá trị logic là sai, FALSE
 - `(a > b) && (a < b)` // Có giá trị sai, FALSE. Giả sử a, b là 2 biến kiểu int

56

2.6.2. Sử dụng biểu thức

- Làm về phải của lệnh gán.
- Làm toán hạng trong các biểu thức khác.
- Làm tham số thực trong lời gọi hàm.
- Làm chỉ số trong các cấu trúc lặp `for`, `while`, `do while`.
- Làm biểu thức kiểm tra trong các cấu trúc rẽ nhánh `if`, `switch`.

57

2.7. Một số toán tử đặc trưng

- 2.7.1. Các phép toán tăng giảm một đơn vị
- 2.7.2. Phép toán lấy địa chỉ biến
- 2.7.3. Phép toán chuyển đổi kiểu bắt buộc
- 2.7.4. Biểu thức điều kiện

58

2.7.1. Các phép toán tăng giảm một đơn vị

- Tăng hoặc giảm một đơn vị cho biến:
 - `<tên biến> = <tên biến> + 1;`
→ `<tên biến> ++`
 - `<tên biến> = <tên biến> - 1;`
→ `<tên biến> --`
 - Ví dụ:


```
int a = 5;
float x = 10;
a ++; // tương đương với a = a + 1;
x --; // tương đương với x = x - 1;
```

59

Các phép toán tăng giảm một đơn vị (tiếp)

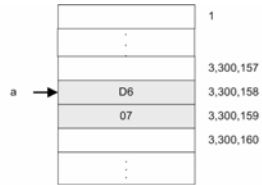
- Tiền tố: Thay đổi giá trị của biến trước khi sử dụng
- Hậu tố: Tính toán giá trị của biểu thức bằng giá trị ban đầu của biến, sau đó mới thay đổi giá trị của biến
- Ví dụ:


```
int a, b, c;
a = 3; // a bằng 3
b = a++; // Dang hau to
// b bằng 3; a bằng 4
c = ++b; // Dang tien to
// b bằng 4, c bằng 4;
```

60

2.7.2. Phép toán lấy địa chỉ biến (&)

- `&<tên biến>;`
 - Ví dụ: a là kiểu dữ liệu int (2 bytes)
- ```
&a; // có giá trị là 3,300,159 hay 0032:5B3F
```



61

## 2.7.3. Phép toán chuyển đổi kiểu bắt buộc

- `(<kiểu dữ liệu mới>) <biểu thức>;`
- Chương trình dịch sẽ tự động chuyển đổi kiểu
  - Số nguyên **int** → **long int**
  - Số **long int** → **float**...
- Ngược lại
  - Số nguyên **long int** 50,000 không phải là một số nguyên kiểu **int** vì phạm vi biểu diễn của kiểu **int** là từ (-32,768 đến 32,767).
  - → Phải ép kiểu

62

## Phép toán chuyển đổi kiểu bắt buộc (tiếp)

- Ví dụ:
- ```
#include <stdio.h>
#include <conio.h>
void main()
{
    long int li; int i; float f; clrscr();
    li = 0x12345678; f = 123.456;
    i = (int) li;
    printf("\n li = %ld", li);
    printf("\n i = %d", i); i = (int) f;
    printf("\n f = %f", f);
    printf("\n i = %d", i);
    getch();
}
```

63

Phép toán chuyển đổi kiểu bắt buộc (tiếp)

- Kết quả
 - li = 1193046
 - i = 13398
 - f = 123.456001
 - i = 123
 - C hỗ trợ chuyển kiểu tự động trong những trường hợp sau
- ```
char int long int float
double long double
```

64

## 2.7.4. Biểu thức điều kiện

- Cú pháp
    - `biểu_thức_1 ? biểu_thức_2 : biểu_thức_3`
    - Giá trị của biểu thức điều kiện
      - Giá trị của biểu\_thức\_2 nếu biểu\_thức\_1 có giá trị khác 0 (tương ứng với giá trị logic ĐÚNG),
      - Ngược lại: Giá trị của biểu\_thức\_3 nếu biểu\_thức\_1 có giá trị bằng 0 (tương ứng với giá trị logic SAI).
  - Ví dụ:
- ```
float x, y, z; // khai báo biến
x = 3.8; y = 2.6; // gán giá trị cho các biến x, y
z = (x < y) ? x : y; // z sẽ có giá trị bằng giá trị
// nhỏ nhất trong 2 số x và y
```

65

Bài tập

- Bài 1
 - Viết chương trình tính chu vi và diện tích hình tròn theo giá trị bán kính r nhập từ bàn phím.
- Bài 2
 - Nhập phần thực x và phần ảo y của một số phức, sau đó in ra theo mẫu: (x,y) với các yêu cầu số in ra có 2 chữ số sau dấu chấm thập phân.

66

Bài tập

Bài 3

- Tính giá trị của biểu thức sau và đưa ra giá trị của biến a, b, c sau khi thực hiện phép toán:

- $a++ + ++a$
- $--a - b-- * ++c$

- với a,b,c là các số nguyên nhập vào từ bàn phím

Bài 4

- Dùng biểu thức điều kiện ? : để đưa ra số lớn nhất trong 3 số thực nhập vào từ bàn phím

67



68

Lời hay ý đẹp

"Tình bạn chân thực là cây sồi và dây thường xuân. Đứng cùng đứng, ngã cùng ngã"

69

Lời giải bài tập chương 2: Kiểu dữ liệu và biểu thức trong C

Nguyễn Hồng Phương
Email: phuongnh-fit@mail.hut.edu.vn
Website: <http://is.hut.edu.vn/~phuongnh>
Bộ môn Hệ thống thông tin
Viện Công nghệ thông tin và Truyền thông
Đại học Bách Khoa Hà Nội

1

Bài 1

- Viết chương trình tính chu vi và diện tích hình tròn theo giá trị bán kính r nhập từ bàn phím.
- Công thức toán học:
 - Chu vi đường tròn: $cv = 2 \times \pi \times r$
 - Diện tích hình tròn: $dt = \pi \times r^2$
- Lời giải

2

Bài 1 (tiếp)

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
    float r, cv, dt; // Khai bao 3 bien thuc
    printf("\n Nhap vao ban kinh r = ");
    scanf("%f",&r);
    cv = 2*M_PI*r; // Tinh chu vi
    dt = M_PI*r*r; // Tinh dien tich
    printf("\nChu vi = %10.2f\nDien tich = %10.2f",cv, dt);
    getch();
}
```

3

Bài 2

- Nhập phần thực x và phần ảo y của một số phức, sau đó in ra theo mẫu: (x,y) với các yêu cầu số in ra có 2 chữ số sau dấu chấm thập phân.

4

Bài 2 (tiếp)

```
#include<stdio.h>
#include<conio.h>
void main()
{
    float x, y; // Khai bao 2 bien thuc
    printf("\nNhapphan thuc = "); scanf("%f",&x);
    printf("\nNhapphan ao = "); scanf("%f",&y);
    printf("\n(%0.2f,%0.2f)",x,y);
    getch();
}
```

5

Bài 3

- Tính giá trị của biểu thức sau và đưa ra giá trị của biến a, b, c sau khi thực hiện phép toán:
 - $a++ + ++a$
 - $--a - b-- * ++c$
- với a, b, c là các số nguyên nhập vào từ bàn phím

6

Bài 3 (tiếp)

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a, b, c, z;
    printf("Nhap vao a = ");scanf("%d",&a);
    printf("\nNhap vao b = ");scanf("%d",&b);
    printf("\nNhap vao c = ");scanf("%d",&c);
    z = a++ + ++a;
    printf("\nBieu thuc 1 = %d, a = %d",z,a);
    z = --a - b-- * ++c;
    printf("\nBieu thuc 2 = %d, a = %d, b = %d, c = %d",z,a,b,c);
    getch();
}
```

7

Bài 3 - kết quả

8

Bài 4

- Dùng biểu thức điều kiện ? : để đưa ra số lớn nhất trong 3 số thực nhập vào từ bàn phím

9

Bài 4 (tiếp)

```
#include<stdio.h>
#include<conio.h>
void main()
{
    float a, b, c, z;
    printf("\nNhap vao so a = ");scanf("%f",&a);
    printf("\nNhap vao so b = ");scanf("%f",&b);
    printf("\nNhap vao so c = ");scanf("%f",&c);
    z = (a>b)?a:b;
    z = (z>c)?z:c;
    printf("\nSo lon nhat = %0.2f\n",z);
    getch();
}
```

10

Bài 4 - Kết quả

11

Bài 4 (tiếp)

```
#include<stdio.h>
#include<conio.h>
void main()
{
    float a, b, c, z;
    printf("\nNhap vao so a = ");scanf("%f",&a);
    printf("\nNhap vao so b = ");scanf("%f",&b);
    printf("\nNhap vao so c = ");scanf("%f",&c);
    z = (a>b)?a:b;
    z = (z>c)?z:c;
    printf("\nSo lon nhat = %0.2f\n",z);
    getch();
}
```

Liệu có viết gọn trên một dòng được không?

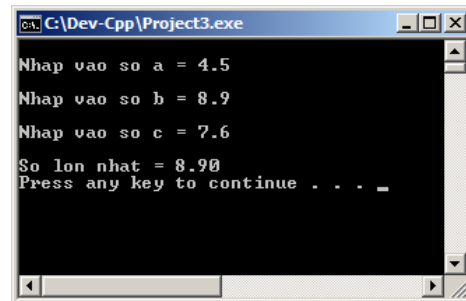
12

Bài 4 (tiếp)

```
#include<stdio.h>
#include<conio.h>
void main()
{
    float a, b, c, z;
    printf("\nNhap vao so a = ");scanf("%f",&a);
    printf("\nNhap vao so b = ");scanf("%f",&b);
    printf("\nNhap vao so c = ");scanf("%f",&c);
    z = (a>b)?(a>c?a:c):(b>c?b:c);
    printf("\nSo lon nhat = %0.2f\n",z);
    getch();
}
```

13

Bài 4 - kết quả (tiếp)



```
C:\Dev-Cpp\Project3.exe
Nhap vao so a = 4.5
Nhap vao so b = 8.9
Nhap vao so c = 7.6
So lon nhat = 8.90
Press any key to continue . . . _
```

14

Hỏi - đáp



15

Chương 3: Các cấu trúc lập trình trong ngôn ngữ C

Nguyễn Hồng Phương
Email: phuongnh-fit@mail.hut.edu.vn
Website: <http://is.hut.edu.vn/~phuongnh>
Bộ môn Hệ thống thông tin
Viện Công nghệ thông tin và Truyền thông
Đại học Bách Khoa Hà Nội

1

Nội dung chương này

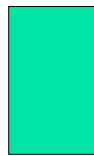
- 3.1. Cấu trúc lệnh khối
- 3.2. Cấu trúc rẽ nhánh
 - 3.2.1. Cấu trúc if, if ... else
 - 3.2.2. Cấu trúc lựa chọn switch
- 3.3. Cấu trúc lặp
 - 3.3.1. Vòng lặp for
 - 3.3.2. Vòng lặp while
- 3.4. Các lệnh thay đổi cấu trúc lập trình
 - 3.4.1. continue
 - 3.4.2. break

2

3.1. Cấu trúc lệnh khối

- Thể hiện cấu trúc tuần tự
- Lệnh khối là dãy các câu lệnh được đặt trong cặp dấu ngoặc nhọn { }

```
{  
    lenh_1;  
    lenh_2;  
    ...  
    lenh_n;  
}
```



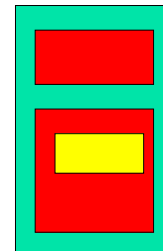
- C cho phép khai báo biến trong lệnh khối, nhưng phần khai báo phải nằm trước câu lệnh.

3

3.1. Cấu trúc lệnh khối (tiếp)

- Lệnh khối lồng nhau:
 - Trong một lệnh khối có thể chứa lệnh khối khác
 - Sự lồng nhau là không hạn chế

```
{  
    lenh;  
    {  
        lenh;  
        ...  
    }  
    ...  
}
```



4

3.1. Cấu trúc lệnh khối - ví dụ

```
#include <conio.h>  
#include <stdio.h>  
void main()  
//Nội dung của hàm main() cũng  
//là một khối lệnh  
{  
    // khai báo biến  
    int c;  
    c = 10;  
    printf("Giá trị của c = %d  
    đây là c ngoài",c);  
    // bắt đầu một khối lệnh  
    khác  
    {  
        int c;  
        c = 10;  
        printf("\n Giá trị của c = %d  
        đây là c trong",c);  
        printf("\n Tang giá trị của c  
        thêm 10 đơn vị");  
        c = c + 10;  
        printf("\n Giá trị của c  
        = %d đây là c trong",c);  
    }  
    printf("\n Giá trị của c =  
    %d đây là c ngoài",c);  
    getch();  
} // kết thúc khối lệnh của hàm  
main()
```

5

3.1. Cấu trúc lệnh khối - ví dụ (tiếp)

- Kết quả:
 - Giá trị của c = 10 đây là c ngoài
 - Giá trị của c = 10 đây là c trong
 - Tang giá trị của c thêm 10 đơn vị
 - Giá trị của c = 20 đây là c trong
 - Giá trị của c = 10 đây là c ngoài

6

3.2. Cấu trúc rẽ nhánh

- 3.2.1. Cấu trúc if, if...else
- 3.2.2. Cấu trúc lựa chọn switch

7

3.2.1. Cấu trúc if, if ... else

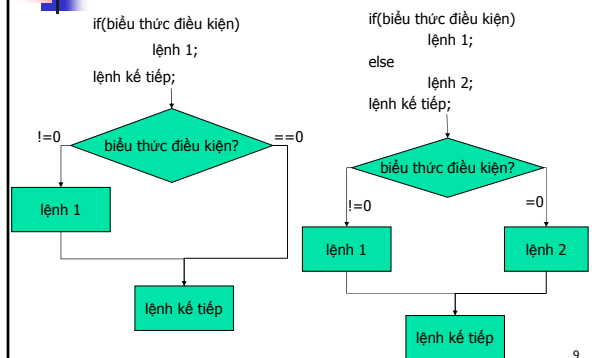
- Cú pháp cấu trúc `if`

```
if (bieu_thuc_dieu_kien)
    lenh;
```
- Cú pháp cấu trúc `if ... else`

```
if (bieu_thuc_dieu_kien)
    lenh_1;
else
    lenh_2;
```

8

3.2.1. Cấu trúc if, if ... else (tiếp)



9

3.2.1. Cấu trúc if, if ... else (tiếp)

- Ví dụ: Bài toán tìm số lớn nhất trong 2 số thực a và b :

```
#include <conio.h>
#include <stdio.h>
void main()
{
    // khai bao bien
    float a, b;
    float max;
    printf("Nhap gia tri a va b: ");
    scanf("%f %f",&a,&b);
    if(a<b)
        max = b;
    else
        max = a;
    printf("\n So lon nhat trong 2 so %.0f va %.0f la %.0f ",a,b,max);
    getch();
} //ket thuc ham main()
```

■ Kết quả:
Nhap vao 2 gia tri a va b: 23 247
So lon nhat trong hai so 23 va 247 la 247

10

3.2.1. Cấu trúc if, if ... else - Luyện tập

- Lập chương trình tìm số nhỏ nhất trong ba số thực nhập vào từ bàn phím.
- Viết chương trình nhập vào một số nguyên từ bàn phím, nếu là số chẵn thì hiển thị "Ban vua nhap so chan", nếu là số lẻ thì hiển thị "Ban vua nhap so le".
- Nhập vào 3 số thực, kiểm tra xem nó có tạo thành 3 cạnh của tam giác không?

11

3.2.2. Cấu trúc lựa chọn switch

- Cú pháp cấu trúc `switch`

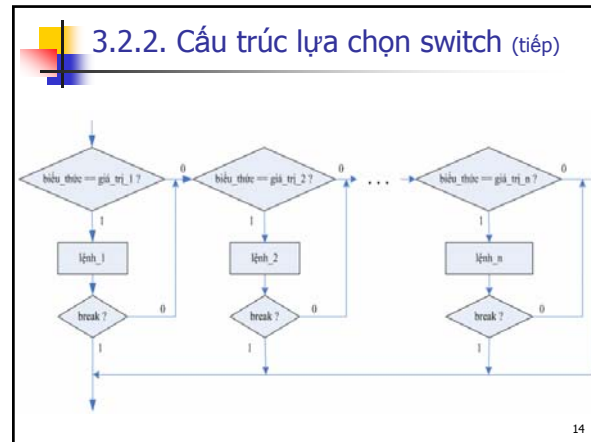
```
switch (bieu_thuc)
{
    case gia_tri_1: lenh_1; [break];
    case gia_tri_2: lenh_2; [break];
    ...
    case gia_tri_n: lenh_n; [break];
    [default: lenh_n+1; [break];]
}
```

12

3.2.2. Cấu trúc lựa chọn switch (tiếp)

- Giá trị của biểu thức kiểm tra (bieu_thuc) phải là số nguyên:
 - Phải có kiểu dữ liệu là **char**, **int**, **long**.
- Tương ứng các giá trị sau case (gia_tri_1, gia_tri_2,...) cũng phải là số nguyên.

13



3.2.2. Cấu trúc lựa chọn switch (tiếp)

- Quan sát ví dụ:

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int n;
    printf("Hay cho toi mot so nguyen");
    scanf("%d",&n);
    switch(n)
    {
        case 0: printf("\n So khong");
                break;
        case 1: printf("\nSo mot");
                break;
        case 2: printf("\nSo hai");
                break;
        default: printf("\nKhong fai 0,1,2");
    }
    getch();
}
    
```

15

3.2.2. Cấu trúc lựa chọn switch (tiếp)

- Trong ví dụ trên, khi chạy chương trình thì sẽ như thế nào?
- Nếu ta bỏ break đi thì sao?

16

3.2.2. Cấu trúc lựa chọn switch (tiếp)

- Kết quả khi chạy chương trình:

17

3.2.2. Cấu trúc lựa chọn switch (tiếp)

- Nếu ta bỏ break đi:

18

3.2.2. Cấu trúc lựa chọn switch-Luyện tập

- Trong một năm các tháng có 30 ngày là 4, 6, 9, 11 còn các tháng có 31 ngày là 1, 3, 5, 7, 8, 10, 12. Riêng tháng hai có thể có 28 hoặc 29 ngày.
- Hãy viết chương trình nhập vào 1 tháng, sau đó đưa ra kết luận tháng đó có bao nhiêu ngày.

19

3.2.2. Cấu trúc lựa chọn switch (tiếp)

```
#include <conio.h>
#include <stdio.h>
int main ()
{
    int thang; clrscr();
    printf("\n Nhập vào thang trong nam
");
    scanf("%d",&thang);
    switch(thang)
    {
        case 1:
        case 3:
        case 5:
        case 7:
        case 8:
        case 10:
        case 12:
            printf("\n Tháng %d có 31 ngày
",thang);
            break;
        case 4:
        case 6:
        case 9:
        case 11:
            printf("\n Tháng %d
có 30 ngày ",thang);
            break;
        case 2:
            printf ("\ Tháng 2 có
28 hoặc 29 ngày");
            break;
        default :
            printf("\n Không có
thang %d", thang);
            break;
    }
    getch();
    return 0;
}
```

20

3.2.2. Cấu trúc lựa chọn switch (tiếp)

```
#include <conio.h>
#include <stdio.h>
int main ()
{
    int thang; clrscr();
    do {
        printf("\n Nhập vào thang trong
nam ");
        scanf("%d",&thang);
    }
    while(thang < 1 || thang > 12);
    switch(thang)
    {
        case 2:
            printf ("\ Tháng 2 có 28 hoặc 29
ngày");
            break;
        case 4:
        case 6:
        case 9:
        case 11:
            printf("\n Tháng %d
có 30 ngày ",thang);
            break;
        default :
            printf("\n Tháng %d
có 31 ngày ", thang);
            break;
    }
    getch();
    return 0;
}
```

21

3.3. Cấu trúc vòng lặp

- 3.3.1. Vòng lặp for
- 3.3.2. Vòng lặp while

22

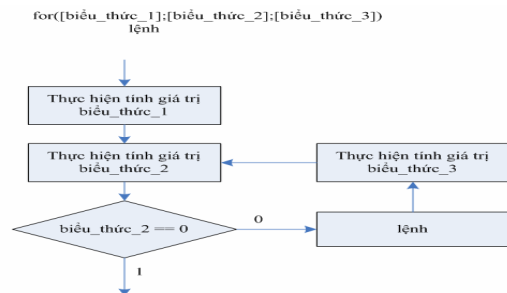
3.3.1. Vòng lặp for

- Mục đích
 - Dùng để thực hiện lặp đi lặp lại một công việc nào đó với số lần lặp xác định.
- Cú pháp:

```
for(bieu_thuc_1;bieu_thuc_2;bieu_thuc_3)
{
    day_cac_lenh;
}
```
- Trong đó:
 - bieu_thuc_1**: Khởi tạo giá trị ban đầu cho vòng lặp
 - bieu_thuc_2**: Điều kiện tiếp tục vòng lặp
 - bieu_thuc_3**: Thực hiện bước tăng của vòng lặp
 - Chú ý các biểu thức 1, 2, 3 có thể có hoặc không

23

3.3.1. Vòng lặp for (tiếp)



24

3.3.1. Vòng lặp for - ví dụ

- Đưa ra màn hình các số nguyên lẻ nhỏ hơn 100

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int i;
    for(i = 1; i < 100; i++) // so vong lap la 99
    {
        if(i%2 == 1) printf("%5d", i);
        if((i+1)%20 == 0) printf("\n");
    }
    getch();
}
```

25

3.3.1. Vòng lặp for - ví dụ (tiếp)

- Đưa ra màn hình các số nguyên lẻ nhỏ hơn 100

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int i;
    for(i = 1; i < 100; i += 2) // so vong lap la 50
    {
        printf("%5d", i);
        if((i+1)%20 == 0) printf("\n");
    }
    getch();
}
```

26

3.3.1. Vòng lặp for - ví dụ (tiếp)

- Kết quả:

```
1   3   5   7   9  11  13  15  17  19
21  23  25  27  29  31  33  35  37  39
41  43  45  47  49  51  53  55  57  59
61  63  65  67  69  71  73  75  77  79
81  83  85  87  89  91  93  95  97  99
```

27

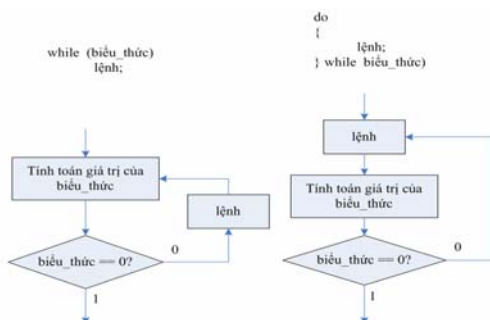
3.3.2. Vòng lặp while

- Mục đích:
 - Dùng để thực hiện lặp đi lặp lại một công việc nào đó với số lần lặp **không** xác định.
- Cú pháp: Có 2 dạng:

```
while (bieu_thuc)
{
    lenh;
}
hoặc
do
{
    lenh;
} while (bieu_thuc);
```

28

3.3.2. Vòng lặp while (tiếp)



29

3.3.2. Vòng lặp while (tiếp)

- while** và **do{...} while**

- while:**
 - Kiểm tra điều kiện vòng lặp (tức là giá trị của biểu thức) trước rồi mới thực hiện lệnh.
 - Các **lenh** sau **while** có thể không được thực hiện lần nào.
- do{...} while:**
 - Thực hiện **lenh** trước rồi mới kiểm tra **dieu_kien** của vòng lặp.
 - Các **lenh** sau **do** được thực hiện ít nhất 1 lần dù **bieu_thuc** có giá trị như thế nào.

30

3.3.2. Vòng lặp while (tiếp)

- Ví dụ: Nhập vào điểm của một sinh viên, nếu điểm đó không $[0, 10]$ thì thông báo cho người dùng nhập lại.
- Cách làm:
 - Nếu dùng lệnh `if` → Chỉ kiểm tra được 1 lần
 - Không dùng `for` được vì chưa biết trước số lần lặp.
 - → Sử dụng vòng lặp `while`

31

3.3.2. Vòng lặp while - ví dụ (tiếp)

```
#include <stdio.h>
#include <conio.h>
void main()
{
    float diem; clrscr();
    printf("Chương trình nhập điểm sinh viên\n");
    printf("Nhập điểm (0<=diem<=10):");
    scanf("%f",&diem);
    while (diem < 0 || diem > 10)
    {
        printf("\nBan nhap khong dung!\n");
        printf("Ban hay nhap lai (0<=diem<=10):");
        scanf("%f",&diem);
    }
    printf("\nDiem ban vua nhap la: %.2f", diem);
    getch();
}
```

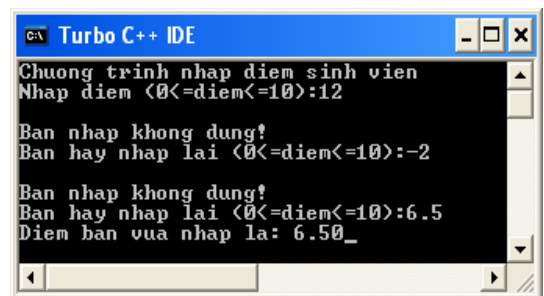
32

3.3.2. Vòng lặp while - ví dụ (tiếp)

```
#include <stdio.h>
#include <conio.h>
void main()
{
    float diem; clrscr();
    printf("Chương trình nhập điểm sinh viên\n");
    do
    {
        printf("Nhập điểm (0<=diem<=10):");
        scanf("%f",&diem);
        if (diem < 0 || diem > 10)
            printf("\nBan nhap khong dung!\n");
    }
    while (diem < 0 || diem > 10);
    printf("\nDiem ban vua nhap la: %.2f", diem);
    getch();
}
```

33

3.3.2. Vòng lặp while - kết quả (tiếp)



```
Turbo C++ IDE
Chương trình nhập điểm sinh viên
Nhập điểm (0<=diem<=10):12

Ban nhap khong dung!
Ban hay nhap lai (0<=diem<=10):-2

Ban nhap khong dung!
Ban hay nhap lai (0<=diem<=10):6.5
Diem ban vua nhap la: 6.50_
```

34

3.4. Các lệnh thay đổi cấu trúc lập trình

- `continue`
- `break`

35

3.4. Các lệnh thay đổi cấu trúc lập trình

- Đối với các lệnh lặp:
 - `while`, `do{...} while`, hoặc `for`
- Thay đổi việc thực hiện lệnh trong vòng lặp → C cung cấp 2 lệnh:
 - `continue;`
 - `break;`

36

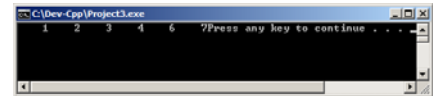
3.4. Các lệnh thay đổi cấu trúc lập trình (tiếp)

- **continue**
 - Bỏ qua việc thực hiện các câu lệnh nằm sau lệnh **continue** trong thân vòng lặp.
 - Chuyển sang thực hiện một vòng lặp mới
- **break**
 - Thoát khỏi vòng lặp ngay cả khi biểu thức điều kiện của vòng lặp vẫn còn được thỏa mãn.

37

3.4. Các lệnh thay đổi cấu trúc lập trình (tiếp)

```
■ Ví dụ:  
#include <stdio.h>  
#include <conio.h>  
void main()  
{  
    int i;  
    for(i = 1;i<=10;i++)  
    {  
        if(i == 5) continue;  
        printf("%5d",i);  
        if(i==7) break;  
    }  
    getch();  
}
```



38

3.4. Các lệnh thay đổi cấu trúc lập trình (tiếp)

- Ví dụ: Nhập vào 1 số nguyên. Kết luận số đó là số nguyên tố hay là hợp số?

39

3.4. Các lệnh thay đổi cấu trúc lập trình (tiếp)

```
#include <stdio.h>  
#include <conio.h>  
#include <math.h>  
void main(){  
    int N, i, OK = 1;  
    printf("\nNhập giá trị N : "); scanf("%d", &N);  
    if (N<2) printf("\nSố %d không là số nguyên tố và  
                    không là hợp số.", N);  
    for (i=2; i<=(int)sqrt(N); i++)  
        if (N%i == 0) {  
            OK = 0;  
            break;  
        }  
    if (OK)  
        printf("\nSố %d là số nguyên tố.", N);  
    else  
        printf("\nSố %d là hợp số.", N);  
    getch();  
}
```

40

Lệnh nhảy goto

- Lệnh này sẽ chuyển điều khiển chương trình tới một nơi nào đó có gán một nhãn để đánh dấu.
- Nói chung, người ta khuyến cáo không nên dùng lệnh nhảy goto.
- Quan sát ví dụ sau:

41

Lệnh nhảy goto (tiếp) - Ví dụ

```
#include <stdio.h>  
#include <conio.h>  
void main()  
{  
    for(int i=0;i<=10;i++)  
    {  
        printf("Vong lap thu %d \n",i);  
        if(i==3) goto thoat; // nhảy tới chỗ có nhãn thoat  
        printf("Ket thuc vong lap %d\n",i);  
    }  
    thoat: printf("Het vong lap for\n");  
    getch();  
}
```

42

Lệnh nhảy goto (tiếp)

- Kết quả:

```
C:\Dev-Cpp\Project3.exe
Vòng lặp thu 0
Ket thuc vong lap 0
Vòng lặp thu 1
Ket thuc vong lap 1
Vòng lặp thu 2
Ket thuc vong lap 2
Vòng lặp thu 3
Het vong lap for
Press any key to continue . . .
```

43

Bài tập

- Bài 1: Lập chương trình giải phương trình bậc hai: $ax^2 + bx + c = 0$ với a, b, c nhập vào từ bàn phím.
- Bài 2: Viết chương trình nhập một số nguyên n từ bàn phím và đưa ra $n!$ (theo 3 cách)
- Bài 3: Viết chương trình nhập một số nguyên dương n từ bàn phím, đưa ra tổng các số từ 1 đến n (theo 3 cách)

44

Bài tập (tiếp)

- Bài 4: Viết chương trình nhập một số nguyên dương n từ bàn phím, đưa ra tổng các số chẵn từ 1 đến n (theo 3 cách)
- Bài 5: Viết chương trình kiểm tra một số nguyên nhập từ bàn phím có phải là số nguyên tố hay không ?
- Bài 6: Lập trình tính tổng
 - $S = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$
 - với n là số nguyên dương nhập từ bàn phím

45

Bài tập (tiếp)

- Bài 7: Viết chương trình nhập vào 2 số nguyên dương từ bàn phím và đưa ra ước số chung lớn nhất của chúng.
- Bài 8: Viết chương trình nhập vào 2 số nguyên từ bàn phím và đưa ra bội số chung nhỏ nhất của chúng.

46



47

Lời hay ý đẹp

"Phẩm cách chân chính của người là ở trong cách họ sống chứ không ở cái họ có"

Blackie

48

Chương 4: Con trỏ và mảng

Nguyễn Hồng Phương
Email: phuongnh-fit@mail.hut.edu.vn
Website: <http://is.hut.edu.vn/~phuongnh>
Bộ môn Hệ thống thông tin
Viện Công nghệ thông tin và Truyền thông
Đại học Bách Khoa Hà Nội

1

Nội dung chương này

- 4.1. Con trỏ và địa chỉ
 - 4.1.1. Tổng quan về con trỏ
 - 4.1.2. Các phép toán làm việc với con trỏ
- 4.2. Mảng
 - 4.2.1. Khái niệm mảng
 - 4.2.2. Khai báo và sử dụng mảng
 - 4.2.3. Các thao tác cơ bản làm việc trên mảng
- 4.3. Sử dụng con trỏ làm việc với mảng

2

Nội dung trình bày

- 4.1. Con trỏ và địa chỉ
 - 4.1.1. Tổng quan về con trỏ
 - 4.1.2. Các phép toán làm việc với con trỏ
- 4.2. Mảng
 - 4.2.1. Khái niệm mảng
 - 4.2.2. Khai báo và sử dụng mảng
 - 4.2.3. Các thao tác cơ bản làm việc trên mảng
- 4.3. Sử dụng con trỏ làm việc với mảng

3

4.1. Con trỏ và địa chỉ

- 4.1.1. Tổng quan về con trỏ
- 4.1.2. Các phép toán làm việc với con trỏ

4

4.1.1. Tổng quan về con trỏ

- a. Địa chỉ và giá trị của một biến
 - Bộ nhớ như một dãy các byte nhớ.
 - Các byte nhớ được xác định một cách duy nhất qua một *địa chỉ*.
 - Biến được lưu trong bộ nhớ.
 - Khi khai báo một biến
 - Chương trình dịch sẽ cấp phát cho biến đó một số ô nhớ liên tiếp đủ để chứa nội dung của biến. Ví dụ một biến số nguyên (int) được cấp phát 2 byte.
 - Địa chỉ của một biến chính là địa chỉ của byte đầu tiên trong số đó.

5

Địa chỉ và giá trị của một biến (tiếp)

- Một biến luôn có hai đặc tính:
 - Địa chỉ của biến.
 - Giá trị của biến.
- Ví dụ:
 - `int i, j;`
 - `i = 3;`
 - `j = i + 1;`

Biến	Địa chỉ	Giá trị
i	FFEC	3
j	FFEE	4

6

b. Khái niệm và khai báo con trỏ

- Con trỏ là một biến mà giá trị của nó là địa chỉ của một vùng nhớ.
- Khai báo con trỏ:
 - Cú pháp khai báo một con trỏ như sau:
Kieu_du_lieu *ten_bien_con_trò;
- Ví dụ
 - `int i = 3;`
 - `int *p;`
 - `p = &i;`
- Một con trỏ chỉ có thể trỏ tới một đối tượng cùng kiểu.



Biến	Địa chỉ	Giá trị
i	FFEC	3
p	FFEE	FFEC

7

Toán tử & và *

- Toán tử **&**: Trả về địa chỉ của biến.
- Toán tử *****: Trả về giá trị chứa trong vùng nhớ được trỏ bởi con trỏ.
- Cả hai toán tử ***** và **&** có độ ưu tiên cao hơn tất cả các toán tử số học ngoại trừ toán tử đảo dấu.
- Ví dụ:

```
void main()
{
    int i = 3; int *p;
    p = &i;
    printf("*p = %d \n", *p);
    getch();
}
```



8

c. Sử dụng biến con trỏ:

- Một biến con trỏ có thể được gán bởi:
 - Địa chỉ của một biến khác:
 - `ten_bien_con_trò = &ten_bien;`
 - Giá trị của một con trỏ khác (tốt nhất là cùng kiểu):
 - `ten_bien_con_trò2 = ten_bien_con_trò1;`
 - Giá trị NULL (số 0):
 - `ten_bien_con_trò = 0; // hoặc là NULL`
- Gán giá trị cho biến con trỏ:
 - `*ten_bien_con_trò = 10;`

9

4.1.1. Tổng quan về con trỏ - ví dụ 1

```
main()
{
    int i = 3, j = 6;
    int *p1, *p2;
    p1 = &i;
    p2 = &j;
    *p1 = *p2;
}
```

biến	địa chỉ	giá trị
i	FFEC	3
j	FFEB	6
p1	FFDA	FFEC
p2	FFDC	FFEB

biến	địa chỉ	giá trị
i	FFEC	6
j	FFEB	6
p1	FFDA	FFEC
p2	FFDC	FFEB

10

4.1.1. Tổng quan về con trỏ - ví dụ 2

```
main()
{
    int i = 3, j = 6;
    int *p1, *p2;
    p1 = &i;
    p2 = &j;
    p1 = p2;
}
```

biến	địa chỉ	giá trị
i	FFEC	3
j	FFEB	6
p1	FFDA	FFEB
p2	FFDC	FFEB

biến	địa chỉ	giá trị
i	FFEC	3
j	FFEB	6
p1	FFDA	FFEB
p2	FFDC	FFEB

11

d. Con trỏ void

- void *ten_bien_con_trò;**
- Con trỏ đặc biệt, không có kiểu,
- Có thể nhận giá trị là địa chỉ của một biến thuộc bất kỳ kiểu dữ liệu nào.
- Ví dụ:
 - `void *p, *q;`
 - `int x = 21;`
 - `float y = 34.34;`
 - `p = &x; q = &y;`

12

4.1.2. Các phép toán làm việc với con trỏ

- Cộng/trừ con trỏ với một số nguyên (`int`, `long`)
→ Kết quả là một con trỏ cùng kiểu
 - `ptr--`; //ptr trỏ đến vị trí của phần tử đứng trước nó.
- Trừ hai con trỏ cho nhau
 - Kết quả là một số nguyên
 - Kết quả này nói lên khoảng cách (số phần tử thuộc kiểu dữ liệu của con trỏ) ở giữa hai con trỏ.
- Các phép toán: Cộng, nhân, chia, lấy số dư trên con trỏ là không hợp lệ.
- Ví dụ: (p2 trỏ đến số nguyên nằm ngay sau x trong bộ nhớ)

```
int x, *p1, *p2;  
p1= &x;  
p2= p1+1;
```

13

Nội dung trình bày

- 4.1. Con trỏ và địa chỉ
 - 4.1.1. Tổng quan về con trỏ
 - 4.1.2. Các phép toán làm việc với con trỏ
- 4.2. Mảng
 - 4.2.1. Khái niệm mảng
 - 4.2.2. Khai báo và sử dụng mảng
 - 4.2.3. Các thao tác cơ bản làm việc trên mảng
- 4.3. Sử dụng con trỏ làm việc với mảng

14

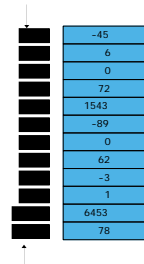
4.2. Mảng

- 4.2.1. Khái niệm mảng
- 4.2.2. Khai báo và sử dụng mảng
- 4.2.3. Các thao tác cơ bản làm việc trên mảng

15

4.2.1. Khái niệm mảng

- Mảng là một tập hợp **hữu hạn** các phần tử có **cùng kiểu dữ liệu** được lưu trữ **liên tiếp** nhau trong bộ nhớ.
- Các phần tử trong mảng có **cùng tên** (và cũng là tên mảng) nhưng phân biệt với nhau ở **chỉ số** cho biết vị trí của chúng trong mảng



16

4.2.2. Khai báo và sử dụng mảng

a. Khai báo:

- Cú pháp:
 - `Kieu_du_lieu ten_mang[kich_thuoc_mang];`
- Ví dụ:
 - `char c[12];`
 - Phần tử đầu tiên có chỉ số 0
 - n phần tử của mảng có tên c:
 - `c[0], c[1]...c[n - 1]`
 - `int mang_nguyen[4];`
 - `float mang_thuc[6];`



17

4.2.2. Khai báo và sử dụng mảng (tiếp)

a. Khai báo (tiếp):

- Mảng nhiều chiều:
 - Mỗi phần tử của mảng cũng là một mảng khác
→ Giống vector trong toán học.
 - Ví dụ:
 - Mảng 2 chiều: `int a[6][5];`
 - Mảng 3 chiều: `int b[3][4][5];`

18

4.2.2. Khai báo và sử dụng mảng (tiếp)

- b. Sử dụng mảng:
 - Truy cập vào 1 phần tử của mảng thông qua tên mảng và chỉ số của phần tử đó.
 - Cú pháp: `ten_mang[chí_số_của_phần_tử]`
 - Ví dụ 1: `int mang_nguyen[3];`
 - `mang_nguyen[0]`: Phần tử thứ 1.
 - `mang_nguyen[1]`: Phần tử thứ 2.
 - `mang_nguyen[2]`: Phần tử thứ 3.

19

4.2.2. Khai báo và sử dụng mảng (tiếp)

- b. Sử dụng mảng (tiếp):
 - Ví dụ 2: `int a[6][5];`
 - `a[0]` là phần tử đầu tiên của mảng, là 1 mảng
→ Phần tử đầu tiên của mảng `a[0]` là `a[0][0]`, ...
 - ...
 - `a[2][3]` sẽ là phần tử thứ 4 của phần tử thứ 3 của `a`.
→ `a[i][j]` sẽ là phần tử thứ `j+1` của `a[i]`, mà phần tử `a[i]` lại là phần tử thứ `i+1` của `a`.

20

4.2.3. Các thao tác cơ bản làm việc trên mảng

a. Nhập dữ liệu cho mảng:

- Nhập dữ liệu cho từng phần tử của mảng
- Ví dụ 1:

```
float a[10]; int i;
scanf("%f", &a[1]);
a[2] = a[1] + 5;
```

- Ví dụ 2:

```
int b[10];
int i;
//Nhập giá trị từ bàn phím cho tất cả các phần tử mảng b
for(i = 0; i < 10; i++)
{
    printf("\n Nhập giá trị cho b[%d]", i);
    scanf("%d", &b[i]);
}
```

21

Nhập dữ liệu cho mảng (tiếp)

- Trường hợp không biết mảng sẽ có bao nhiêu phần tử mà chỉ biết số phần tử tối đa có thể có của mảng. Ví dụ:

```
int a[100]; //Khai báo mảng, số phần tử tối đa là 100
int n; // Ghi lưu giữ số phần tử thực sự của mảng
int i;
printf("\n Cho biết số phần tử của mảng: ");
scanf("%d", &n);
for(i = 0; i < n; i++)
{
    printf("\n a[%d] = ", i);
    scanf("%d", &a[i]);
}
```

22

Nhập dữ liệu cho mảng (tiếp)

- Mảng có thể được khởi tạo giá trị ngay khi khai báo
- ```
int a[4] = {4, 9, 22, 16};
float b[3] = {40.5, 20.1, 100};
char c[5] = {'h', 'e', 'l', 'l', 'o'};
```
- Câu lệnh thứ nhất có tác dụng tương đương với 4 lệnh gán:
- ```
a[0] = 4; a[1] = 9; a[2] = 22; a[3] = 16;
```

23

b. Xuất dữ liệu chứa trong mảng

```
#include <stdio.h>
#include <conio.h>
#define KT 6
void main()
{
    int a[KT];
    int i, k;
    //Nhập giá trị cho các phần tử mảng a từ bàn phím
    for(i = 0; i < KT; i++)
    {
        printf("\n a[%d] = ", i);
        scanf("%d", &a[i]);
    }
    printf("\nBắt đầu hiển thị giá trị các phần tử\n");
    printf("\n a[3] = %d", a[3]);
    //Hiển thị giá trị tất cả các phần tử
    //Mỗi phần tử trên 1 dòng
    for(i = 0; i < KT; i++)
        printf("\n%d", a[i]);
}
```

24

b. Xuất dữ liệu chứa trong mảng

```
printf("\n"); // Xuong dong moi
// Hien thi gia tri cua tat ca cac phan tu mang a
// tren 1 dong, cac phan tu cach nhau 1 dau tab
for(i = 0; i < KT; i++)
    printf("%d\t", a[i]);
// Hien thi k phan tu tren mot dong
printf("\n Cho biet gia tri cua k = ");
scanf("%d",&k);
for(i = 0; i < KT; i++)
{
    printf("%d ", a[i]);
    if((i+1)%k == 0)//xuong dong khi da hien thi k ptu
        printf("\n");
}
```

25

Kết quả

```
a[0] = -2
a[1] = 4
a[2] = 1
a[3] = -11
a[4] = 3
a[5] = 8
Bat dau hien thi gia tri cac phan tu
a[3] = -11
-2
4
1
-11
3
8
-2      4      1      -11      3      8
Cho biet gia tri cua k = 2
-2  4
1  -11
3  8
```

26

c. Tìm các phần tử có giá trị lớn nhất, nhỏ nhất

- `#define KT 100`
- `int a[KT];`
- `max = a[0];`
- Dùng vòng lặp for so sánh max với phần tử có chỉ số từ 1, 2,..., n-1 của mảng a.

27

c. Tìm các phần tử có giá trị lớn nhất, nhỏ nhất

```
int a[100]; int i, n; int max;
printf("\n Cho biet so phan tu cua mang: ");
scanf("%d",&n);
for(i = 0; i < n; i++)
{
    printf("\n a[%d] = ",i);
    scanf("%d",&a[i]);
}
max = a[0]; // Ban dau gia su phan tu lon nhat la a[0]
//Lan luot so sanh voi cac phan tu con lai trong mang
for(i = 1; i < n; i++)
    if(max < a[i]) // gap phan tu co gia tri lon hon
        max = a[i]; // coi phan tu nay la phan tu lon nhat
printf("\n Phan tu lon nhat trong mang la: %d", max);
```

28

4.2.3. Các thao tác cơ bản làm việc trên mảng (tiếp)

- Tìm kiếm trên mảng
- Sắp xếp mảng

29

Nội dung trình bày

- 4.1. Con trỏ và địa chỉ
 - 4.1.1. Tổng quan về con trỏ
 - 4.1.2. Các phép toán làm việc với con trỏ
- 4.2. Mảng
 - 4.2.1. Khái niệm mảng
 - 4.2.2. Khai báo và sử dụng mảng
 - 4.2.3. Các thao tác cơ bản làm việc trên mảng
- 4.3. Sử dụng con trỏ làm việc với mảng

30

4.3.1. Con trỏ và mảng 1 chiều

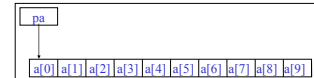
- Xét câu lệnh:
`int a[10];`
- Khai báo này hệ thống sẽ cấp cho mảng a 10 khối gồm 10 khối nhớ liên tiếp: a[0], a[1], a[2], a[3], ..., a[9] (trong trường hợp này mỗi khối nhớ 2 byte).

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
------	------	------	------	------	------	------	------	------	------

31

4.3.1. Con trỏ và mảng 1 chiều (tiếp)

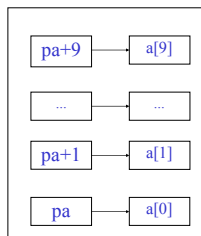
- Có một điều cần nói ở đây là tên mảng là một hằng địa chỉ, nó chính là địa chỉ của phần tử đầu tiên của mảng; điều này có nghĩa là:
a tương đương với &a[0]
- Nếu pa là một con trỏ kiểu nguyên,
`int *pa;`
- Khi đó phép gán
`pa = &a[0];`
- Đem pa trỏ đến phần tử thứ nhất (có chỉ số là 0) của a; nghĩa là pa chứa địa chỉ của a[0].



32

4.3.1. Con trỏ và mảng 1 chiều (tiếp)

- `pa + i` là địa chỉ của `a[i]` và `*(pa+i)` là nội dung của `a[i]`



33

4.3.1. Con trỏ và mảng 1 chiều (tiếp)

```
#define N 5
int tab[N] = {1, 2, 6, 0, 7};
main()
{
    int i; int *p;
    p = tab;
    for (i = 0; i < N; i++)
    {
        printf(" %d \n", *p);
        p++;
    }
}
```

34

4.3.1. Con trỏ và mảng 1 chiều (tiếp)

- Tóm lại, một biểu thức chứa một mảng và một chỉ số tương đương với một cách viết khác sử dụng một con trỏ cùng một độ lệch.
- Có sự khác nhau giữa tên của mảng và con trỏ. Một con trỏ là một biến và vì thế, các câu lệnh `pa=a` và `pa++` là hợp lệ. Nhưng một tên mảng không phải là một con trỏ, do đó các câu lệnh kiểu như `a=pa`; `a++` là không hợp lệ. (Về thực chất, tên mảng là một hằng con trỏ do đó chúng ta không thể thay đổi giá trị của nó được).
- Để minh họa cho các ý tưởng vừa trình bày ta xét một ví dụ sau: viết chương trình thực hiện các công việc sau:
 - Đọc từ bàn phím các phần tử của một mảng.
 - Tính tổng của chúng

35

4.3.1. Con trỏ và mảng 1 chiều (tiếp)

```
#include <stdio.h>
main()
{
    float a[5],s;
    int i;
    for(i=0;i<5;i++)
    {
        printf("\na[%d] = ",i);
        scanf("%f",&a[i]);
    }
    s=0;
    for(i=0;i<5;i++)
        s+=a[i];
    printf("\n tong la: %10.2f",s);
}

#include <stdio.h>
main()
{
    float a[5],s;
    int i;
    for(i=0;i<5;i++)
    {
        printf("\na[%d] = ",i);
        scanf("%f",a+i);
    }
    s=0;
    for(i=0;i<5;i++)
        s+=a[i];
    printf("\n tong la: %10.2f",s);
}
```

36

4.3.1. Con trỏ và mảng 1 chiều (tiếp)

```
#include <stdio.h>
main()
{
    float a[5],s,*p;
    int i; p=a;
    for(i=0;i<5;i++)
    {
        printf("\na[%d]
= ",i);
        scanf("%f",&p[i]);
    }
    s=0;
    for(i=0;i<5;i++)
    s+=p[i];
    printf("\n tong la: %10.2f",s);
}

#include <stdio.h>
main()
{
    float a[5],s,*p;
    int i; p=a;
    for(i=0;i<5;i++)
    {
        printf("\na[%d]
= ",i);
        scanf("%f",p+i);
    }
    s=0;
    for(i=0;i<5;i++)
    s+=*(p+i);
    printf("\n tong la: %10.2f",s);
}
```

37

4.3.2. Con trỏ và mảng nhiều chiều

- Phép toán lấy địa chỉ
- Phép cộng địa chỉ trong mảng hai chiều
- Con trỏ và mảng hai chiều

38

4.3.2. Con trỏ.... - Phép toán lấy địa chỉ

- Phép toán lấy địa chỉ nói chung không dùng được đối với các thành phần của mảng nhiều chiều (trừ trường hợp mảng hai chiều các số nguyên).
- Xét chương trình sau với ý định nhập số liệu cho ma trận thực.

```
#include <stdio.h>
main()
{
    float a[10][20];
    int i,j,n,m;
    printf("Nhap vao kích thước ma trận m,n=");
    scanf("%d%d",&m,&n); /*Chương trình chạy đúng cho đến đây*/
    for(i=0;i<m;i++)
    for(j=0;j<n;j++)
    {
        printf("a[%d][%d] = ",i,j);
        scanf("%f",&a[i][j]);
    }
}
```

39

4.3.2. Con trỏ.... - Phép cộng địa chỉ trong mảng 2 chiều

- Xét khai báo
`float a[2][3];`
- Với khai báo này hệ thống cấp sáu phần tử liên tiếp trong bộ nhớ theo thứ tự sau:
`a[0][0] a[0][1] a[0][2] a[1][0] a[1][1] a[1][2]`
- Cũng như mảng một chiều tên của mảng hai chiều được hiểu như địa chỉ của phần tử đầu tiên của nó; phép cộng địa chỉ phải hiểu như sau: C cho rằng mảng hai chiều là mảng của mảng; như vậy với khai báo trên thì a là mảng mà mỗi phần tử của nó là một dãy gồm 3 số thực. Vì vậy
 - a trỏ tới đầu hàng thứ nhất (phần tử a[0][0])
 - a+1 trỏ tới đầu hàng thứ hai (phần tử a[1][0])

40

4.3.2. Con trỏ.... - Con trỏ và mảng 2 chiều

- Để truy xuất vào các phần tử của mảng hai chiều ta vẫn có thể dùng con trỏ theo cách sau:

```
float *p, a[2][3];
p=(float*)a;
```

- Khi đó

```
p trỏ tới a[0][0]
p+1 trỏ tới a[0][1]
p+2 trỏ tới a[0][2]
p+3 trỏ tới a[1][0]
p+4 trỏ tới a[1][1]
p+5 trỏ tới a[1][2]
```

41

Minh họa cách đọc dữ liệu cho mảng 2 chiều

```
#include <stdio.h>
main()
{
    float a[2][3],*p;
    int i,j,m,n;
    p=(float*)a;
    for(i=0;i<2;i++)
    for(j=0;j<3;j++)
    {
        printf("a[%d][%d] = ",i,j);
        scanf("%f",p+i*3+j);
    }
    for(i=0;i<2;i++)
    for(j=0;j<3;j++)
    {
        printf("%6.2f",a[i][j]);
        if(j==2) printf("\n");
    }
}
```

42

Con trỏ và mảng 2 chiều

- Các bạn để ý rằng, a là một hằng con trỏ trỏ đến các dòng của một ma trận hai chiều, vì vậy
 a trỏ đến dòng thứ nhất
 $a+1$ trỏ đến dòng thứ hai
- Để tính toán được địa chỉ của phần tử ở dòng i cột j chúng ta phải dùng phép chuyển đổi kiểu bắt buộc đối với a : $(float *)a$, đây là con trỏ trỏ đến thành phần $a[0][0]$ của ma trận. Và vì vậy thành phần $a[i][j]$ sẽ có địa chỉ là $(float *)a + i*n + j$ (n là số cột).
- Một cách tổng quát, nếu mảng có kiểu $type$ và có kích thước các chiều tương ứng là n_1, n_2, \dots, n_k (Giả sử mảng a có k chiều). Địa chỉ của thành phần $a[i_1][i_2] \dots [i_k]$ (k chỉ số 0) là $(type *)a$, và địa chỉ của $a[i_1][i_2] \dots [i_k]$ được tính như sau

$$(type *)a + \sum_{j=1}^{k-1} i_j \prod_{l=j+1}^k n_l + i_k$$

43

4.3.3. Mảng các con trỏ

- Trước tiên chúng ta có thể khai báo một mảng con trỏ bằng câu lệnh sau:
`type *pointer_array[size];`
- ví dụ câu lệnh,
`char *ma[10];`
Sẽ khai báo một mảng 10 con trỏ char có thể được dùng để khai báo một mảng để lưu trữ địa chỉ của mười chuỗi ký tự nào đó.
- Nếu các con trỏ được chuẩn bị để chỉ đến một biến nào đó đã có, thì như vậy, chúng ta có thể truy xuất được các biến này thông qua một mảng mà không cần đến vị trí thực sự của các biến đó có liên tiếp hay không.

44

4.3.3. Mảng các con trỏ

- Ví dụ:
 - Xem xét một mảng các con trỏ `ptr_array` được gán các địa chỉ của các biến `int` có giá trị và vị trí bất kỳ.
 - Chúng ta sẽ dùng một hàm để sắp xếp lại các địa chỉ này trong mảng để sao cho các địa chỉ của các số bé được xếp trước địa chỉ của các số lớn hơn. Lúc đó dù chúng ta không làm thay đổi vị trí hoặc thay đổi các giá trị của các biến nhưng mảng vẫn có vẻ như là một mảng chỉ đến các giá trị đã sắp xếp có thứ tự.

45

4.3.3. Mảng các con trỏ - ví dụ (tiếp)

```
#include <stdio.h>
main() {
    int i,j,*x;
    int d=10,e=3,f=7;
    int a=12,b=2,c=6;
    int *ptr_array[6];
    /*Gán các thành phần của mảng*/
    ptr_array[0]=&a;
    ptr_array[1]=&b;
    ptr_array[2]=&c;
    ptr_array[3]=&d;
    ptr_array[4]=&e;
    ptr_array[5]=&f;
```

46

4.3.3. Mảng các con trỏ - ví dụ (tiếp)

```
/*Sắp xếp lại dãy số*/
for(i=0;i<5;i++)
    for(j=i+1;j<6;j++)
        if(*ptr_array[i]>*ptr_array[j])
        {
            x=ptr_array[i];
            ptr_array[i]=ptr_array[j];
            ptr_array[j]=x;
        }
/*In kết quả sau khi sắp xếp*/
for(i=0;i<6;i++)
    printf(" %d \n",*ptr_array[i]);
getch();
}
```

47

4.3.3. Mảng các con trỏ - ví dụ (tiếp)

- Kết quả chạy chương trình
2
3
6
7
10
12

48

4.3.3. Mảng các con trỏ - nhận xét

- việc sử dụng một mảng các con trỏ có nhiều ý niệm gần giống như sử dụng một mảng hai chiều. Ví dụ, nếu các biến n và m được khai báo là:

```
int m[10][9];
```

```
int *n[10];
```

thì cách viết để truy xuất được các phần tử của các mảng này có thể tương tự nhau, chẳng hạn: m[6][5] và n[6][5] đều cho ta một kết quả là một số int.

- Khai báo:

```
char slist[10][100];
```

```
char *plist[10];
```

đều khiến cho C hiểu rằng slist[i] và plist[i] là các con trỏ chỉ đến một đối tượng là char, hoặc là mảng char

49

4.3.3. Mảng các con trỏ - nhận xét

- Tuy vậy, giữa mảng nhiều chiều và mảng các con trỏ cũng tồn tại nhiều điểm khác nhau:

- Mảng nhiều chiều thực sự là mảng có khai báo, do đó có chỗ đầy đủ cho tất cả các phần tử của nó. Còn mảng các con trỏ chỉ mới có chỗ cho các biến con trỏ mà thôi.

- Bên cạnh đó, việc sử dụng mảng các con trỏ có hai ưu điểm, đó là

- Việc truy xuất đến các phần tử là truy xuất gián tiếp thông qua các con trỏ và như vậy, vị trí của các mảng con này có thể là bất kỳ, và chúng có thể là những mảng đã cõ bằng cách xin cấp phát chỗ động hay bằng khai báo biến mảng bình thường, tùy ý.
- Các mảng con của nó được chỉ đến bởi các con trỏ, có thể có độ dài tùy ý, hoặc có thể không có (nếu con trỏ đó không được chuẩn bị, hoặc được gán bằng NULL).

50

4.3.4. Con trỏ trỏ đến con trỏ

- Ví dụ

```
char *monthname[20] =
```

```
{
    "January", "February", "March", "April",
    "May", "June", "July", "August", "September",
    "October", "November", "December"
};
```

```
char **pp;
```

```
pp=monthname; /*Tên mảng là địa chỉ của phần tử đầu tiên*/
```

Khi đó *pp sẽ chỉ đến con trỏ đầu tiên của mảng, con trỏ này lại chỉ đến chuỗi "January" như đã khởi đầu cho mảng.

- Nếu tăng pp lên 1 thì khi đó pp sẽ chỉ đến phần tử kế tiếp của mảng có giá trị là con trỏ đến xâu ký tự "February".v.v.

51

Bài tập

- Bài 1: Nhập vào một dãy số thực, sắp xếp tăng dần (sử dụng mảng một chiều).
- Bài 2: Nhập vào một dãy gồm 6 số nguyên, sắp xếp giảm dần (không sử dụng mảng).

52



53

Lời hay ý đẹp

"Con người ta có ba điều làm lỗi để mắc phải là: chưa đến lượt đã vội nói, điều đáng nói lại không nói và không nhìn về mặt người khác mà đã nói"

Không Từ

54

Chương 5 Xâu ký tự

Nguyễn Hồng Phương
Email: phuongnh-fit@mail.hut.edu.vn
Website: <http://is.hut.edu.vn/~phuongnh>
Bộ môn Hệ thống thông tin
Viện Công nghệ thông tin và Truyền thông
Đại học Bách Khoa Hà Nội

1

Nội dung

- 5.1. Khái niệm xâu ký tự
- 5.2. Khai báo và sử dụng xâu
 - 5.2.1. Khai báo xâu ký tự
 - 5.2.2. Truy cập vào một phần tử của xâu
- 5.3. Một số hàm làm việc với ký tự và xâu ký tự trong C.
 - 5.3.1. Các hàm xử lý ký tự
 - 5.3.2. Các hàm xử lý xâu ký tự
 - 5.3.3. Một số hàm xử lý xâu ký tự khác
 - 5.3.4. Con trỏ và xâu ký tự

2

Nội dung

5.1. Khái niệm xâu ký tự

- 5.2. Khai báo và sử dụng xâu
 - 5.2.1. Khai báo xâu ký tự
 - 5.2.2. Truy cập vào một phần tử của xâu
- 5.3. Một số hàm làm việc với ký tự và xâu ký tự trong C.
 - 5.3.1. Các hàm xử lý ký tự
 - 5.3.2. Các hàm xử lý xâu ký tự
 - 5.3.3. Một số hàm xử lý xâu ký tự khác
 - 5.3.4. Con trỏ và xâu ký tự

3

5.1. Khái niệm xâu ký tự

- Là một dãy các kí tự viết liên tiếp nhau.
- Xâu rỗng: Xâu không gồm kí tự nào cả.
- Độ dài xâu: Số kí tự có trong xâu.
- Ví dụ:
 - "Tin hoc" là một xâu kí tự gồm 7 kí tự: 'T', 'i', 'n', dấu cách (' '), 'h', 'o', và 'c'.
- Lưu trữ xâu ký tự:
 - Ký tự kết thúc xâu: NULL hoặc '\0'.

'T'	'i'	'n'	' '	'h'	'o'	'c'	'\0'
-----	-----	-----	-----	-----	-----	-----	------

4

5.1. Khái niệm xâu ký tự (tiếp)

- Cần phân biệt giữa ký tự và xâu bao gồm một ký tự. Ví dụ 'A' là ký tự A được mã hóa bằng 1 byte, trong khi "A" là một xâu ký tự chứa ký tự A xâu này được mã hóa bằng 2 bytes cho ký tự A và ký tự '\0'.
- Trong C không tồn tại các phép toán so sánh, gán nội dung của xâu này cho xâu khác. Để thực hiện các công việc này C cung cấp cho người lập trình một thư viện các hàm chuẩn, được khai báo trong tệp header có tên là **string.h**.
- Để sử dụng các hàm thao tác xâu, trên đầu chương trình cần phải có dòng khai báo **#include <string.h>**

5

Nội dung

- 5.1. Khái niệm xâu ký tự
- 5.2. Khai báo và sử dụng xâu
 - 5.2.1. Khai báo xâu ký tự
 - 5.2.2. Truy cập vào một phần tử của xâu
- 5.3. Một số hàm làm việc với ký tự và xâu ký tự trong C.
 - 5.3.1. Các hàm xử lý ký tự
 - 5.3.2. Các hàm xử lý xâu ký tự
 - 5.3.3. Một số hàm xử lý xâu ký tự khác
 - 5.3.4. Con trỏ và xâu ký tự

6

5.2.1. Khai báo chuỗi ký tự

- Cú pháp:
 - `char ten_xau[so_ky_tu_toi_da];`
- Ví dụ:
 - `char ho_va_ten[20];`

7

5.2.2. Truy cập vào một phần tử của chuỗi

- Cú pháp:
 - `ten_xau[chi_so_ky_tu_can_truy_cap]`
- Ví dụ:
 - `char que_quan[10] = "Ha noi";`

Phần tử thứ	Chỉ số của phần tử	Tên của phần tử	Nội dung lưu trữ
1	0	que_quan[0]	'H'
2	1	que_quan[1]	'a'
3	2	que_quan[2]	' '
4	3	que_quan[3]	'N'
5	4	que_quan[4]	'o'
6	5	que_quan[5]	'i'
7	6	que_quan[6]	'\0'
8	7	que_quan[7]	
9	8	que_quan[8]	
10	9	que_quan[9]	

8

Nội dung

- 5.1. Khái niệm chuỗi ký tự
- 5.2. Khai báo và sử dụng chuỗi
 - 5.2.1. Khai báo chuỗi ký tự
 - 5.2.2. Truy cập vào một phần tử của chuỗi
- 5.3. Một số hàm làm việc với ký tự và chuỗi ký tự trong C.
 - 5.3.1. Các hàm xử lý ký tự
 - 5.3.2. Các hàm xử lý chuỗi ký tự
 - 5.3.3. Một số hàm xử lý chuỗi ký tự khác
 - 5.3.4. Con trỏ và chuỗi ký tự

9

5.3.1. Các hàm xử lý ký tự

- Để sử dụng các hàm này ta khai báo tệp tiêu đề `#include <ctype.h>`
- `int toupper(int ch)`
 - Chuyển một ký tự chữ cái thường (các ký tự 'a', 'b', ..., 'z') thành ký tự chữ cái hoa tương ứng ('A', 'B', ..., 'Z').
- `int tolower(int ch)`
 - Chuyển một ký tự chữ cái hoa ('A', 'B', ..., 'Z') thành ký tự chữ cái thường tương ứng ('a', 'b', ..., 'z').
- `int isalpha(int ch)`
 - Kiểm tra một ký tự có phải là chữ cái hay không ('a', 'b', ..., 'z', 'A', 'B', ..., 'Z').
 - Hàm trả về giá trị khác không nếu đúng là chữ cái, trả về giá trị 0 nếu ngược lại.

10

5.3.1. Các hàm xử lý ký tự (tiếp)

- `int isdigit(int ch)`
 - Kiểm tra một ký tự có phải là chữ số hay không ('0', '1', ..., '9').
 - Hàm trả về giá trị khác không nếu đúng, trả về giá trị 0 nếu ngược lại.
- `int islower(int ch)`
 - Kiểm tra một ký tự có phải là chữ cái thường hay không ('a', 'b', ..., 'z').
 - Hàm trả về giá trị khác không nếu đúng, trả về giá trị 0 nếu ngược lại.
- `int isupper(int ch)`
 - Kiểm tra một ký tự có phải là chữ cái hoa hay không ('A', 'B', ..., 'Z').
 - Hàm trả về giá trị khác không nếu đúng, trả về giá trị 0 nếu ngược lại.

11

5.3.1. Các hàm xử lý ký tự (tiếp)

- `int iscntrl(int ch)`
 - Kiểm tra một ký tự có phải là ký tự điều khiển hay không (là các ký tự không hiển thị được và có mã ASCII từ 0 đến 31).
 - Hàm trả về giá trị khác không nếu đúng, trả về giá trị 0 nếu ngược lại.
- `int isspace(int ch)`
 - Kiểm tra một ký tự có phải là dấu cách (space, mã ASCII là 32), ký tự xuống dòng ('\n', mã ASCII là 10), ký tự về đầu dòng ('\r', mã ASCII là 13), dấu tab ngang ('\t', mã ASCII là 9) hay dấu tab dọc ('\v', mã ASCII là 11) hay không.
 - Hàm trả về giá trị khác không nếu đúng, trả về giá trị 0 nếu ngược lại.

12

5.3.2. Các hàm xử lý chuỗi ký tự

- Vào ra dữ liệu
 - Vào ra dữ liệu trên chuỗi ký tự tức là nhập dữ liệu cho chuỗi và hiển thị dữ liệu chứa trong chuỗi.
 - Khai báo thư viện <stdio.h>.
 - Để nhập dữ liệu cho chuỗi ta có thể sử dụng 2 hàm `scanf()` hoặc `gets()`
 - `scanf("%s", ten_xau);`
 - `gets(ten_xau);`
 - Để hiển thị nội dung chuỗi ta có thể dùng 2 hàm `printf()` hoặc `puts()`
 - `printf("%s", ten_xau);`
 - `puts(ten_xau);`

13

5.3.2. Các hàm xử lý chuỗi ký tự

- Ví dụ: Nhập vào một chuỗi và hiển thị trên màn hình chuỗi vừa nhập.

```
#include <conio.h>
#include <stdio.h>
#include <string.h>
int main()
{
    char ten[12];
    printf("Nhập chuỗi: "); gets(ten);
    printf("Chuỗi vừa nhập: "); puts(ten);
    getch();
    return 0;
}
```

14

5.3.3. Một số hàm xử lý chuỗi ký tự khác

- Khai báo tệp tiêu đề `#include <string.h>`
- `int strlen(char* tên_xâu);`
 - Trả về độ dài (số ký tự có trong chuỗi) của chuỗi ký tự `tên_xâu`.
- `char* strcpy(char* chuỗi_đích, char* chuỗi_nguồn)`
 - Sao chép nội dung chuỗi nguồn và ghi lên chuỗi đích.
- `char* strncpy(char* chuỗi_đích, char* chuỗi_nguồn, int n)`
 - Tương tự `strcpy()` nhưng ngừng sao chép sau `n` ký tự. Trong trường hợp không có đủ số ký tự trong chuỗi nguồn thì hàm sẽ điền thêm các ký tự trống vào chuỗi đích.

15

5.3.3. Một số hàm xử lý chuỗi ký tự khác

- `int strcmp(char* chuỗi_thứ_nhất, char* chuỗi_thứ_hai);`
 - Trả về
 - giá trị < 0 nếu chuỗi_thứ_nhất nhỏ hơn chuỗi_thứ_hai
 - giá trị 0 nếu chuỗi_thứ_nhất bằng chuỗi_thứ_hai
 - giá trị > 0 nếu chuỗi_thứ_nhất lớn hơn chuỗi_thứ_hai
- `int strncmp(char* chuỗi_thứ_nhất, char* chuỗi_thứ_hai);`
 - Tương tự như hàm `strcmp` nhưng giới hạn việc so sánh với `n` ký tự đầu tiên của hai chuỗi.
- `int stricmp(char* chuỗi_thứ_nhất, char* chuỗi_thứ_hai);`
 - Tương tự như `strcmp()` nhưng không phân biệt chữ hoa hay chữ thường.
- `int strnicmp(char* chuỗi_thứ_nhất, char* chuỗi_thứ_hai);`
 - Tương tự như hàm `strncmp` nhưng giới hạn việc so sánh với `n` ký tự đầu tiên của hai chuỗi.

16

5.3.3. Một số hàm xử lý chuỗi ký tự khác

- `char* strchr(char* str, int ch);`
 - Tìm kiếm vị trí của ký tự `ch` trong chuỗi `str`.
 - Nếu có ký tự `ch` trong `str` thì hàm `strchr()` trả về con trỏ tới ký tự `ch` đầu tiên trong `str`, ngược lại nó sẽ trả về con trỏ NULL.
- `char* strrchr(char* str, int ch);`
 - Tương tự như `strchr()` nhưng việc tìm kiếm bắt đầu từ cuối chuỗi `str`.
- `char* strstr(char* str1, char* str2);`
 - Tìm kiếm vị trí của chuỗi con `str2` trong chuỗi `str1`.
 - Nếu `str2` là chuỗi con của `str1` thì hàm `strstr()` trả về con trỏ tới ký tự đầu tiên của chuỗi con `str2` đầu tiên trong `str1`, ngược lại nó sẽ trả về con trỏ NULL.

17

5.3.3. Một số hàm xử lý chuỗi ký tự khác

- `char* strcat(char* chuỗi_đích, char* chuỗi_nguồn);`
 - Ghép nối chuỗi nguồn vào ngay sau chuỗi đích.
 - Kết quả trả về của hàm `strcat()` là chuỗi mới ghép nối từ 2 chuỗi chuỗi nguồn và chuỗi đích.
- `char* strncat(char* chuỗi_đích, char* chuỗi_nguồn, int n);`
 - Tương tự `strcat` nhưng chỉ giới hạn với `n` ký tự đầu tiên của chuỗi nguồn.
- `char* strlwr(char* chuỗi)`
 - Chuyển đổi các chữ in hoa trong chuỗi sang chữ thường.
- `char*strupr(char* chuỗi)`
 - Chuyển đổi các chữ thường trong chuỗi sang chữ in hoa.

18

5.3.3. Một số hàm xử lý chuỗi ký tự khác

- **void strset(char* s, char c)**
 - Khởi đầu giá trị tất cả các ký tự của chuỗi s bằng ký tự c
- **void strnset(char* s, char c, int n)**
 - Khởi đầu giá trị cho n ký tự đầu tiên của chuỗi s bằng ký tự c
- **int atoi(char* str)**
 - Chuyển một chuỗi ký tự là biểu diễn của một số nguyên thành số nguyên tương ứng.
 - Nếu chuyển đổi thành công, hàm atoi() trả về giá trị số nguyên chuyển đổi được, ngược lại trả về giá trị 0.

19

5.3.4. Con trỏ và chuỗi ký tự

- Như đã biết chuỗi ký tự là một dãy các ký tự đặt trong hai dấu nháy kép, ví dụ: "Đại Học Bách Khoa".
- Cũng giống như mảng, chuỗi ký tự là một hằng địa chỉ biểu thị địa chỉ đầu của mảng ký tự chứa chuỗi (Mảng này chứa các ký tự của chuỗi công thêm ký tự NULL).
- Vì vậy, nếu chúng ta khai báo **char *pc;** là một con trỏ kiểu char thì phép gán **pc="Đại Học Bách Khoa"** là hợp lệ. Sau khi thực hiện câu lệnh này, trong con trỏ pc có địa chỉ đầu của chuỗi "Đại Học Bách Khoa".

20

5.3.4. Con trỏ và chuỗi ký tự

- Giữa con trỏ kiểu char và mảng kiểu char có sự khác biệt. Xét ví dụ sau:

```
char ch[20];
char *pc;
pc = "Đại Học Bách Khoa"; /*Hợp lệ*/
ch = "Đại Học Bách Khoa"; /*Không hợp lệ*/
```

- câu lệnh gán đầu tiên là hợp lệ vì pc là con trỏ kiểu char, còn câu lệnh gán thứ hai không hợp lệ vì ch được khai báo là một mảng ký tự, do đó ch là một hằng, chúng ta không thể gán giá trị cho hằng được.

21

5.3.4. Con trỏ và chuỗi ký tự

→ Có thể thao tác trên tất cả các ký tự của chuỗi thông qua một con trỏ.

```
#include <stdio.h>
main()
{
    int i;
    char *str;
    str = "Lập trình C thật thú vị";
    for (i = 0; *str != '\0'; i++)
        str++;
    printf("so cac ky tu = %d\n", i);
}
```

22

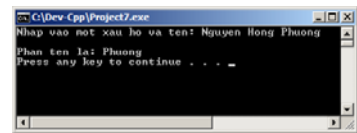
Bài tập

- Bài 1: Nhập một chuỗi ký tự từ bàn phím gồm các từ, ví dụ "Thu đo Ha Noi". Lập chương trình để bỏ bớt các dấu trống giữa các từ sao cho các từ chỉ cách nhau ít nhất một dấu trống.
- Bài 2: Viết chương trình nhập vào từ bàn phím họ và tên của một người, sau đó in phần tên ra màn hình. Ví dụ: "Tran Hung Dao" thì in ra "Dao".
- Bài 3: Nhập vào một câu, kết thúc bằng dấu chấm. In ra câu đó có bao nhiêu từ.

23

Bài chữa bài 2

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main(){
    char s[30], *p;
    clrscr();
    printf("Nhap vao mot xau ho va ten");
    gets(s);
    /* Dùng hàm strrchr duyệt ngược từ cuối chuỗi, tìm vị trí đầu tiên của
    một ký tự và trả về con trỏ trỏ đến ký tự đó. */
    p = strrchr(s, ','); // Tìm ký tự trống đầu tiên từ cuối chuỗi s
    p = p + 1;
    printf("\nPhan ten la: %s\n", p);
    getch();
}
```



24



Lời hay ý đẹp

Có những cuộc tìm kiếm đơn giản chỉ là tìm kiếm nhưng nhờ có nó ta hiểu rằng tình yêu là giữa một biển người vẫn tìm thấy nhau.

<http://my.opera.com/catviet/blog/2007/01/11/tat-ca-la-cuoc-song>

26

Chương 6 Hàm

Nguyễn Hồng Phương
Email: phuongnh-fit@mail.hut.edu.vn
Website: <http://is.hut.edu.vn/~phuongnh>
Bộ môn Hệ thống thông tin
Viện Công nghệ thông tin và Truyền thông
Đại học Bách Khoa Hà Nội

1

Nội dung chương này

- 6.1. Giới thiệu
- 6.2. Module chương trình trong C
- 6.3. Các hàm toán học
- 6.4. Các hàm người dùng định nghĩa
- 6.5. Nguyên mẫu hàm
- 6.6. Các tệp header
- 6.7. Truyền tham số cho hàm
- 6.8. Phạm vi biến
- 6.9. Đệ quy
- 6.10. Hàm main có đối dòng lệnh

2

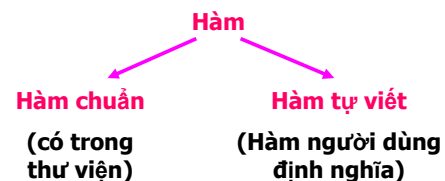
6.1. Giới thiệu

- Có những đoạn chương trình được thực hiện lặp đi lặp lại nhiều lần, tuy dữ liệu có khác nhưng bản chất các công việc lại giống nhau.
- Viết gộp những đoạn chương trình đó lại thành một chương trình con mà khi cần chỉ việc truyền dữ liệu cho nó?
- Tư tưởng đó cũng dẫn chúng ta tới việc chia một chương trình lớn thành nhiều phần nhỏ rồi giải quyết từng phần; sau đó sẽ ráp nối chúng lại là sẽ hoàn tất một chương trình lớn. Các chương trình nhỏ này trong C chính là các **hàm (function)**.
- Như vậy một chương trình sẽ là một tập hợp các định nghĩa hàm riêng biệt.

3

6.2. Module chương trình trong C

- Các module trong C được gọi là hàm.



- Hàm: viết một lần, dùng lại ở nhiều nơi.

4

6.2. Module chương trình trong C (tiếp)

- Các hàm được gọi để thực hiện bằng các **lời gọi hàm**.
- Các lời gọi hàm xác định tên của hàm và cung cấp các thông tin (hay còn gọi là các tham số) mà hàm được gọi theo đúng trình tự khi khai báo hàm đó.

5

6.3. Các hàm toán học trong C

- Khai báo tệp tiêu đề `#include<math.h>`
- Khi dùng chỉ việc viết lời gọi hàm
- Ví dụ: viết hàm tính và in ra căn bậc 2 của 900.0 là : `printf("%.2f", sqrt(900.0));`
 - `sqrt` là hàm khai căn bậc 2
 - số 900.0 là tham số của hàm `sqrt`
 - hàm `sqrt` nhận tham số kiểu `double` và trả về giá trị kiểu `double`
 - Câu lệnh sẽ in ra 30.00

6

6.3. Các hàm toán học trong C (tiếp)

- Các tham số của hàm có thể là các hằng, biến hay các biểu thức
- Ví dụ nếu $c1 = 13.0$, $d = 3.0$ thì câu lệnh `printf("%.2f", sqrt(c1 + d * 4.0));` sẽ tính và in ra căn bậc 2 của $13.0 + 3.0 * 4.0 = 25.0$ là 5.00

7

6.3. Các hàm toán học trong C (tiếp)

Hàm	Mô tả	Ví dụ
sqrt(x)	Căn bậc 2 của x	sqrt(9.00) bằng 3.0
exp(x)	hàm mũ e^x	exp(1.0) bằng 2.718282
log(x)	logarithm tự nhiên (cơ số e) của x	log(2.718282) bằng 1.0
log10(x)	logarithm thập phân (cơ số 10) của x	log10(1.0) bằng 0.0 log10(10.0) bằng 1.0
fabs(x)	giá trị tuyệt đối của x	nếu $x \geq 0$ thì fabs(x) bằng x nếu $x < 0$ thì fabs(x) bằng -x
ceil(x)	làm tròn thành số nguyên nhỏ nhất lớn hơn x	ceil(9.2) bằng 10.0 ceil(-9.8) bằng -9.0

8

6.3. Các hàm toán học trong C (tiếp)

floor(x)	làm tròn thành số nguyên lớn nhất nhỏ hơn x	floor(9.2) bằng 9.0 floor(-9.8) bằng -10.0
pow(x,y)	x mũ y (x^y)	pow(2,7) bằng 128
fmod(x,y)	phần dư của phép chia x cho y	fmod(13.657,2.333) bằng 1.992
sin(x)	sin của x (x theo radian)	sin(0.0) bằng 0.0
cos(x)	cos của x (x theo radian)	cos(0.0) bằng 1.0
tan(x)	tan của x (x theo radian)	tan(0.0) bằng 0.0

9

6.4. Hàm người dùng định nghĩa

- Xét bài toán: tính bình phương của 10 số tự nhiên từ 1 đến 10;
- Ta sẽ tổ chức chương trình thành 2 hàm `main()` và một hàm nữa mà ta đặt tên là `square()`; nhiệm vụ của hàm `square()` là tính bình phương của một số nguyên; còn hàm `main()` là sẽ gọi hàm `square()` để thực hiện nhiệm vụ được đề ra.

10

6.4. Hàm người dùng định nghĩa (tiếp)

```
#include<stdio.h>
#include<conio.h>
int square(int) /* Hàm nguyên mẫu */
void main()
{
    int x;
    for (x = 1; x <= 10; x++)
        printf( "%d  ", square(x));
    printf("\n");
    getch();
}
int square(int y) /* Định nghĩa hàm square */
{
    return  y * y;
}
Kết quả
1 4 9 16 25 36 49 64 81 100
```

11

6.4. Hàm người dùng định nghĩa (tiếp)

- Khuôn dạng một hàm:


```
kiểu  tên-hàm (danh-sách-tham-số)
{
    Các khai báo
    ...
    Các câu lệnh
    [return [biểu_thức];]
}
```

12

6.4. Hàm người dùng định nghĩa (tiếp)

- Giải thích:
- Ở đây *kiểu* là kiểu dữ liệu của kết quả của hàm, *tên-hàm* là một cái tên do người lập trình tự đặt;
- *Danh-sách-tham-số* gồm danh sách các tham số hàm nhận được, khi nó được gọi và được ngăn cách nhau bởi dấu phẩy.
- Các khai báo và các câu lệnh trong ngoặc là *phần thân của hàm*. Khai báo và cài đặt một hàm lại không được nằm trong một hàm khác trong bất kỳ tình huống nào.

13

6.4. Hàm người dùng định nghĩa (tiếp)

- Hàm có thể có giá trị trả về hoặc không có giá trị trả về.
 - Nếu có giá trị trả về, trong thân hàm có ít nhất một lệnh **return**.
 - Nếu không có giá trị trả về cần khai báo cho hàm đó có kiểu trả về là **void**.

14

6.4. Hàm người dùng định nghĩa (tiếp)

- Ví dụ:

<pre>int giai_thua(int a)</pre>	Dòng đầu hàm
<pre>{ int ket_qua; int i;</pre>	Các khai báo
<pre> ket_qua = 1; for(i = 1; i <= a; i++) ket_qua = ket_qua * i; if(a < 0) ket_qua = -1; if(a == 0) ket_qua = 1; return ket_qua; }</pre>	Các câu lệnh

15

6.4. Hàm người dùng định nghĩa

Ví dụ thứ ba

- Xây dựng hàm maximum trả về số nguyên lớn nhất trong ba số nguyên.

```
#include<stdio.h>  
#include<conio.h>  
int maximum(int, int, int) /* Hàm nguyên mẫu */  
void main() /* Hàm main */  
{  
    int a, b, c, largest;  
    printf(" Vào 3 số nguyên : ");  
    scanf(" %d%d%d ", &a, &b, &c);  
    largest = maximum(a,b,c);  
    printf(" Số lớn nhất là : %d\n", largest);  
    getch();  
}
```

16

6.4. Hàm người dùng định nghĩa

Ví dụ thứ ba (tiếp)

```
/* Định nghĩa hàm maximum */  
int maximum(int x, int y, int z)  
{  
    int max = x;  
    if (y > max)  
        max = y;  
    if (z > max)  
        max = z;  
    return max;  
}
```

- Kết quả
Vào 3 số nguyên : 22 85 17
Số lớn nhất là : 85
Vào 3 số nguyên : 85 22 17
Số lớn nhất là : 85

17

6.5. Nguyên mẫu hàm (Prototype)

- Một trong những đặc trưng quan trọng của ANSI C là hàm nguyên mẫu (khai báo hàm). Hàm nguyên mẫu thông báo cho trình biên dịch biết kiểu dữ liệu hàm trả lại, số lượng, kiểu và trình tự của các tham số truyền cho hàm. Trình biên dịch dùng hàm nguyên mẫu để kiểm tra các lời gọi hàm.
- Quan sát lại ví dụ đầu tiên:

18

6.5. Nguyên mẫu hàm (tiếp)

```
#include<stdio.h>
#include<conio.h>
int square(int) /* Hàm nguyên mẫu */
void main()
{
    int x;
    for (x = 1; x <= 10; x++)
        printf( "%d  ", square(x));
    printf("\n");
    getch();
}
int square(int y) /* Định nghĩa hàm square */
{
    return  y * y;
}
```

19

6.6. Các tệp header

- Mỗi một thư viện chuẩn tương ứng có một tệp header chứa các khai báo của tất cả các hàm trong thư viện này cùng với các định nghĩa các kiểu dữ liệu khác nhau, các hằng dùng trong các hàm đó
- Lập trình viên có thể tạo các tệp header riêng của mình. Các tệp header này cũng thường có kiểu .h và có thể dùng chỉ thị tiền xử lý #include để thêm vào chương trình.

20

6.7. Truyền tham số cho hàm

- Có hai cách để truyền các tham số cho hàm:
 - Khi các tham số được truyền theo trị, một bản sao giá trị của tham số được tạo ra và truyền cho hàm. Vì vậy mọi sự thay đổi trong hàm trên bản sao sẽ không ảnh hưởng đến giá trị ban đầu của biến nằm trong hàm gọi.
 - Khi một tham số được truyền theo con trỏ, hàm gọi sẽ truyền trực tiếp tham số đó cho hàm bị gọi thay để đổi giá trị nguyên thủy của biến truyền vào tham số.
- Trong C, tất cả các tham số được truyền đều là truyền theo trị.
- Ta có thể mô phỏng cách truyền tham số theo tham chiếu bằng cách truyền địa chỉ của các biến vào tham số.

21

6.7. Truyền tham số cho hàm - ví dụ

- Viết hàm hoán đổi hai số nguyên
- Hãy quan sát hai cách sau

22

6.7. Truyền tham số cho hàm - ví dụ

```
▪ Cách 1:
#include<stdio.h>
void swap(int, int); /* Hàm nguyên mẫu */
void main(){
    int x, y;
    x = 10; y = 20;
    printf("Ban đầu x = %d, y = %d", x, y);
    swap(x, y);
    printf("Sau đó x = %d, y = %d", x, y);
}
void swap( int x, int y){ /* Định nghĩa hàm swap */
    int temp;
    temp = x;
    x = y;
    y = temp;
}
```

Liệu có thực sự hoán đổi?

23

6.7. Truyền tham số cho hàm - ví dụ

```
▪ Cách 2:
include <stdio.h>
void swap( int* , int* ); /* Hàm nguyên mẫu */
void main(){
    int x, y;
    x = 10; y = 20;
    printf("Ban đầu x = %d, y = %d", x, y);
    swap(&x, &y);
    printf("Sau đó x = %d, y = %d", x, y);
}
void swap( int* x, int* y){ /* Định nghĩa hàm swap */
    int temp;
    temp = *x;
    *x = *y;
    *y = temp;
}
```

Truyền địa chỉ của biến vào vị trí các tham số

24

6.7. Truyền tham số cho hàm-nhận xét

- Nếu đối của hàm là con trỏ kiểu int, float, double,... thì tham số thực sự tương ứng trong lời gọi hàm phải là địa chỉ của biến hoặc địa chỉ của phần tử mảng có kiểu int, float, double... Khi đó, địa chỉ của biến được truyền cho đối con trỏ tương ứng. Do đã biết được địa chỉ của biến nên có thể gán cho biến các giá trị mới.
- Các đối của hàm có hai loại: các đối chứa dữ liệu đã biết (gọi là đối vào), các đối chứa kết quả mới nhận được (đối ra). Các đối ra phải khai là các đối con trỏ.

25

6.7. Truyền tham số cho hàm

- Hàm có đối là mảng một chiều: Hai cách khai báo hàm:
 - void Tên_hàm(float *pa,...): pa là một con trỏ.
 - void Tên_hàm(float pa[],...): pa coi như một mảng hình thức.
 - Nếu tham số thực là tên mảng a một chiều kiểu float thì ta gọi Tên_hàm(a,...).
 - Khi hàm bắt đầu làm việc thì giá trị của a (địa chỉ phần tử a[0]) được truyền cho pa. Do đó, muốn truy nhập tới phần tử a[i] trong thân hàm ta dùng *(pa+i) hoặc pa[i].
 - Với các kiểu char, int, double, cấu trúc,...ta cũng làm tương tự như trên.

26

6.7. Truyền tham số cho hàm

- Hàm có đối là mảng 2 chiều: Giả sử a là mảng 2 chiều float a[20][30]. Ta có thể khai báo theo hai cách sau:
 - Cách 1:
 - void Tên_hàm(float (*pa)[50], int m, int n)
 - void Tên_hàm(float pa[][50], int m, int n)
 - pa là con trỏ kiểu float[50] (chứa địa chỉ đầu của vùng nhớ dành cho 50 số thực, tức là vùng nhớ 200 byte), m là số hàng và n là số cột
 - Trong thân hàm, để truy nhập tới phần tử hàng i cột j, ta dùng kí hiệu pa[i][j].
 - Theo cách này, ta chỉ làm việc với mảng 2 chiều có tối đa 50 cột. Lời gọi hàm Tên_hàm(a, m, n)

27

6.7. Truyền tham số cho hàm

- Hàm có đối là mảng 2 chiều
 - Cách 2:
 - float *pa; biểu thị địa chỉ đầu của mảng a
 - int N; biểu thị số cột của mảng a
 - Khai báo: Tên_hàm(float *pa, int N,...)
 - Lời gọi: Tên_hàm(a, 30,...)
 - Trong thân hàm, để truy nhập tới phần tử a[i][j] ta dùng công thức *(pa + i*N + j)
 - Theo cách này, mảng 2 chiều quy về mảng 1 chiều, hàm có thể dùng cho bất kỳ mảng 2 chiều nào.

28

6.8. Phạm vi biến

- Biến địa phương (Local Variable):
 - Là các biến được khai báo trong lệnh khối hoặc trong thân chương trình con.
- Biến toàn cục (Global Variable):
 - Là biến được khai báo trong chương trình chính.
 - Vị trí khai báo của biến toàn cục là sau phần khai báo tệp tiêu đề và khai báo hàm nguyên mẫu

29

6.8. Phạm vi biến - ví dụ 1:

```
#include <stdio.h>
void main() {
    {
        int a = 1;
        printf("\n a = %d", a);
        {
            int a = 2;
            printf("\n a = %d", a);
        }
        printf("\n a = %d", a);
    }
    {
        int a = 3;
        printf("\n a = %d", a);
    }
}
```



a1	=	1
a2	=	2
a3	=	1
a4	=	3

30

6.8. Phạm vi biến - ví dụ 2:

```
#include <stdio.h>
#include <conio.h>
int a, b, c;
int tinh()
{
    printf("\n Gia tri cac bien tong the a, b, c: ");
    printf(" a = %-5d b = %-5d c = %-5d", a, b, c);
    return a*b*c;
}

void main()
{
    clrscr();
    printf("\n Nhap gia tri cho 3 so nguyen a, b, c: ");
    scanf("%d %d %d", &a, &b, &c);
    printf("\n Tich cua 3 so la %d", tinh());
    getch();
}
```

```
Nhap 3 so nguyen a, b, c: 2 3 4
Gia tri cac bien tong the a, b, c:
a = 2    b = 3    c = 4
Tich cua 3 so la 24
```

31

Biến register

- Thanh ghi có tốc độ truy nhập nhanh hơn so với các loại bộ nhớ khác (RAM, bộ nhớ ngoài).
- Nếu một biến thường xuyên sử dụng được lưu vào trong thanh ghi thì tốc độ thực hiện của chương trình sẽ được tăng lên.
- Để làm điều này ta đặt từ khóa **register** trước khai báo của biến đó.
 - Ví dụ: `register int a;`
- Số lượng và kích thước các thanh ghi có hạn → Số lượng biến khai báo **register** sẽ không nhiều và thường chỉ áp dụng với những biến có kích thước nhỏ như kiểu **char**, **int**.

32

Biến static

- Một biến cục bộ khi ra khỏi phạm vi của biến đó thì bộ nhớ dành để lưu trữ biến đó sẽ được giải phóng.
- Nếu cần lưu giá trị của các biến cục bộ này, cần khai báo biến với từ khóa **static**.
 - Ví dụ: `static int a;`
- Biến **static** là biến tĩnh, nghĩa là nó sẽ được cấp phát một vùng nhớ thường xuyên từ lúc khai báo và chỉ giải phóng khi chương trình chính kết thúc.

33

Ví dụ

```
# include <stdio.h>
# include <conio.h>
void fct()
{
    static int count = 1;
    printf("\n Day la lan goi ham fct lan thu %2d",
        count++);
}

void main()
{
    int i;
    for(i = 0; i < 10; i++)
        fct();
    getch();
}
```

```
Day la lan goi ham fct lan thu 1
Day la lan goi ham fct lan thu 2
Day la lan goi ham fct lan thu 3
Day la lan goi ham fct lan thu 4
Day la lan goi ham fct lan thu 5
Day la lan goi ham fct lan thu 6
Day la lan goi ham fct lan thu 7
Day la lan goi ham fct lan thu 8
Day la lan goi ham fct lan thu 9
Day la lan goi ham fct lan thu 10
```

34

6.9. Đệ quy

- Trong C cho phép trong thân một hàm có thể gọi ngay đến chính nó, cơ chế này gọi là đệ quy.
- Có thể định nghĩa **hàm đệ quy** là hàm sẽ gọi đến chính nó trực tiếp hay gián tiếp thông qua các hàm khác.
- Cách tiến hành giải một bài toán đệ quy nhìn chung có những điểm chung sau.
 - Có trường hợp cơ sở
 - Nó gọi đến chính dạng của nó nhưng có sự chuyển biến.
 - Sự chuyển biến ấy đến một lúc nào đó phải kết thúc.

35

6.9. Đệ quy - ví dụ 1

- Ta xem xét ví dụ tính n giai thừa (**n!**).
 - Theo định nghĩa n! bằng tích của tất cả các số tự nhiên từ 1 đến n: $n! = n * (n-1) * ... * 2 * 1$
 - Ví dụ: $5! = 5 * 4 * 3 * 2 * 1 = 120$, 0! được định nghĩa bằng 1.
 - Để tính n! có thể dùng vòng lặp như sau:

```
fact = 1;
for (i = n; i >= 1; i--)
    fact *= i;
```
 - Tuy nhiên cũng có thể định nghĩa đệ quy hàm giai thừa như sau:

```
nếu n>0 thì n! = n * (n-1)!
nếu n=0 thì n! = 0! = 1
```

36

6.9. Đệ quy - ví dụ 1

```
/* Tính giai thừa theo phương pháp đệ quy */
#include <stdio.h>
long fact( long ); /* Hàm nguyên mẫu */
main()
{
    int i;
    for ( i=0; i<=10; i=i+2)
        printf("%2d! = %ld\n", i, fact(i));
}
/* Định nghĩa đệ quy hàm factorial */
long fact( long n)
{
    if ( n == 0 ) return 1;
    else return (n*fact(n-1));
}
```

37

6.9. Đệ quy - ví dụ 2

- Một ví dụ thứ hai dùng đệ quy là tính dãy số Fibonacci
 - Dãy số Fibonacci gồm những số 0, 1, 1, 2, 3, 5, 8, 13, 21 ...
 - Bắt đầu từ hai số 0 và 1, tiếp sau đó các số Fibonacci sau bằng tổng của 2 số Fibonacci trước nó.
 - Dãy Fibonacci có thể định nghĩa đệ quy như sau:
 $fibonacci(0) = 0; fibonacci(1) = 1;$
 $fibonacci(n) = fibonacci(n-1) + fibonacci(n-2) \quad n > 1$

38

6.9. Đệ quy - ví dụ 2

```
/* Tính dãy số Fibonacci phương pháp đệ quy */
#include <stdio.h>
long fibo( long ); /* Hàm nguyên mẫu */
void main(){
    long result, n;
    printf("Hãy nhập vào một số nguyên : ");
    scanf("%ld", &n);
    result = fibo(n);
    printf("Fibonacci thứ %ld là : %ld\n", number, result);
}
/* Định nghĩa đệ quy hàm fibonacci */
long fibo( long n){
    if ( n == 0 || n == 1 ) return n;
    else return fibo(n-1) + fibo(n-2);
}
```

39

So sánh đệ quy và lặp

- So sánh:
 - Lặp trình đệ quy sử dụng cấu trúc lựa chọn, còn lặp sử dụng cấu trúc lặp.
 - Cả hai phương pháp đều phải kiểm tra khi nào thì kết thúc. Phương pháp lặp kết thúc khi điều kiện để tiếp tục vòng lặp sai, còn phương pháp đệ quy kết thúc khi đến trường hợp cơ sở.
 - Tuy nhiên đệ quy tồi hơn, nó liên tục đưa ra các lời gọi hàm làm tốn thời gian xử lý của bộ vi xử lý và không gian nhớ.
- "*Công nghệ phần mềm*" là quan trọng nhưng *hiệu năng của chương trình* cũng quan trọng và thật không may hai mục tiêu đó lại thường loại trừ lẫn nhau.
- Vậy khi module hoá chương trình phải khôn ngoan cân đối giữa hai mục tiêu là *Công nghệ phần mềm* và *Hiệu năng*

40

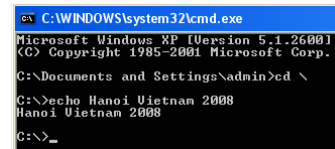
6.10. Hàm main có đối dòng lệnh

- Giả sử ta có chương trình echo.exe và chạy nó theo dòng lệnh:
C:\>echo Hanoi 2008
- Khi đó, "Hanoi" và "2008" là các tham số dòng lệnh.
- Hàm main có 2 tham số:
 - argc (argument count): số lượng đối số dòng lệnh
 - argv (argument vector): con trỏ trỏ tới mảng chuỗi ký tự, mảng này chứa các đối số dòng lệnh, mỗi đối là một chuỗi
- Khi gõ vào từ dòng lệnh, các tham số phải cách nhau bởi dấu cách. Chương trình xử lý các tham số dòng lệnh như các dữ liệu vào.

41

Mã nguồn chương trình echo.c

```
#include <stdio.h>
int main(int argc, char *argv[]){
    int i;
    for(i=1; i<argc; i++)
        printf("%s%s", argv[i],(i<argc-1)? " ": "");
    printf("\n");
    return 0;
}
```



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\Documents and Settings\admin>cd \
C:\>echo Hanoi Vietnam 2008
Hanoi Vietnam 2008
C:\>_
```

42

Bài tập

- **Bài 1:** Viết hàm tìm ước số chung lớn nhất `uscln(int x, int y)` và bội số chung nhỏ nhất `bscnn(int x, int y)` của hai số.
- **Bài 2:** Lập hàm giải phương trình bậc 2 `ptb2(float a, float b, float c, float *x1, float *x2)`, trong đó `a,b,c` là các hệ số của phương trình. Hàm nhận giá trị 0 nếu phương trình vô nghiệm, giá trị 1 nếu phương trình có nghiệm kép (chứa trong `*x1`), giá trị 2 nếu phương trình có 2 nghiệm (chứa trong `*x1` và `*x2`), giá trị 3 nếu phương trình có vô số nghiệm.

43

Bài tập

- **Bài 3:** Lập hàm tính giai thừa và dùng hàm này tính tổng:
 $1! + 2! + \dots + N!$ (`N` nhập từ bàn phím và ≤ 10)
- **Bài 4:** Đa thức cấp `n` được tính theo công thức:
 $f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x^1 + a_0 x^0$
 - Lập hàm đa thức `f(float a[], int n, float x)` để tính giá trị của đa thức.

44



45

Lời hay ý đẹp

Có những ước mơ sẽ vẫn chỉ là ước mơ dù cho ta có nỗ lực đến đâu nhưng nhờ có nó ta mạnh mẽ hơn, yêu cuộc sống hơn và biết cố gắng từng ngày.
Có những lời hứa cũng vẫn chỉ là lời hứa dù ta có mãi chờ đợi bởi người hứa đã không còn nhớ, nhưng nhờ có nó ta biết hi vọng và mong chờ.

<http://my.opera.com/catviet/blog/2007/01/11/tat-ca-la-cuoc-song>

46

Chương 7 Cấu trúc (struct)

Nguyễn Hồng Phương
Email: phuongnh-fit@mail.hut.edu.vn
Website: <http://is.hut.edu.vn/~phuongnh>
Bộ môn Hệ thống thông tin
Viện Công nghệ thông tin và Truyền thông
Đại học Bách Khoa Hà Nội

1

Nội dung

- 7.1. Khái niệm cấu trúc
- 7.2. Khai báo và sử dụng cấu trúc
- 7.3. Xử lý dữ liệu cấu trúc
- 7.4. Mảng cấu trúc
- 7.5. Bài tập

2

7.1. Khái niệm cấu trúc

- Kiểu dữ liệu cấu trúc (**struct**) là kiểu dữ liệu phức hợp bao gồm nhiều thành phần, mỗi thành phần có thể thuộc những kiểu dữ liệu khác nhau.
- Các thành phần dữ liệu trong cấu trúc được gọi là các trường dữ liệu (*field*).

3

7.1. Khái niệm cấu trúc (tiếp)

- Ví dụ: Cấu trúc mô tả sinh viên:
 - Cần lưu giữ các thông tin liên quan đến sinh viên như họ tên, tuổi, kết quả học tập...
 - Mỗi thông tin thành phần lại có kiểu dữ liệu khác nhau như họ tên có kiểu dữ liệu là chuỗi ký tự, tuổi có kiểu dữ liệu là số nguyên, kết quả học tập có kiểu dữ liệu là số thực.

4

Nội dung

- 7.1. Khái niệm cấu trúc
- 7.2. Khai báo và sử dụng cấu trúc
- 7.3. Xử lý dữ liệu cấu trúc
- 7.4. Mảng cấu trúc
- 7.5. Bài tập

5

7.2. Khai báo kiểu dữ liệu cấu trúc

- Cú pháp:

```
struct ten_cau_truc
{
    <khai_bao_cac_truong_du_lieu>;
};
```
- Ví dụ:

```
struct sinh_vien
{
    char ma_so_sinh_vien[10];
    char ho_va_ten[30];
    float diem_TinDC;
};
```

6

7.2.2. Khai báo biến cấu trúc

- Cú pháp:

```
struct ten_cau_truc ten_bien_cau_truc;
```

- Ví dụ:

```
struct sinh_vien a, b, c;
```

7

7.2.3. Khai báo kết hợp

- Có thể kết hợp vừa khai báo kiểu dữ liệu cấu trúc vừa khai báo biến cấu trúc:

```
struct [ten_cau_truc]
{
    //khai_bao_cac_truong;
} ten_bien_cau_truc;
```

- Ví dụ:

```
struct thi_sinh
{
    char SBD[10]; // so bao danh
    char ho_va_ten[30];
    struct diem_thi ket_qua;
} thi_sinh_1, thi_sinh_2;
```

8

7.2.4. Định nghĩa kiểu dữ liệu cấu trúc với typedef

- Đặt tên mới cho kiểu dữ liệu cấu trúc cú pháp:

```
typedef struct ten_cũ ten_moi;
```

- Hoặc ta có thể đặt lại tên cho cấu trúc ngay khi khai báo bằng cú pháp

```
typedef struct [ten_cũ]
{
    //khai_bao_cac_truong;
} danh_sach_cac_ten_moi;
```

9

7.2.4. Định nghĩa kiểu dữ liệu cấu trúc với typedef (tiếp)

- Lưu ý: Được phép đặt tên_mới trùng với tên_cũ.

- Ví dụ:

```
struct point_3D
{
    float x, y, z;
} P;

struct point_3D M;
typedef struct point_3D p_3D;
p_3D N;
```

10

Nội dung

- 7.1. Khái niệm cấu trúc
- 7.2. Khai báo và sử dụng cấu trúc
- 7.3. Xử lý dữ liệu cấu trúc
- 7.4. Mảng cấu trúc
- 7.5. Bài tập

11

7.3. Xử lý dữ liệu cấu trúc

- Truy nhập các trường dữ liệu của cấu trúc
- Phép gán giữa các biến cấu trúc
- Con trỏ cấu trúc

12

7.3.1. Truy nhập các trường dữ liệu của cấu trúc

- Cú pháp:
`ten_bien_cau_truc.ten_truong`
- Ví dụ:

```
struct point_2D
{
    char ten_diem;
    float x, y;
} p;
```

Truy cập tên của 1 điểm: `p.ten_diem`

13

7.3.2. Phép gán giữa các biến cấu trúc

- Giả sử ta có 2 biến cấu trúc là **a** và **b** có **cùng kiểu dữ liệu** là một cấu trúc nào đó.
- Phép gán cấu trúc có cú pháp tương tự như phép gán thông thường:
`bien_cau_truc_1 = bien_cau_truc_2;`

14

7.3.3. Con trỏ cấu trúc

- Cú pháp:
`struct ten_cau_truc *ten_bien_con_tro_ctruc;`
- Có 2 cách truy nhập vào trường dữ liệu của cấu trúc từ biến con trỏ cấu trúc là
`(*ten_bien_con_tro_ctruc).ten_truong_du_lieu`
Hoặc:
`ten_bien_con_tro_ctruc->ten_truong_du_lieu`

15

Nội dung

- 7.1. Khái niệm cấu trúc
- 7.2. Khai báo và sử dụng cấu trúc
- 7.3. Xử lý dữ liệu cấu trúc
- 7.4. Mảng cấu trúc
- 7.5. Bài tập

16

7.4. Mảng cấu trúc

- Cú pháp:
`struct ten_cau_truc ten_mang_ctruc[so_ptu];`
- Ví dụ:

```
struct sinh_vien
{
    char ho_ten[20];
    float diem_thi;
};

struct sinh_vien lop_Tin5[80];
```

17

Ví dụ

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
void main()
{
    struct sinh_vien
    {
        char ma_sv[10];
        char ho_ten[20];
        float diem_thi;
    };
    struct sinh_vien sv[3];
    int i;
    clrscr();
```

18

Ví dụ (tiếp)

```
for(i=0;i<3;i++)
{
    char str[20];
    float diem;
    printf("\nNhap thong tin cho sinh vien %d",i+1);
    printf("\nMa so sinh vien:");
    fflush(stdin); gets(str);
    strcpy(sv[i].ma_sv,str);
    printf("\n Ho va ten: ");
    fflush(stdin); gets(str);
    strcpy(sv[i].ho_ten,str);
    printf("\n Diem thi: ");
    scanf("%f",&diem);
    sv[i].diem_thi = diem;
}
```

19

Ví dụ (tiếp)

```
printf("\n Thong tin ve cac sinh vien");
for(i=0;i<3;i++)
{
    printf("\n Sinh vien thu %d ",i+1);
    printf("%-10s %-20s %-3.1f",
    sv[i].ma_sv, sv[i].ho_ten,
    sv[i].diem_thi);
}
getch();
}
```

20

Ví dụ - Kết quả

```
Nhap thong tin cho sinh vien thu 1
Ma so sinh vien: SV0032
Ho va ten: Nguyen Thanh Binh
Diem thi: 8.5
Nhap thong tin cho sinh vien thu 2
Ma so sinh vien: SV0002
Ho va ten: Pham Hong Phuc
Diem thi: 9
Nhap thong tin cho sinh vien thu 3
Ma so sinh vien: SV0046
Ho va ten: Le Minh Hoa
Diem thi: 10
Thong tin ve cac sinh vien
Sinh vien thu 1: SV0032 Nguyen Thanh Binh 8.5
Sinh vien thu 2: SV0002 Pham Hong Phuc 9.0
Sinh vien thu 3: SV0046 Le Minh Hoa 10.0
```

21

Bài tập

- Viết 1 chương trình thực hiện những công việc sau:
 - Yêu cầu người dùng nhập vào một số nguyên dương n với 5 ≤ n ≤ 20 (có kiểm tra tính hợp lệ của giá trị được nhập vào, nếu giá trị n nhập vào không thỏa mãn điều kiện thì yêu cầu nhập lại)
 - Yêu cầu người dùng nhập vào thông tin của n sinh viên gồm những mục sau
 - Họ và tên: có kiểu dữ liệu là xâu kí tự gồm không quá 30 kí tự
 - Lớp: xâu kí tự có độ dài không quá 5 kí tự
 - Điểm thi Tin đại cương: là một số nguyên có giá trị từ 0 đến 10
 - Đưa ra màn hình danh sách các sinh viên cùng thông tin của họ mà người dùng vừa nhập vào.

22

Bài tập

- Yêu cầu người dùng nhập vào từ bàn phím một số thực. Đưa ra màn hình danh sách các sinh viên có điểm thi Tin đại cương nhỏ hơn giá trị số thực vừa nhập vào.
- Đưa ra màn hình danh sách sinh viên được sắp xếp theo chiều giảm dần của điểm thi Tin đại cương.
- Đưa ra màn hình danh sách sinh viên với họ và tên được sắp xếp theo chiều của bảng chữ cái.

23

Hỏi - đáp



24

Lời hay ý đẹp



Chương 8 Tập (FILE)

Nguyễn Hồng Phương
Email: phuongnh-fit@mail.hut.edu.vn
Website: <http://is.hut.edu.vn/~phuongnh>
Bộ môn Các hệ thống thông tin
Viện Công nghệ thông tin và Truyền thông
Đại học Bách Khoa Hà Nội

1

Nội dung

- 8.1. Giới thiệu
- 8.2. Kiểu xuất nhập nhị phân và văn bản
- 8.3. Các hàm thao tác cấp 2
- 8.4. Đóng/mở tệp, xóa vùng đệm, kiểm tra lỗi
- 8.5. Nhập xuất ký tự
- 8.6. Các hàm nhập xuất theo kiểu văn bản
- 8.7. Tệp văn bản và các thiết bị chuẩn
- 8.8. Các hàm nhập xuất theo kiểu nhị phân
- 8.9. Nhập xuất ngẫu nhiên, di chuyển con trỏ chỉ vị

2

8.1. Giới thiệu

- Một tệp tin đơn giản chỉ là một dãy các byte (mỗi byte có giá trị từ 0 đến 255) ghi trên đĩa. Số byte của dãy chính là độ dài của tệp.
- Chương này trình bày các thao tác trên tệp như tạo một tệp mới, ghi dữ liệu từ bộ nhớ lên tệp, đọc dữ liệu từ tệp vào bộ nhớ,...
- Trong C, các thao tác trên tệp được thực hiện nhờ các hàm thư viện. Các hàm này được chia thành 2 nhóm: cấp 1 và cấp 2.
- Mỗi hàm (cấp 1 hay cấp 2) đều có thể truy xuất theo cả hai kiểu nhị phân và văn bản.

3

8.1. Giới thiệu

- Các hàm cấp 1:
 - thực hiện việc đọc/ghi như DOS
 - Không có dịch vụ xuất nhập riêng cho từng kiểu dữ liệu mà chỉ có dịch vụ đọc/ghi một dãy các byte. Ví dụ: để ghi 1 số thực lên đĩa, ta dùng dịch vụ ghi 4 byte; để ghi 10 số nguyên lên đĩa, ta dùng dịch vụ ghi 20 byte.
 - Mỗi tệp có một số hiệu (handle). Các hàm cấp 1 làm việc với tệp thông qua số hiệu tệp này.

4

8.1. Giới thiệu

- Các hàm cấp 2:
 - được xây dựng từ các hàm cấp 1 nên dễ sử dụng và có nhiều khả năng hơn.
 - có dịch vụ truy xuất cho từng kiểu dữ liệu. Ví dụ: hàm xuất nhập ký tự, chuỗi, số nguyên, số thực, cấu trúc,...
 - C tự động cung cấp một vùng đệm. Mỗi lần đọc/ghi thường tiến hành trên vùng đệm chứ không hẳn trên tệp. Khi ghi dữ liệu thì dữ liệu được đưa vào vùng đệm, khi nào vùng đệm đầy thì dữ liệu ở vùng đệm mới được đẩy lên đĩa. Khi đọc, thông tin được lấy ra từ vùng đệm, khi nào vùng đệm trống thì máy mới lấy dữ liệu từ đĩa đưa vào vùng đệm giảm số lần nhập xuất trên đĩa, nâng cao tốc độ làm việc.
 - làm việc với tệp thông qua một **biến con trỏ tệp**.

5

8.2. Kiểu nhập xuất nhị phân và văn bản

- 8.2.1. Kiểu nhị phân
 - Bảo toàn dữ liệu: trong quá trình xuất nhập, dữ liệu không bị biến đổi
 - Mã kết thúc tệp: trong khi đọc, nếu gặp cuối tệp thì ta nhận được mã kết thúc tệp EOF (giá trị là -1) và hàm feof cho giá trị khác 0. Tại sao lại chọn giá trị -1? Lý do rất đơn giản: chưa gặp cuối tệp thì sẽ đọc được một byte có giá trị từ 0 đến 255. Giá trị -1 sẽ không trùng với bất kỳ byte nào.

6

8.2. Kiểu nhập xuất nhị phân và văn bản

- 8.2.2. Kiểu văn bản:
 - chỉ khác kiểu nhị phân khi xử lý ký tự chuyển dòng (mã 10) và ký tự mã 26
 - Mã chuyển dòng:
 - khi ghi, một ký tự LF (mã 10) được chuyển thành 2 ký tự CR (mã 13) và LF.
 - khi đọc, 2 ký tự liên tiếp CR và LF trên tệp chỉ cho ta một ký tự LF.
 - Ví dụ: xét hàm `fputc(10,fp)`; nếu tệp `fp` mở theo kiểu nhị phân thì hàm sẽ ghi lên tệp một ký tự mã 10; nhưng nếu `fp` mở theo kiểu văn bản thì hàm ghi lên tệp hai mã là 13 và 10.
 - Mã kết thúc tệp: khi đọc, nếu gặp ký tự có mã 26 hoặc cuối tệp thì ta nhận được mã kết thúc tệp EOF (số -1) và hàm `feof(fp)` cho giá trị khác 0.

7

8.2.3. Ví dụ minh họa 1

- Chương trình sau tạo 2 tệp có tên là `vb` và `np`. Trong chương trình dùng các hàm:
 - `fopen` để mở tệp
 - `fputc` để ghi một ký tự lên tệp
 - `fclose` để đóng tệp

8

8.2.3. Ví dụ minh họa 1 (tiếp)

```
#include<stdio.h>
void main(){
    FILE *fvb, *fnp; //Khai báo 2 biến con trỏ tệp
    fvb = fopen("vb","wt"); //Mở tệp vb để ghi theo kiểu văn bản
    fnp = fopen("np","wb"); //Mở tệp np để ghi theo kiểu nhị phân
    // Ghi các ký tự lên tệp vvb
    fputc('A',fvb); fputc(26,fvb); fputc(10,fvb); fputc('B',fvb);
    // Ghi các ký tự lên tệp fnp
    fputc('A',fnp); fputc(26,fnp); fputc(10,fnp); fputc('B',fnp);
    fclose(fvb);
    fclose(fnp);
}
```

9

8.2.3. Ví dụ minh họa 1 (tiếp)

- Kết quả:
 - Tệp `vb` có các ký tự ứng với các mã: 65 26 13 10 66
 - Tệp `np` có các ký tự ứng với các mã: 65 26 10 66
- Chú ý:
 - nếu dùng kiểu văn bản để đọc tệp `vb` hay tệp `np` thì ta chỉ nhận được một ký tự đầu (mã 65) vì khi gặp ký tự thứ hai (mã 26) thì ta nhận được mã kết thúc tệp.
 - muốn đọc tất cả các ký tự của tệp, ta cần dùng hàm `fgetc` theo kiểu nhị phân.

10

8.2.4. Ví dụ minh họa 2

- Xét chương trình sau:

```
#include<stdio.h>
void main(){
    FILE *f; // Khai báo biến con trỏ tệp
    f = fopen("sl","wt"); //Mở tệp sl để ghi theo kiểu văn bản
    // Ghi 3 dòng lên tệp f
    fprintf(f,"%2d\n%2d\n%2d",56,7,8);
    fclose(f); // Đóng tệp
}
```

11

8.2.4. Ví dụ minh họa 2 (tiếp)

- Hàm `fprintf()` đưa kết quả ra tệp theo cách như hàm `printf()` đưa ra màn hình. Vì tệp `f` mở theo kiểu văn bản nên ký tự xuống dòng `'\n'` được ghi thành 2 mã 13 và 10. Kết quả là 10 ký tự ứng với các mã sau được ghi lên tệp:
53 54 13 10 32 55 13 10 32 56
- trong đó: 53 là mã của chữ số 5, 54 là mã của chữ số 6, 13 là CR, 10 là LF, 32 là mã của khoảng trống, 55 là mã của chữ số 7, 56 là mã của chữ số 8.

12

8.2.4. Ví dụ minh họa 2 (tiếp)

- Nếu dùng trình soạn thảo văn bản (ví dụ notepad) để mở tệp trên thì ta sẽ nhìn thấy các số 56, 7, 8 trên 3 dòng khác nhau.

- Nếu mở tệp sl theo kiểu nhị phân bằng cách dùng câu lệnh:

```
f = fopen("sl", "wb");
```

thì tệp sl sẽ gồm 8 mã sau:

```
53 54 10 32 55 10 32 56
```

13

8.3. Các hàm cấp 2

- Các hàm dùng chung cho cả 2 kiểu:
 - fopen dùng để mở tệp.
 - fclose dùng để đóng tệp.
 - fcloseall dùng để đóng tất cả các tệp đang mở.
 - fflush dùng để làm sạch vùng đệm của tệp.
 - fflushall dùng để làm sạch vùng đệm của các tệp đang mở.
 - feof cho biết đã gặp cuối tệp hay chưa.
 - rewind dùng để chuyển con trỏ chỉ vị về đầu tệp.
 - fseek dùng để di chuyển con trỏ chỉ vị đến bất kỳ vị trí trên tệp (hàm này chỉ nên dùng cho kiểu nhị phân).
 - ftell cho biết vị trí hiện tại của con trỏ chỉ vị.
 - ferror cho biết có lỗi (khác 0) hay không lỗi (=0).
 - perror thông báo lỗi trên màn hình.
 - unlink và remove dùng để loại tệp trên đĩa

14

8.3. Các hàm cấp 2 (tiếp)

- Các hàm xuất nhập ký tự: dùng cho cả 2 kiểu
 - putchar và fputc dùng để ghi ký tự lên tệp
 - getchar và fgetc dùng để đọc ký tự từ tệp
- Các hàm xuất nhập theo kiểu văn bản:
 - fprintf dùng để ghi dữ liệu theo khuôn dạng lên tệp
 - fscanf dùng để đọc dữ liệu từ tệp theo khuôn dạng
 - fputs dùng để ghi một chuỗi ký tự lên tệp
 - fgets dùng để đọc một dãy ký tự từ tệp
- Các hàm xuất nhập theo kiểu nhị phân:
 - putw dùng để ghi một số nguyên (2 byte) lên tệp
 - getw dùng để đọc một số nguyên (2 byte) từ tệp
 - fwrite dùng để ghi một số mẫu tin lên tệp
 - fread dùng để đọc một số mẫu tin từ tệp

15

8.4. Đóng mở tệp, xóa vùng đệm và kiểm tra lỗi

- Dùng chung cho cả 2 kiểu nhị phân và văn bản
- 8.4.1. Hàm fopen: Mở tệp
 - Dạng hàm:
`FILE *fopen(const char *tên_tệp, const char *kiểu)`
 - Các đối:
 - Đối thứ nhất là tên tệp (có thể có đường dẫn đầy đủ)
 - Đối thứ hai là kiểu truy nhập, có các giá trị sau:

16

8.4.1. Hàm fopen: Mở tệp

Kiểu	Ý nghĩa
"r", "rt"	Mở 1 tệp để đọc theo kiểu văn bản. Tệp cần tồn tại nếu không sẽ có lỗi.
"w", "wt"	Mở 1 tệp mới để ghi theo kiểu văn bản. Nếu tệp đã tồn tại, nó sẽ bị xóa.
"a", "at"	Mở 1 tệp để ghi bổ sung theo kiểu văn bản. Nếu tệp chưa tồn tại thì tạo tệp mới.
"rb"	Mở 1 tệp để đọc theo kiểu nhị phân. Tệp cần tồn tại nếu không sẽ có lỗi.
"wb"	Mở 1 tệp mới để ghi theo kiểu nhị phân. Nếu tệp đã tồn tại, nó sẽ bị xóa.
"ab"	Mở 1 tệp để ghi bổ sung theo kiểu nhị phân. Nếu tệp chưa tồn tại thì tạo tệp mới.

17

8.4.1. Hàm fopen: Mở tệp (tiếp)

Kiểu	Ý nghĩa
"r+", "r+t"	Mở 1 tệp để đọc/ghi theo kiểu văn bản. Tệp cần tồn tại nếu không sẽ có lỗi.
"w+", "w+t"	Mở 1 tệp mới để đọc/ghi theo kiểu văn bản. Nếu tệp đã tồn tại, nó sẽ bị xóa.
"a+", "a+t"	Mở 1 tệp để đọc/ghi bổ sung theo kiểu văn bản. Nếu tệp chưa tồn tại thì tạo tệp mới.
"r+b"	Mở 1 tệp để đọc/ghi theo kiểu nhị phân. Tệp cần tồn tại nếu không sẽ có lỗi.
"w+b"	Mở 1 tệp mới để đọc/ghi theo kiểu nhị phân. Nếu tệp đã tồn tại, nó sẽ bị xóa.
"a+b"	Mở 1 tệp để đọc/ghi bổ sung theo kiểu nhị phân. Nếu tệp chưa tồn tại thì tạo tệp mới.

18

8.4.1. Hàm fopen: Mở tệp (tiếp)

- Công dụng: hàm dùng để mở tệp. Nếu thành công, hàm trả về con trỏ kiểu FILE ứng với tệp vừa mở. Các hàm cấp 2 sẽ làm việc với tệp thông qua con trỏ này. Nếu có lỗi hàm trả về giá trị NULL.
- Chú ý: Trong các kiểu đọc/ghi, cần làm sạch vùng đệm trước khi chuyển từ đọc sang ghi hoặc ngược lại. Dùng các hàm fflush và di chuyển đầu từ.

19

8.4.2. Hàm fclose: đóng tệp

- Dạng hàm: `int fclose(FILE *f);`
- Đối: f là con trỏ tương ứng với tệp cần đóng.
- Công dụng: hàm dùng để đóng tệp. Nội dung đóng tệp gồm:
 - đẩy dữ liệu còn trong vùng đệm lên đĩa (khi đang ghi)
 - xóa vùng đệm (khi đang đọc)
 - giải phóng biến f để nó có thể dùng cho tệp khác. Nếu thành công, hàm cho giá trị 0, trái lại hàm cho EOF.

20

8.4.3. Hàm fcloseall: đóng các tệp đang mở

- Dạng hàm: `int fcloseall(void);`
- Công dụng: hàm dùng để đóng tất cả các tệp đang mở. Nếu thành công, hàm cho giá trị nguyên bằng số tệp đóng được, trái lại hàm cho EOF.

21

8.4.4. Hàm fflush: làm sạch vùng đệm

- Dạng hàm: `int fflush(FILE *f);`
- Đối: f là con trỏ tệp
- Công dụng: hàm làm sạch vùng đệm của tệp f. Nếu thành công hàm cho giá trị 0, trái lại hàm cho EOF.

22

8.4.5. Hàm fflushall: làm sạch vùng đệm

- Dạng hàm: `int fflushall(void);`
- Công dụng: hàm dùng làm sạch vùng đệm của các tệp đang mở. Nếu thành công hàm cho giá trị nguyên bằng số tệp đang mở, trái lại hàm cho EOF.

23

8.4.6. Hàm feof: kiểm tra cuối tệp

- Dạng hàm `int feof(FILE *f);`
- Đối: f là con trỏ tệp
- Công dụng: hàm dùng để kiểm tra cuối tệp. Hàm cho giá trị khác 0 nếu gặp cuối tệp khi đọc, trái lại hàm cho giá trị 0.

24

8.4.7. Hàm perror: kiểm tra lỗi

- Dạng hàm: `int perror(FILE *f);`
- Đối: f là con trỏ tệp
- Công dụng: hàm dùng để kiểm tra lỗi thao tác trên tệp f. Hàm cho giá trị 0 nếu không lỗi, trái lại hàm cho giá trị khác 0.

25

8.4.8. Hàm perror: thông báo lỗi hệ thống

- Dạng hàm: `void perror(const char *s);`
- Đối: s là con trỏ tới một chuỗi ký tự
- Công dụng: hàm in chuỗi s và thông báo lỗi

26

8.4.9. Hàm unlink: xóa tệp

- Dạng hàm: `int unlink(const char *tên_tệp);`
- Đối: là tên tệp cần xóa
- Công dụng: hàm dùng để xóa 1 tệp trên đĩa. Nếu thành công, hàm cho giá trị 0, trái lại hàm cho giá trị EOF.

27

8.4.10. Hàm remove: xóa tệp

- Dạng hàm: `remove(const char *tên_tệp);`
- Đối: là tên tệp cần xóa.
- Công dụng: hàm dùng để xóa một tệp trên đĩa. Nó là hàm macro gọi tới unlink.

28

8.4.11. Ví dụ: mở 1 tệp và kiểm tra lỗi

```
FILE *fp;  
/*Mở tệp so_lieu để đọc theo kiểu nhị phân. Nếu  
thành công, con trỏ tệp so_lieu gán cho biến fp*/  
fp = fopen("so_lieu","rb");  
// Kiểm tra lỗi  
if(fp==NULL) perror("Lỗi khi mở tệp so_lieu");
```

29

8.5. Nhập xuất ký tự

- Dùng được cả trong kiểu nhị phân và văn bản nhưng tác dụng khác nhau.
- 8.5.1. Hàm putc và fputc
 - Dạng hàm:
`int putc(int ch, FILE *fp);`
`int fputc(int ch, FILE *fp);`
 - Đối: ch là một giá trị nguyên, fp là con trỏ tệp.
 - Công dụng: hàm ghi lên tệp fp một ký tự có mã bằng: `m = ch%256`, trong đó ch được xem là số nguyên không dấu. Nếu thành công hàm cho mã ký tự được ghi, trái lại hàm cho EOF

30

8.5.1. Hàm putc và fputc (tiếp)

- Ví dụ: câu lệnh `putc(-1,fp)`; sẽ ghi lên tệp fp mã 255 vì dạng không dấu của -1 là 65535.
- Ghi chú:
 - Hai hàm trên có ý nghĩa như nhau.
 - Trong kiểu văn bản, nếu `m = 10` thì hàm sẽ ghi lên tệp hai mã 13 và 10.

31

8.5.2. Hàm getc và fgetc

- Dạng hàm:
 - `int getc(FILE *fp);`
 - `int fgetc(FILE *fp);`
- Đối: fp là con trỏ tệp
- Công dụng: hàm đọc 1 ký tự từ tệp fp. Nếu thành công hàm cho mã đọc được (có giá trị từ 0 đến 255). Nếu gặp cuối tệp hay có lỗi hàm cho EOF
- Ghi chú:
 - hai hàm trên có ý nghĩa như nhau
 - trong kiểu văn bản, hàm đọc một lượt cả hai mã 13, 10 và trả về giá trị 10; khi gặp mã 26 thì hàm không trả về 26 mà trả về EOF

32

8.5.3. Ví dụ

```
■ Chương trình sao tệp chế độ nhị phân và dùng hàm fgetc, fputc
#include"stdio.h"
#include"stdlib.h"
void main(){
    int c;
    char t1[14], t2[14];
    FILE *f1, *f2;
    printf("\nTỆP NGUỒN:"); gets(t1);
    printf("\nTỆP ĐÍCH:"); gets(t2);
    f1=fopen(t1,"rb");
    if(f1==NULL){
        printf("\nTỆP %s không tồn tại",t1); getch(); exit(1);
    }
    f2=fopen(t2,"wb");
    while((c=fgetc(f1))!=EOF) fputc(c,f2);
    fclose(f1); fclose(f2);
}
```

33

8.5.3. Ví dụ (tiếp)

- Chương trình trên thực hiện sao tệp theo thuật toán sau:
 - bước 1: đọc 1 ký tự của tệp f1, kết quả đặt vào biến c
 - bước 2: nếu c bằng EOF thì kết thúc; nếu c khác EOF thì ghi c vào tệp f2 rồi quay trở lại bước 1.
- Nhận xét 1: nếu trong chương trình trên, ta thay bằng kiểu văn bản thì chỉ các byte đứng trước mã 26 đầu tiên của tệp f1 được sao sang tệp f2.
- Nhận xét 2: nếu dùng hàm feof và thuật toán:
 - bước 1: nếu feof(f1) khác 0 thì kết thúc, trái lại chuyển xuống bước 2.
 - bước 2: đọc 1 ký tự từ tệp f1, ghi lên tệp f2 thì ta có đoạn chương trình:

34

8.5.3. Ví dụ (tiếp)

- ```
while(!feof(f1)) fputc(fgetc(f1),f2);
```
- Đoạn chương trình này lại chưa thật đúng! Tệp f2 sẽ dài hơn tệp f1 đúng một byte có giá trị 255.
  - Lý do: giả sử tệp f1 có đúng một ký tự mã 65, khi đó thuật toán sẽ diễn ra như sau:
    - bước 1: đầu từ đang trỏ vào ký tự A nên feof(f) = 0, chuyển xuống bước 2.
    - bước 2: đọc ký tự A của f1 và ghi lên f2, trở lại bước 1.
    - bước 1: đầu đọc đặt ở cuối tệp f1 nhưng chưa có thao tác đọc nên feof(f1) vẫn bằng 0, chuyển xuống bước 2.
    - bước 2: đọc một ký tự của f1. Khi đó nhận được -1. Ghi -1 lên f2 thì mã 255 sẽ được ghi. Ngoài ra, do khi đọc từ f1 gặp phải cuối tệp nên lúc này feof(f1) khác 0. Đến đây thuật toán kết thúc.

35

### 8.6. Các hàm nhập xuất theo kiểu văn bản

- 8.6.1. Hàm fprintf: ghi dữ liệu theo khuôn dạng
  - Dạng hàm:
    - `int fprintf(FILE *f, const char *dk,...);`
  - Đối:
    - f là con trỏ tệp
    - dk chứa địa chỉ của chuỗi điều khiển
    - ... là danh sách các đối mà giá trị của chúng cần ghi lên tệp.
  - Công dụng: giá trị các đối được ghi lên tệp f theo khuôn dạng xác định trong chuỗi dk. Nếu thành công hàm trả về một giá trị nguyên bằng số byte ghi lên tệp, nếu có lỗi thì trả về EOF.
  - Nhận xét: Hàm làm việc giống hàm printf.

36

## Ví dụ hàm fprintf:

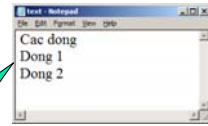
```
#include<stdio.h>
void main(){
 FILE *f;
 int i;
 f=fopen("text","wt");
 fprintf(f,"Cac dong");
 for(i=1;i<=2;i++) fprintf(f,"\nDong%2d",i);
 fclose(f);
}
```

37

## Ví dụ hàm fprintf (tiếp):

- Chương trình trên sẽ tạo ra tệp văn bản tên là **text** gồm 3 dòng với nội dung như sau:

Cac dong  
Dong 1  
Dong 2



Dùng trình notepad mở file **text** ra xem nội dung của nó

38

## 8.6.2. Hàm fscanf: đọc dữ liệu từ tệp theo khuôn dạng

- Dạng hàm:
  - `int fscanf(FILE *f, const char *dk, ...);`
- ĐỐI
  - f là con trỏ tệp
  - dk chứa địa chỉ của chuỗi điều khiển
  - ... là danh sách các đối sẽ chứa kết quả đọc được từ tệp
- Công dụng: đọc dữ liệu từ tệp f, biến đổi theo khuôn dạng trong dk và lưu kết quả vào các đối. Hàm trả về một giá trị bằng số trường được đọc.
- Nhận xét: Hàm làm việc giống hàm scanf.

39

## Ví dụ 1 về hàm fscanf

- Giả sử có tệp văn bản "da\_giac.sl" chứa thông tin về một đa giác. Tệp gồm n+1 dòng với nội dung như sau:

Dòng 1: n (số đỉnh)  
Dòng 2: x1 y1 (tọa độ đỉnh 1)  
Dòng 3: x2 y2 (tọa độ đỉnh 2)  
....  
Dòng n+1: xn yn (tọa độ đỉnh n)

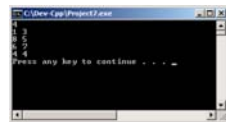


- Chương trình sau sẽ đọc số đỉnh và tọa độ các đỉnh từ tệp "da\_giac.sl"

40

## Ví dụ 1 về hàm fscanf (tiếp)

```
#include<stdio.h>
void main(){
 FILE *f; int i,n,x[10],y[10];
 f=fopen("da_giac.sl","rt");
 fscanf(f,"%d",&n);
 for(i=1;i<=n;i++) fscanf(f,"%d%d",&x[i],&y[i]);
 fclose(f);
 // Hiện thị ra màn hình để kiểm tra
 printf("%d\n",n);
 for(i=1;i<=n;i++) printf("\n%d %d",x[i],y[i]);
 getch();
}
```



Kết quả khi chạy chương trình

41

## Ví dụ 2 về hàm fscanf

- Giả sử có một dãy số nguyên ghi trên tệp văn bản "songuyen.txt". Giữa hai số nguyên có ít nhất một khoảng trống hay các dấu xuống dòng. Yêu cầu đọc và in ra màn hình dãy số nói trên.
- Ta phân biệt 2 trường hợp:
  - Sau chữ số cuối cùng là mã 26 hay cuối tệp
  - Sau chữ số cuối cùng có ít nhất một khoảng trống hay các dấu xuống dòng.

42

### Ví dụ 2 về hàm fscanf (tiếp)

- Trường hợp 1: sau chữ số cuối cùng là mã 26 hay kết thúc tệp

```
#include<stdio.h>
void main(){
FILE *f; int c;
f=fopen("songuyen.txt","r");
while(!feof(f)){
fscanf(f,"%d",&c);
printf("%d\n",c);
}
fclose(f);
getch();
}
```

43

### Ví dụ 2 về hàm fscanf (tiếp)

- Trường hợp 2: sau chữ số cuối cùng có ít nhất một khoảng trống hay các dấu xuống dòng

```
#include<stdio.h>
void main(){
FILE *f; int c;
f=fopen("songuyen.txt","r");
while(1){
fscanf(f,"%d",&c);
if(!feof(f)) break;
printf("%d\n",c);
}
fclose(f);
getch();
}
```

44

### Phân tích ví dụ 2

- Nếu với trường hợp thứ hai mà ta lại dùng đoạn mã cho trường hợp thứ nhất thì sao?
- Kết quả:

45

### Phân tích ví dụ 2

- Nếu với trường hợp thứ nhất mà ta lại dùng đoạn mã cho trường hợp thứ hai thì sao?
- Kết quả:

46

### 8.6.3. Hàm fputs: ghi một chuỗi ký tự lên tệp

- Dạng hàm:
  - int fputs(const char \*s, FILE \*f);
- ĐỐI:
  - s là con trỏ trỏ tới địa chỉ đầu của một chuỗi ký tự kết thúc bằng dấu '\0'.
  - f là con trỏ tệp.
- Công dụng: ghi chuỗi s lên tệp f (dấu '\0' không ghi lên tệp). Nếu thành công hàm trả về ký tự cuối cùng được ghi lên tệp; nếu có lỗi hàm trả về EOF.

47

### Ví dụ hàm fputs

- Chương trình sau sẽ nhập các dòng ký tự từ bàn phím và ghi lên tệp "vanban"

```
#include<stdio.h>
#include<conio.h>
void main(){
int i=0; char d[256]; FILE *f;
f=fopen("vanban","w");
clrscr();
printf("Bam Enter de ket thuc");
while(1){
i++;
printf("\nDong %d:",i); gets(d);
if(d[0]=='\0') break; // Bam Enter de ket thuc
if(i>1) fputs(d,f);
fputs(d,f);
}
fclose(f);
}
```

48

### 8.6.4. Hàm fgets: đọc một dãy ký tự từ tệp

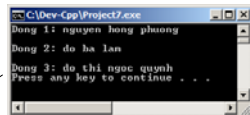
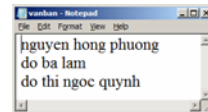
- Dạng hàm:
  - char \*fgets(char \*s, int n, FILE \*f);
- Đối:
  - s là con trỏ tới vùng nhớ đủ lớn để chứa chuỗi ký tự sẽ đọc từ tệp.
  - n là số nguyên xác định độ dài cực đại của dãy cần đọc.
  - f là con trỏ tệp.
- Công dụng: đọc 1 dãy ký tự từ tệp f chứa vào vùng nhớ s. Việc đọc kết thúc khi:
  - hoặc đã đọc n-1 ký tự
  - hoặc gặp dấu xuống dòng (cặp mã 13 10). Khi đó mã 10 được đưa vào xâu kết quả.
  - hoặc kết thúc tệp.
- Xâu kết quả sẽ được bổ sung thêm dấu hiệu kết thúc chuỗi '\0'. Khi thành công hàm trả về địa chỉ vùng nhận kết quả; khi có lỗi hoặc gặp cuối tệp, hàm cho giá trị NULL.

49

### Ví dụ hàm fgets

- Chương trình đọc các dòng ký tự trên tệp "vanban" và đưa ra màn hình.

```
#include<stdio.h>
#include<conio.h>
void main(){
 int i=0; char d[256]; FILE *f;
 f=fopen("vanban","r"); clrscr();
 while(!feof(f)){
 i++;
 fgets(d,256,f);
 printf("Dong %d: %s\n",i,d);
 }
 fclose(f);
 getch();
}
```



Kết quả hiển thị

50

### 8.7. Tệp văn bản và các thiết bị chuẩn

- Có thể dùng các hàm nhập xuất văn bản trên các thiết bị chuẩn. C đã định nghĩa các tệp tin và con trỏ ứng với các thiết bị chuẩn như sau:

| Tệp | Con trỏ | Thiết bị                      |
|-----|---------|-------------------------------|
| in  | stdin   | Thiết bị vào chuẩn (bàn phím) |
| out | stdout  | Thiết bị ra chuẩn (màn hình)  |
| err | stderr  | Thiết bị lỗi chuẩn (màn hình) |
| prn | stdprn  | Thiết bị in chuẩn (máy in)    |

- Khi chương trình C bắt đầu làm việc thì các tệp này được tự động mở, vì vậy có thể dùng các con trỏ nêu trên để nhập xuất trên các thiết bị chuẩn

51

### 8.7. .... ví dụ

```
#include<stdio.h>
#include<conio.h>
void main(){
 char ht[25]; float diem; int ns;
 printf("\nHo ten: ");fgets(ht,25,stdin);
 printf("\nDiem va nam sinh");
 fscanf(stdin,"%f%d",&diem,&ns);
 fputs(ht,stderr);
 fprintf(stdout,"Diem %f nam sinh %d",diem, ns);
}
```

52

### 8.8. Các hàm nhập xuất theo kiểu nhị phân

#### 8.8.1. Hàm putw: ghi một số nguyên

- Dạng hàm: int putw(int n, FILE \*f);
- Đối:
  - n là giá trị nguyên
  - f là con trỏ tệp
- Công dụng: ghi giá trị n lên tệp f dưới dạng 2 byte. Nếu thành công hàm trả về số nguyên được ghi; nếu có lỗi hàm trả về EOF.

53

#### 8.8.2. Hàm getw: đọc một số nguyên

- Dạng hàm: int getw(FILE \* f);
- Đối: f là con trỏ tệp.
- Công dụng: đọc một số nguyên 2 byte từ tệp f. Nếu thành công, hàm trả về số nguyên đọc được; nếu có lỗi hoặc gặp cuối tệp, hàm trả về EOF.

54

## Ví dụ về hàm putw và getw

- Chương trình ghi một dãy số nguyên lên tệp "songuyen", sau đó đọc các số nguyên từ tệp này và đưa ra màn hình.

```
#include<stdio.h>
#include<conio.h>
void main(){
 FILE *f; int i;
 // Ghi các số nguyên
 f=fopen("songuyen","wb");
 for(i=1000;i<=1010;i++) putw(i,f);
 fclose(f);
 // Đọc các số nguyên
 clrscr();
 f=fopen("songuyen","rb");
 while((i=getw(f)!=EOF) printf("\n%d",i);
 fclose(f);
}
```

55

## 8.8.3. Hàm fwrite: ghi các mẫu tin lên tệp

- Dạng hàm:
  - int fwrite(void \*ptr, int size, int n, FILE \*f);
- Đối:
  - ptr là con trỏ tới vùng nhớ chứa dữ liệu cần ghi.
  - size là kích thước của mẫu tin theo byte.
  - n là số mẫu tin cần ghi.
  - f là con trỏ tệp.
- Công dụng: ghi n mẫu tin kích thước size byte từ vùng nhớ ptr lên tệp f. Hàm trả về giá trị bằng số mẫu tin thực sự được ghi.

56

## 8.8.4. Hàm fread: đọc các mẫu tin từ tệp tin

- Dạng hàm:
  - int fread(void \*ptr, int size, int n, FILE \*f);
- Đối:
  - ptr là con trỏ tới vùng nhớ sẽ chứa dữ liệu đọc được từ tệp tin.
  - size là kích thước của mẫu tin theo byte.
  - n là số mẫu tin cần đọc.
  - f là con trỏ tệp.
- Công dụng: đọc n mẫu tin kích thước size byte từ tệp f chứa vào vùng nhớ ptr. Hàm trả về một giá trị bằng số mẫu tin thực sự đọc được.

57

## Ví dụ về fwrite, fread

```
■ Ví dụ 1: sao chép tệp dùng fwrite, fread
#include<stdio.h>
#include<process.h>
void main(){
 int n; char t1[20], t2[20], c[1000];
 FILE *f1, *f2;
 printf("\nTEP NGUON: ");gets(t1);
 printf("\nTEP DICH:");gets(t2);
 f1=fopen(t1,"rb");
 if(f1==NULL){
 printf("\nTEP %s khong ton tai",t1);
 getch(); exit(1);
 }
 f2=fopen(t2,"wb");
 while((n=fread(c,1,1000,f1))>0) fwrite(c,1,n,f2);
 fclose(f1); fclose(f2);
}
```

58

## Ví dụ về fwrite, fread

```
■ Ví dụ 2: ghi và đọc một dãy n phần tử số thực
#include<stdio.h>
#include<conio.h>
void main(){
 FILE *f; float a[20],b[20]; int i,n;
 // Nhập số phần tử n
 do{
 printf("Nhap so phan tu n = ");scanf("%d",&n);
 }while((n<1)||(n>20));
 // Nhập vào n phần tử thực
 for(i=0;i<n;i++){
 printf("\na[%d]= ",i); scanf("%f",&a[i]);
 }
}
```

59

## Ví dụ về fwrite, fread

```
f=fopen("mangsolieu","wb");
// Ghi n phần tử thực của mảng a vào file f
fwrite(a,sizeof(float),n,f);
fclose(f);
f=fopen("mangsolieu","rb");
// Đọc n phần tử thực từ file f đưa vào mảng b
fread(b,sizeof(float),n,f);
// Hiện thị ra màn hình
for(i=0;i<n;i++){
 printf("\nb[%d]=%f",i,b[i]);
}
fclose(f);
getch();
}
```

60

## Ví dụ về fwrite, fread

- Ví dụ 3: ghi và đọc cấu trúc

```
#include<stdio.h>
#include<conio.h>
typedef struct{
 char ht[25];
 float diem;
}HOCSINH;
void main(){
 FILE *f; HOCSINH hs;
 // Nhập số liệu từ bàn phím và ghi lên tệp
 f=fopen("HOSO.DAT","wb");
 printf("Bấm Enter để kết thúc");
 while(1){
```

61

## Ví dụ về fwrite, fread

```
printf("\nHo va ten: ");gets(hs.ht);
if(hs.ht[0]=='\0') break;
printf("\nDiem so: ");scanf("%f%c",&hs.diem);
fwrite(&hs,sizeof(HOCSINH),1,f);
}
fclose(f);
f=fopen("HOSO.DAT","rb");
while(fread(&hs,sizeof(HOCSINH),1,f)>0)
 printf("\n%s %f",hs.ht,hs.diem);
fclose(f);
getch();
}
```

62

## 8.9. Nhập xuất ngẫu nhiên và các hàm di chuyển con trỏ chỉ vị

- Mỗi tệp khi đang mở có một con trỏ chỉ vị dùng để xác định vị trí đọc/ghi trên tệp.
- Khi mở tệp tin để đọc/ghi, con trỏ chỉ vị luôn ở đầu tệp tin. Nhưng nếu mở theo chế độ "a" thì con trỏ chỉ vị ở cuối tệp để ghi thêm dữ liệu vào tệp.
- Việc xuất nhập dữ liệu được thực hiện từ vị trí hiện tại của con trỏ chỉ vị và sau khi hoàn thành thì con trỏ này dịch chuyển đi một số byte bằng số byte đã đọc hay ghi.
- Việc xuất nhập được tiến hành tuần tự từ đầu đến cuối tệp tin.

63

### 8.9.1. Hàm rewind: chuyển con trỏ chỉ vị về đầu tệp

- Dạng hàm: **void rewind(FILE \*f);**
- Đối: f là con trỏ tệp.
- Công dụng: chuyển con trỏ chỉ vị của tệp f về đầu tệp. Khi đó, việc nhập xuất trên tệp f được thực hiện từ đầu tệp.

64

### 8.9.2. Hàm fseek: di chuyển con trỏ chỉ vị đến vị trí mong muốn

- Dạng hàm:
  - **int fseek(FILE \*f, long sb, int xp);**
- Đối:
  - f là con trỏ tệp.
  - sb là số byte cần di chuyển.
  - xp cho biết vị trí xuất phát mà việc dịch chuyển được bắt đầu từ đây. xp cps thể nhận các giá trị sau:
    - SEEK\_SET hay 0: xuất phát từ đầu tệp.
    - SEEK\_CUR hay 1: xuất phát từ vị trí hiện tại của con trỏ chỉ vị.
    - SEEK\_END hay 2: xuất phát từ cuối tệp.

65

### 8.9.2. Hàm fseek (tiếp)

- Công dụng: hàm di chuyển con trỏ chỉ vị của tệp f từ vị trí xác định bởi xp qua một số byte bằng giá trị tuyệt đối của sb. Chiều di chuyển về cuối tệp nếu sb dương, trái lại di chuyển về đầu tệp. Khi thành công, hàm trả về giá trị 0; nếu có lỗi hàm trả về giá trị khác 0.
- Chú ý: Không nên dùng fseek trên kiểu văn bản.

66

### 8.9.3. Hàm ftell: cho biết vị trí hiện tại của con trỏ chỉ vị

- Dạng hàm: **long ftell(FILE \*f);**
- Đối: f là con trỏ tệp.
- Công dụng: khi thành công, hàm cho biết vị trí hiện tại của con trỏ chỉ vị (byte thứ mấy trên tệp f). Số thứ tự byte được tính từ 0. Khi có lỗi, hàm trả về -1L.

67

### Ví dụ: fseek và ftell

- Chương trình dùng fseek và ftell xác định độ dài của tệp.

```
#include<stdio.h>
#include<conio.h>
#include<process.h>
void main(){
 FILE *f; long n; char ten[25]; clrscr();
 puts("Ten tep: "); gets(ten);
 f=fopen(ten,"rb");
 if(f==NULL){
 printf("\nTep %s khong ton tai",ten);exit(1);
 }
 fseek(fp,0,SEEK_END); n=ftell(f); fclose(f);
 printf("\nDo dai cua tep %s la %ld byte",ten,n);
 getch();
}
```

68

### Bài tập

#### ■ Bài 1:

Viết chương trình:

- Nhập từ bàn phím N số thực lưu vào một mảng (N < 100 và N được nhập từ bàn phím).
- Sau đó ghi ra một file văn bản có tên là "float.txt" theo quy cách: dòng đầu tiên lưu số lượng các số thực, các dòng tiếp theo lưu các số thực, mỗi số lưu trên một dòng.
- Đọc lại tệp văn bản đó và lưu các số thực đọc được vào một mảng.
- Sắp xếp các số thực trong mảng theo thứ tự tăng dần và ghi ra một tệp văn bản khác có tên là "floatsx.txt" theo quy cách giống như tệp "float.txt".

69

### Bài tập

#### ■ Bài 2: Viết chương trình ghép nối nội dung 2 file:

- Nhập vào từ bàn phím 2 chuỗi ký tự là đường dẫn của file nguồn và file đích
- Ghép nội dung của file nguồn vào cuối file đích.

70

### Hỏi-đáp



71

### Lời hay ý đẹp



72



# TIN HỌC ĐẠI CƯƠNG

**Thời lượng: 60 tiết**

## Chương I

# KIẾN THỨC CHUNG VỀ CNTT VÀ TRUYỀN THÔNG

Bài 1: Các khái niệm cơ bản

Bài 2: Cấu trúc cơ bản của hệ thống máy tính

Bài 3: Biểu diễn thông tin trong máy tính

# BÀI 1: CÁC KHÁI NIỆM CƠ BẢN

- ◆ Thông tin và xử lý thông tin
- ◆ Phần cứng, phần mềm, công nghệ phần mềm

# I. Thông tin và xử lý thông tin

- ◆ Thông tin
- ◆ Dữ liệu
- ◆ Xử lý thông tin

# 1. Thông tin

- ◆ Những hiểu biết có được về một sự vật hiện tượng gọi là thông tin
- ◆ Có 2 dạng thông tin
  - Loại số: Số nguyên, số thực
  - Loại phi số
    - Văn bản
    - Hình ảnh
    - Âm thanh

## 2. Dữ liệu

- ◆ Dữ liệu là thông tin được lưu trong máy tính.
- ◆ Để thông tin có thể lưu được trong máy tính ta cần biến đổi thông tin thành dãy bit (dãy 2 số 0,1). Cách biến đổi như vậy được gọi là mã hoá thông tin.

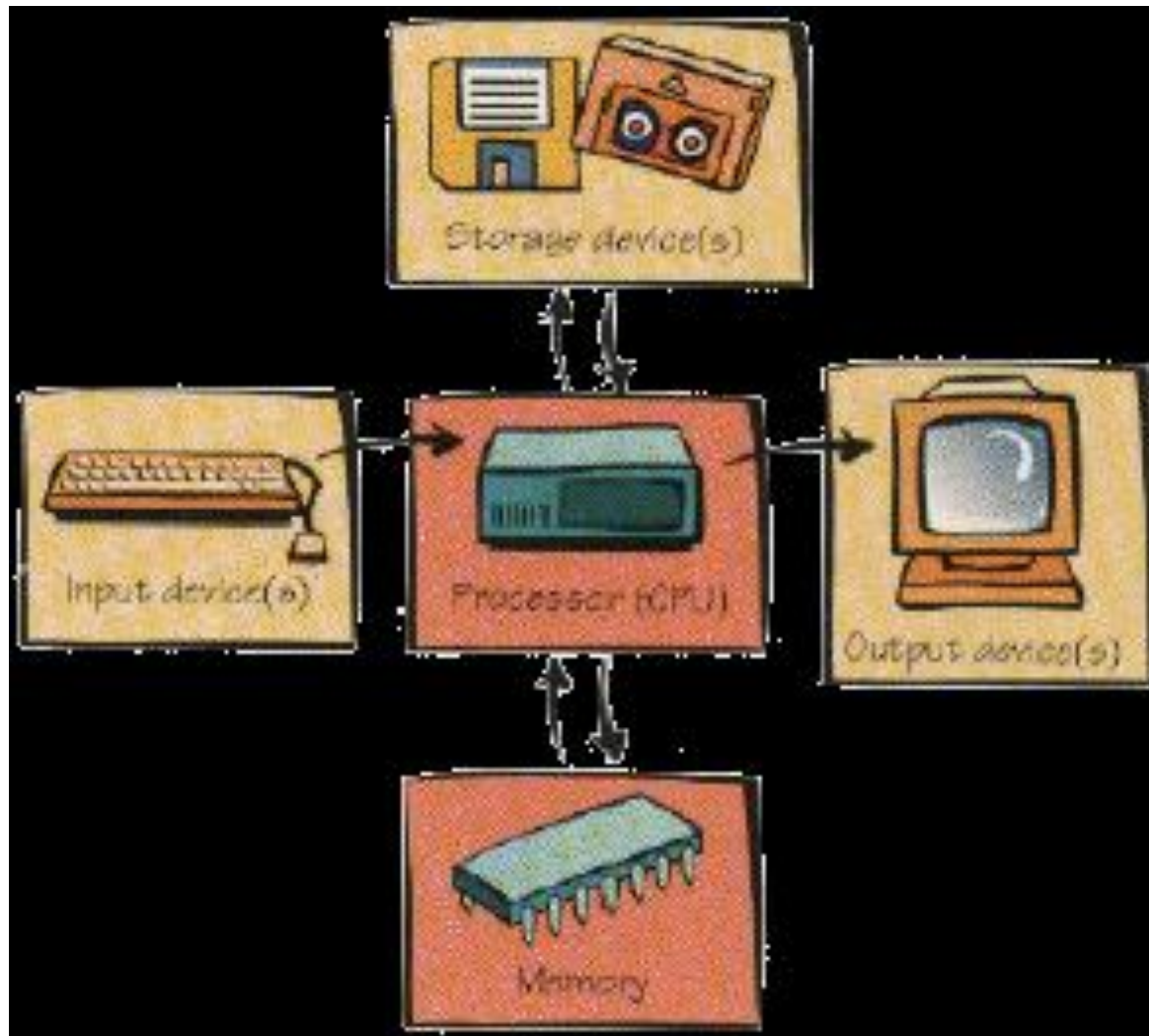
# 3. Xử lý thông tin

- ◆ Quá trình xử lý thông tin gồm 4 giai đoạn:
  - *Nhận thông tin (Receive input)*: thu nhận thông tin từ thế giới bên ngoài vào máy tính. Thực chất đây là quá trình chuyển đổi các thông tin ở thế giới thực sang dạng biểu diễn thông tin trong máy tính thông qua các thiết bị đầu vào.
  - *Xử lý thông tin (process information)*: biến đổi, phân tích, tổng hợp, tra cứu... những thông tin ban đầu để có được những thông tin mong muốn

# 3. Xử lý thông tin

- *Xuất thông tin (produce output)* : đưa các thông tin kết quả (đã qua xử lý) ra trở lại thế giới bên ngoài. Đây là quá trình ngược lại với quá trình ban đầu, máy tính sẽ chuyển đổi các thông tin trong máy tính sang dạng thông tin ở thế giới thực thông qua các thiết bị đầu ra
- *Lưu trữ thông tin (store information)*: ghi nhớ lại các thông tin đã được ghi nhận để có thể đem ra sử dụng trong những lần xử lý về sau.





**Quá trình xử lý thông tin trên máy**

## II. HỆ ĐIỀU HÀNH

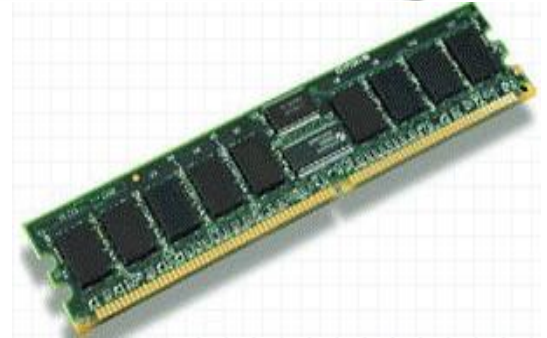
Phần cứng

Phần mềm

Công nghệ thông tin

# 1. Phần cứng

- ◆ Gồm những đối vật lý hữu hình, như vi mạch, bảng mạch, dây cáp nối mạch, bàn phím, chuột, thiết bị đầu cuối, ...
- ◆ Phần cứng xử lý thông tin ở mức xử lý thấp nhất tức là các tín hiệu nhị phân  $\{0,1\}$



*Monitor CRT*

## 2. Phần mềm

- ◆ Là các chương trình (program) điều khiển các hoạt động của phần cứng máy vi tính và chỉ đạo việc xử lý dữ liệu
- ◆ Phần mềm được chia làm 2 loại:
  - Phần mềm hệ thống (system software)
  - Phần mềm ứng dụng (application software)

## a. Phần mềm hệ thống

- ◆ Khi đưa vào bộ nhớ chính của máy tính, nó chỉ đạo máy tính thực hiện các công việc.
- ◆ Ví dụ: Hệ điều hành Windows...

## b. Phần mềm ứng dụng

- ♦ Là các chương trình được thiết kế để giải quyết một bài toán hay một vấn đề cụ thể để đáp ứng nhu cầu riêng trong một số lĩnh vực.
  - Các phần mềm trong lĩnh vực giải trí...
  - Các phần mềm trong lĩnh vực lập trình ứng dụng...
  - Các phần mềm trong lĩnh vực kinh tế...

# 3. Công nghệ thông tin

- ◆ Công nghệ thông tin là tập hợp các ngành khoa học kỹ thuật nhằm giải quyết vấn đề:
  - Thu nhận thông tin,
  - Quản lý thông tin,
  - Xử lý thông tin,
  - Truyền thông tin
  - Cung cấp thông tin.



# a. Xác định hệ thống thông tin

- ◆ Xác định các thể loại thông tin, yêu cầu về chất lượng.
- ◆ Xác định các chuẩn thông tin
- ◆ Xác định hệ thống phần cứng và phần mềm hệ thống
- ◆ Xây dựng tổ chức cho toàn hệ thống

## **b. Thu nhận thông tin**

- ◆ Kỹ thuật đo đạc để lấy số liệu
- ◆ Tổ chức hệ thống thống kê số liệu thông qua bộ máy quản lý của ngành
- ◆ Tổ chức hệ thống cập nhật dữ liệu

## c. Quản lý thông tin

- ◆ Xây dựng hệ thống cơ sở dữ liệu
- ◆ Hệ quản trị cơ sở dữ liệu

## d. Xử lý thông tin

- ◆ Phân tích và tổng hợp hệ thống thông tin
- ◆ Giải các bài toán ứng dụng chuyên ngành

## e. Truyền thông tin

- ◆ Xây dựng hệ thống đường truyền thông tin
- ◆ Giải pháp truyền thông tin trên mạng
- ◆ Hệ quản trị mạng thông tin
- ◆ Bảo vệ an toàn trên đường truyền thông tin
- ◆ Bảo mật thông tin

## g. Cung cấp thông tin

- ◆ Xây dựng giao diện với người sử dụng
- ◆ Hiển thị thông theo nhu cầu
- ◆ Tổ chức mạng dịch vụ thông tin

# Xu hướng phát triển CNTT

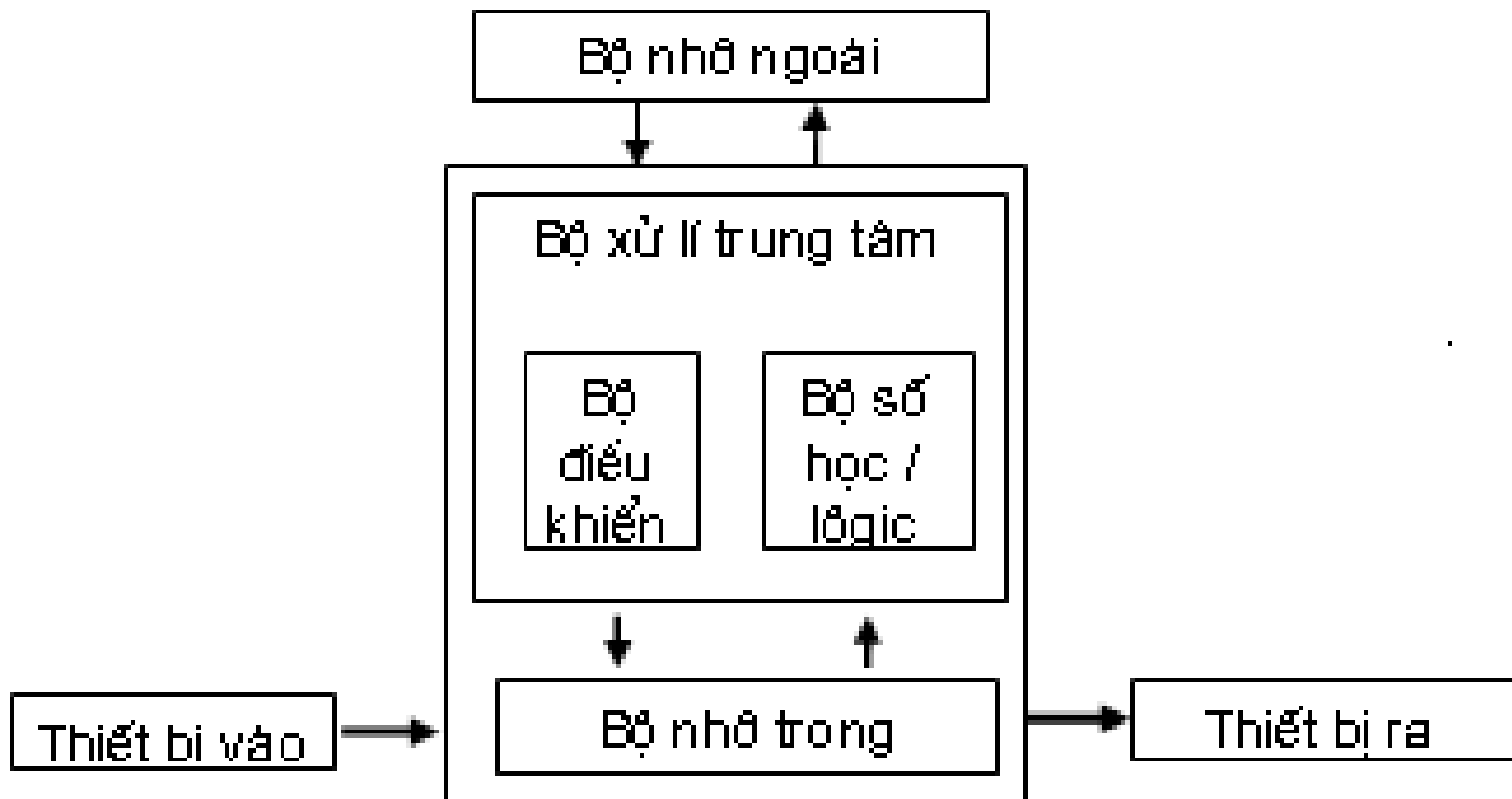
- ◆ Nhu cầu đa dạng hoá thông tin
- ◆ Nhu cầu chính xác hoá thông tin
- ◆ Xu hướng phát triển phần cứng và phần mềm hệ thống
- ◆ Sự phát triển của kỹ thuật xử lý thông tin
- ◆ Sự phát triển trong xây dựng cơ sở dữ liệu
- ◆ Sự phát triển trong kỹ thuật truyền tin
- ◆ Sự phát triển trong kỹ thuật thu nhận và cung cấp thông tin

## **Bài 2: CẤU TRÚC CƠ BẢN CỦA HỆ THỐNG**

- ◆ Mô hình cấu trúc máy tính
- ◆ Phần cứng
- ◆ Phần mềm



# 1. Sơ đồ cấu trúc máy tính



## 2. Phần cứng

### a. Đơn vị xử lý thông tin (CPU)

(Central Processing Unit)

CPU là thành phần quan trọng nhất của máy tính, thực hiện các lệnh của chương trình khi phần mềm nào đó chạy, tốc độ xử lý của máy tính phụ thuộc chủ yếu vào linh kiện này, CPU là linh kiện nhỏ nhưng đắt nhất trong máy vi tính.

## b. Thiết bị nhập

- ♦ Là những thiết bị nhập dữ liệu vào máy tính như bàn phím, chuột, máy quét, máy scan..



## c. Thiết bị xuất (Output device)

- ♦ Là những thiết bị hiển thị và xuất dữ liệu từ máy tính. Thiết bị xuất bao gồm màn hình, đèn chiếu, máy in.



*Monitor CRT*



*Monitor LCD*



## d. Bộ nhớ và thiết bị lưu trữ (Storage device)

- ◆ Là các thiết bị lưu trữ dữ liệu bao gồm bộ nhớ trong và bộ nhớ ngoài.
- ◆ **Bộ nhớ trong** bao gồm bộ nhớ chỉ đọc ROM, bộ nhớ truy xuất ngẫu nhiên RAM



➤ **Bộ nhớ ngoài:** bao gồm ổ cứng, đĩa mềm, đĩa CD, DVD, ổ cứng USB, thẻ nhớ và các thiết bị lưu trữ khác.



## 2. Phần mềm

- ◆ Phần mềm hệ thống (System software)
- ◆ Phần mềm ứng dụng (Application software)

## a. Phần mềm hệ thống

- ◆ Khi đưa vào bộ nhớ chính của máy tính, nó chỉ đạo máy tính thực hiện các công việc.
- ◆ Ví dụ: Hệ điều hành Windows...



## b. Phần mềm ứng dụng

- ◆ Là các chương trình được thiết kế để giải quyết một bài toán hay một vấn đề cụ thể để đáp ứng nhu cầu riêng trong một số lĩnh vực.
  - Các phần mềm trong lĩnh vực giải trí...
  - Các phần mềm trong lĩnh vực lập trình ứng dụng...
  - Các phần mềm trong lĩnh vực kinh tế...

## Chương II

# HỆ ĐIỀU HÀNH

Bài 4. Hệ điều hành MS-DOS

Bài 5: Giới thiệu Windows

Bài 6: Những thao tác cơ bản trên Windows

# Bài 4. Hệ điều hành MS-DOS

- ◆ MS-DOS là gì?
- ◆ Tên ổ đĩa và dấu đợi lệnh
- ◆ Tập và thư mục
- ◆ Các lệnh về đĩa

# 1. MS-Dos là gì?

## ◆ Hệ điều hành là gì?

- Là phần mềm quan trọng nhất trong máy tính.
- Có nhiệm vụ điều khiển mọi hoạt động cơ sở của máy tính
- Giúp phần mềm của người sử dụng có thể chạy trên máy tính
- VD: Hệ điều hành Windows, MS-DOS, Unix, Ninux, OS/2

# MS-Dos là gì?

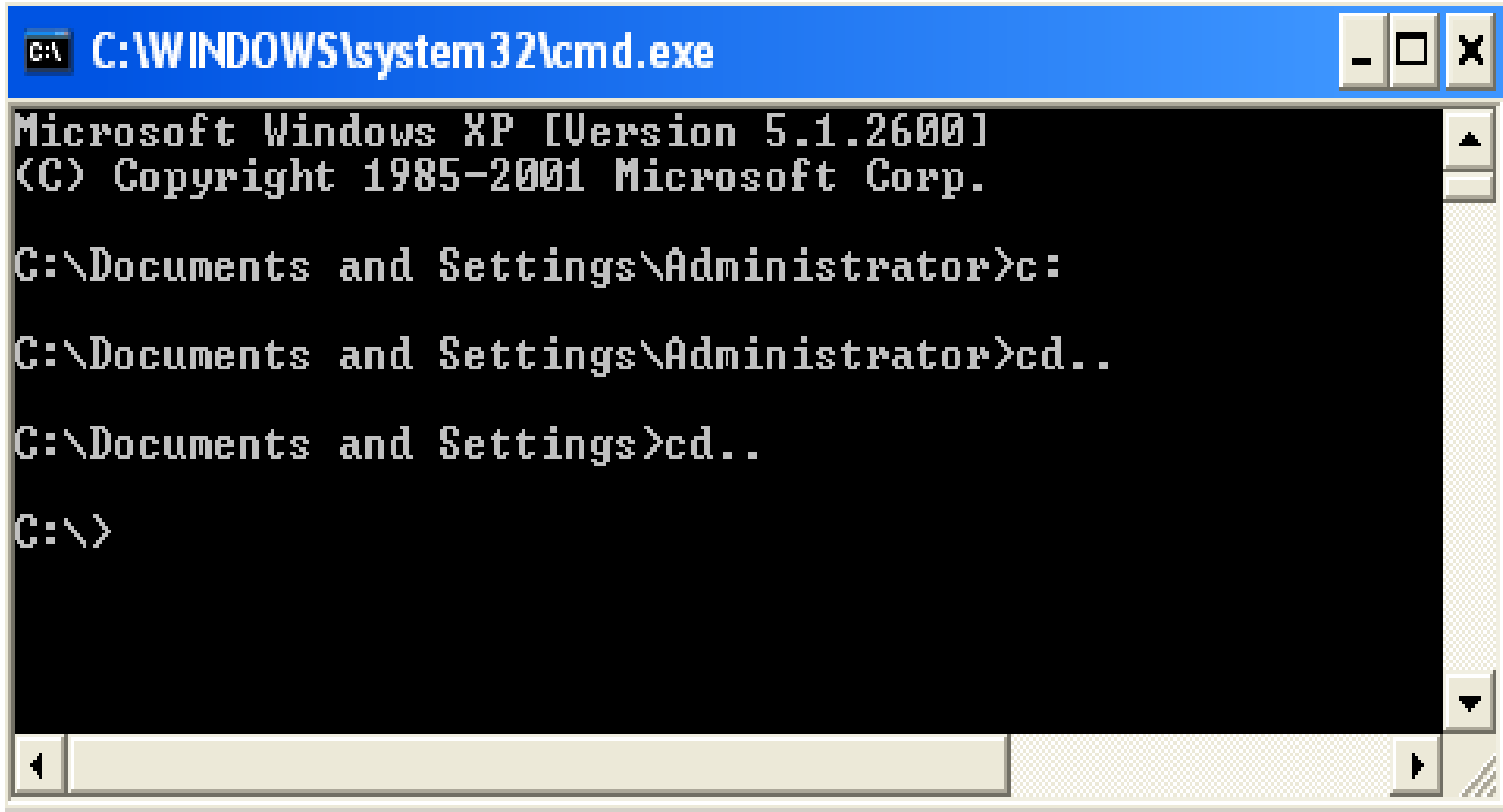
- ◆ Microsoft disk Operating System là Hệ điều hành (HĐH) của tập đoàn Microsoft, phiên bản đầu tiên viết vào năm 1981.
- ◆ MS-Dos là HĐH đơn nhiệm.
- ◆ MS-Dos giao tiếp với người sử dụng thông qua dòng lệnh.

## b. Khởi động hệ thống

- ◆ Để khởi động hệ thống ta phải có một đĩa mềm gọi là đĩa hệ thống hoặc đĩa cứng được cài đặt ổ đĩa C là đĩa hệ thống
- ◆ Đĩa hệ thống chứa các chương trình hạt nhân của hệ điều hành MS DOS
- ◆ Trên đĩa phải có các tập tin: IO.sys, MSDOS.sys, Command.com

# Khởi động hệ thống MS-Dos

- ◆ C1: Khởi động từ ổ đĩa cứng: bật công tắc điện của máy tính
- ◆ C2: Khởi động từ ổ đĩa mềm: đặt đĩa mềm vào giá đỡ của ổ đĩa mềm, bật công tắc điện.
- ◆ C3: Khởi động từ HĐH Windows: start/run/cmd/OK

A screenshot of a Windows XP command prompt window. The title bar is blue and contains the text 'C:\WINDOWS\system32\cmd.exe' and standard window control buttons (minimize, maximize, close). The main area is black with white text. The text shows the following sequence of commands and their outputs:

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator>c:
C:\Documents and Settings\Administrator>cd..
C:\Documents and Settings>cd..
C:\>
```

The window also features a vertical scrollbar on the right and a horizontal scrollbar at the bottom.

Môi trường làm việc của HĐH MS-DOS



## 2. Ổ đĩa, dấu đợi lệnh

### a. Ổ đĩa: bao gồm

- ◆ Ổ đĩa mềm - gọi là ổ đĩa A: có dung lượng 1.4 MB.
- ◆ Ổ đĩa cứng – thường là ổ C,D,E, nằm trong thùng máy và có dung lượng lớn gấp nhiều lần ổ đĩa mềm
- ◆ Ổ đĩa CD – dùng để đọc các đĩa quang, có dung lượng vài trăm MB.

## b. Dấu đợi lệnh

- ◆ Sau khi thực thi các lệnh khởi động, màn hình sẽ xuất hiện dấu đợi lệnh có dạng **C:\>\_** hoặc **A:\>\_** ;
- ◆ **:\>** là quy ước dấu đợi lệnh của DOS
- ◆ Điểm nháy sáng **\_** là con trỏ (cursor): cho ta biết điểm làm việc hiện tại trên màn hình
- ◆ Các ký tự gõ trên bàn phím sẽ xuất hiện tại vị trí con trỏ

# 3. Tập và thư mục

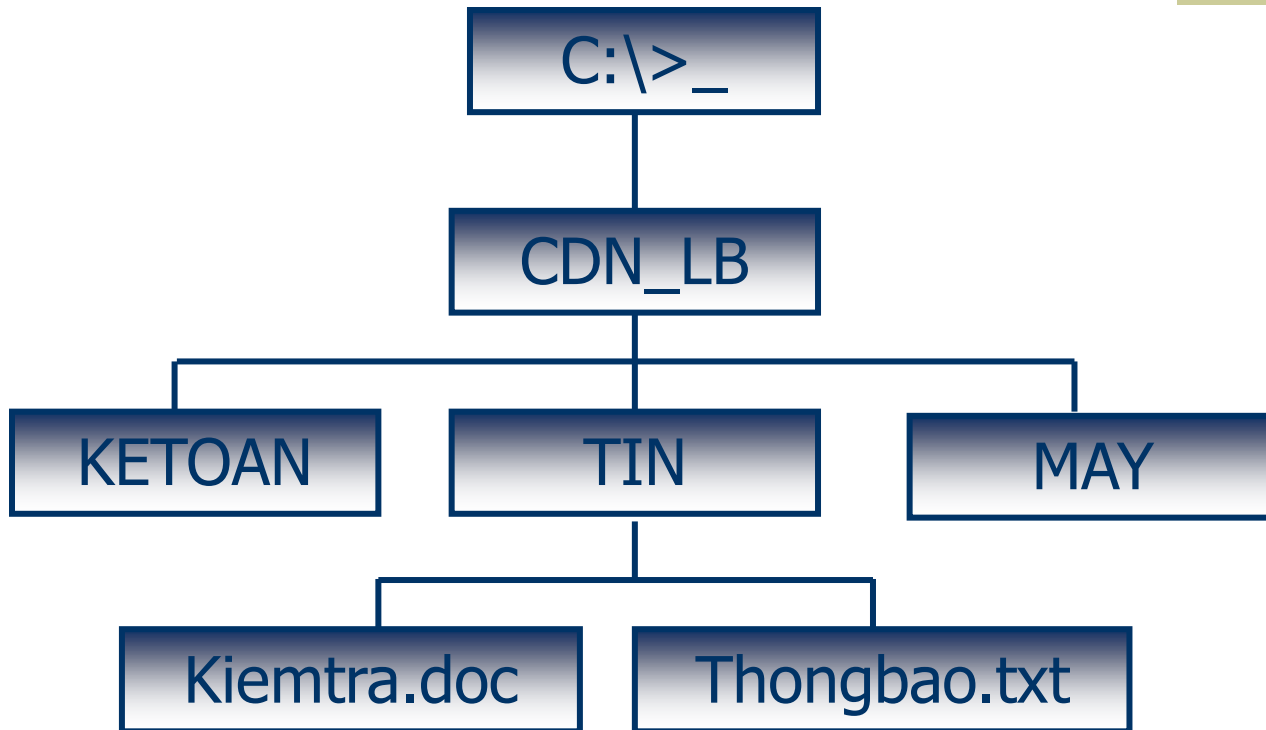
## a. Tập:

- ◆ Tập (File) là nơi lưu trữ thông tin trên đĩa của HĐH.
- ◆ Tập gồm có tên tập và phần mở rộng (phần mở rộng để nhận biết tập đó do chương trình nào tạo ra)
- ◆ Tên tập tin không quá 8 ký tự, không chứa dấu cách, +, -, \*, /. Phần mở rộng không quá 3 ký tự không chứa dấu cách. Giữa tên và phần mở rộng cách nhau bởi dấu chấm (.).

## b. Thư mục

- ◆ Để có thể tổ chức và quản lý tốt tệp tin trên đĩa, người ta lưu các tệp tin thành từng nhóm và lưu vào từng chỗ riêng gọi là thư mục
- ◆ Mỗi thư mục được đặc trưng bởi một cái tên cụ thể, quy tắc đặt tên giống đặt tên cho tệp.
- ◆ Các thư mục có thể đặt lồng trong nhau, tạo thành cây thư mục
- ◆ Thư mục gốc là thư mục do định dạng đĩa tạo ra, ta không thể xoá, mỗi đĩa chỉ có một thư mục gốc, kí hiệu là dấu (\)

# Ví dụ:



# 4. Các lệnh về đĩa

## a. Lệnh định dạng đĩa

- ◆ Dùng để tạo dạng cho đĩa mềm hay đĩa cứng.
- ◆ Cú pháp

**[d] [path] FORMAT [d1] [/tham số]**

- **[d] [path]**: ổ đĩa, đường dẫn đến tên lệnh
- **[d1]**: tên ổ đĩa cần định dạng
- Tham số:
  - + **/s**: tạo đĩa hệ thống
  - + **/u**: format mà sau đó không thể dụng lệnh UNFORMAT để lấy lại dữ liệu
  - + **/q**: định dạng nhanh

# Bài 5: Hệ điều hành Windows

- ◆ Windows là gì?
- ◆ Khởi động và thoát khỏi windows
- ◆ Desktop
- ◆ Thanh Tác vụ
- ◆ Menu start
- ◆ Khởi động và thoát khỏi một ứng dụng.
- ◆ Chuyển đổi giữa các ứng dụng
- ◆ Các thành phần trong cửa sổ
- ◆ Các thao tác với chuột.

# 1. Windows là gì?

- ◆ Là HĐH của tập đoàn Microsoft, phiên bản đầu tiên ra đời vào năm 1985.
- ◆ Là HĐH đa nhiệm
- ◆ Giao tiếp với người dùng chủ yếu qua giao diện đồ họa đó là các cửa sổ, hộp thoại.
- ◆ Đã có rất nhiều phiên bản của Windows được ra đời, ngày nay chủ yếu dùng Windows XP, Win2000 server.



# Đặc điểm của HĐH Windows

- ◆ Hệ thống thực đơn thống nhất cung cấp cho người dùng một cách thức chuẩn để gọi thực hiện các chức năng của ứng dụng.
- ◆ Có thể thực hiện thao tác copy, paste dữ liệu ở bất cứ đâu trong hệ thống
- ◆ Hệ thống thanh cuộn cho phép người dùng hiển thị văn bản liên tục
- ◆ Gọi thi hành các ứng dụng đơn giản
- ◆ Cung cấp tính năng Multimedia cho phép người dùng nghe nhạc, xem phim.
- ◆ Cung cấp khả năng đa nhiệm
- ◆ Cung cấp khả năng làm việc trong môi trường mạng

## 2. Khởi động và thoát khỏi Win

### a. Khởi động

- ◆ Nhấn nút Power, máy tính sẽ tự động khởi động windows

## b. Thoát khỏi windows

- ◆ Chọn **start** -> **turn off** -> chọn 1 trong 3 giá trị sau:
  - **Turn off:** Thoát khỏi windows và đóng các chương trình đang mở
  - **Stand by:** tạm thời thoát khỏi window, chuyển sang chế độ làm việc tiết kiệm năng lượng nhất
  - **Restart:** thoát khỏi windows bằng cách đóng các chương trình và tự động khởi động lại máy.

# 3. Desktop

- ◆ Sau khi khởi động máy tính bằng windows, hệ điều hành sẽ chạy một chương trình quản lý có tên là Explorer.exe. Chương trình này giữ nhiệm vụ chính trong việc giao tiếp với người dùng
- ◆ Màn hình chính của Explorer là màn hình Desktop, trên màn hình này thường có một số biểu tượng sau:
  - My Document, My Computer
  - Recycle bin
  - Internet Explorer
  - My Network Place



Màn hình Desktop

## 4. Thanh tác vụ (task bar)

- ◆ Là thanh ngang nằm dưới đáy màn hình Desktop gồm 3 phần:
  - **Phần bên trái:** là nút Start hiển thị menu chính của Windows
  - **Phần bên phải:** vùng hiển thị ngày giờ và biểu tượng thu gọn của các ứng dụng đang được thi hành
  - **Phần giữa:** liệt kê các ứng dụng đang chạy trong hệ thống, mỗi ứng dụng là một nút mà người dùng có thể nhấp chuột để chọn làm việc với ứng dụng đó

# 5. Menu start

- ◆ Là nơi giúp người dùng truy cập và sử dụng tất cả các chức năng mà Windows cung cấp
- ◆ Các mục chọn trong menu Start bao gồm:
  - **Shutdown**: tắt máy tính
  - **Run**: chạy một ứng dụng
  - **Search**: tìm kiếm đối tượng
  - **Settings**: thiết lập thông số hệ thống
  - **Programs**: các chương trình ứng dụng được cài đặt
  - **Document**: lưu đường dẫn các tập tin đã được mở.


## 6. Khởi động và thoát khỏi ứng dụng

### a. Khởi động ứng dụng

- ◆ Để khởi động một ứng dụng đã được cài đặt trong máy ta dùng một trong các cách sau:
  - Start -> programs -> chọn ứng dụng
  - Start -> run -> nhập tên ứng dụng
  - Nháy đúp chuột vào biểu tượng của ứng dụng cần mở.




## b. Thoát khỏi ứng dụng

- ◆ Chọn nút Close  trên thanh tiêu đề
- ◆ Nhấn tổ hợp phím Alt+X hoặc Alt + F4
- ◆ File -> Exit

## c. Chuyển đổi giữa các cửa sổ

- ◆ Nhấn tổ hợp phím ALT+Tab
- ◆ Chọn ứng dụng cần làm việc trên thanh task bar.

## d. Thu nhỏ một cửa sổ ứng dụng

- ◆ Chọn nút  Minimize trên thanh tiêu đề
- ◆ Nháy phải chuột lên thanh tiêu đề -> chọn Minimize

# 7. Thao tác sử dụng chuột

- ◆ Nhấp – Click: là thao tác bấm một phím của chuột. Nhấp chuột trái là bấm phím trái của chuột 1 lần, ngược lại với thao tác nhấp phải chuột.
- ◆ Nhấp đúp – Double click: là thao tác bấm một phím của chuột hai lần liên tiếp. Khi nói nhấp đúp thì thường chỉ dùng nút chuột trái
- ◆ Chọn – Select: là thao tác nhấp chuột trên một đối tượng nào đó
- ◆ Kéo thả - Drag and Drop: là thao tác chọn một đối tượng bằng cách nhấn giữ nút trái chuột sau đó kéo đối tượng đến một vị trí mới rồi thả nút trái ra

# Bài 6:

## Những thao tác cơ bản trên Windows

- ◆ File và Folder
  - Tạo, đổi tên, xoá...
  - Copy, cut, move...
- ◆ Quản lý tài nguyên
  - My Computer
  - Windows Explorer

# 1. File and Folder

## a. Tạo thư mục

- Mở thư mục chứa đối tượng cần tạo
- Từ File -> New -> Folder
- Đặt tên cho đối tượng
- Gõ Enter

## b. Đổi tên:

- ◆ Chọn đối tượng cần đổi tên
- ◆ Từ File -> Rename
- ◆ Đặt tên mới cho đối tượng
- ◆ Gõ Enter

## c. Xoá

- ◆ Chọn đối tượng cần xoá
- ◆ Từ File -> Delete
- ◆ Chọn Yes nếu muốn xoá, No hủy bỏ thao tác xoá
- ◆ Gõ Enter



## d. Sao chép (Copy):

- ◆ Chọn đối tượng cần sao chép
- ◆ Từ Edit -> Copy
- ◆ Mở thư mục chứa đối tượng sẽ được copy tới
- ◆ Từ Edit -> Paste

## e. Di chuyển (Move)

- ◆ Chọn đối tượng cần di chuyển
- ◆ Từ Edit -> Cut
- ◆ Mở thư mục chứa đối tượng sẽ được cut tới
- ◆ Từ Edit -> Paste

## g. Một số thao tác khác

- ◆ Thay đổi chế độ liệt kê danh sách tệp, thư mục.
  - View -> Details
  - Lựa chọn các hình thức liệt kê phù hợp
    - Details: hiển thị chi tiết các thông tin về đối tượng
    - Titles: Hiển thị thông tin về đối tượng: tên, type. Size
    - Icon: Hiển thị các đối tượng theo biểu tượng
    - Thumbnails: Hiển thị nội dung hình ảnh của đối tượng

## g. Một số thao tác khác

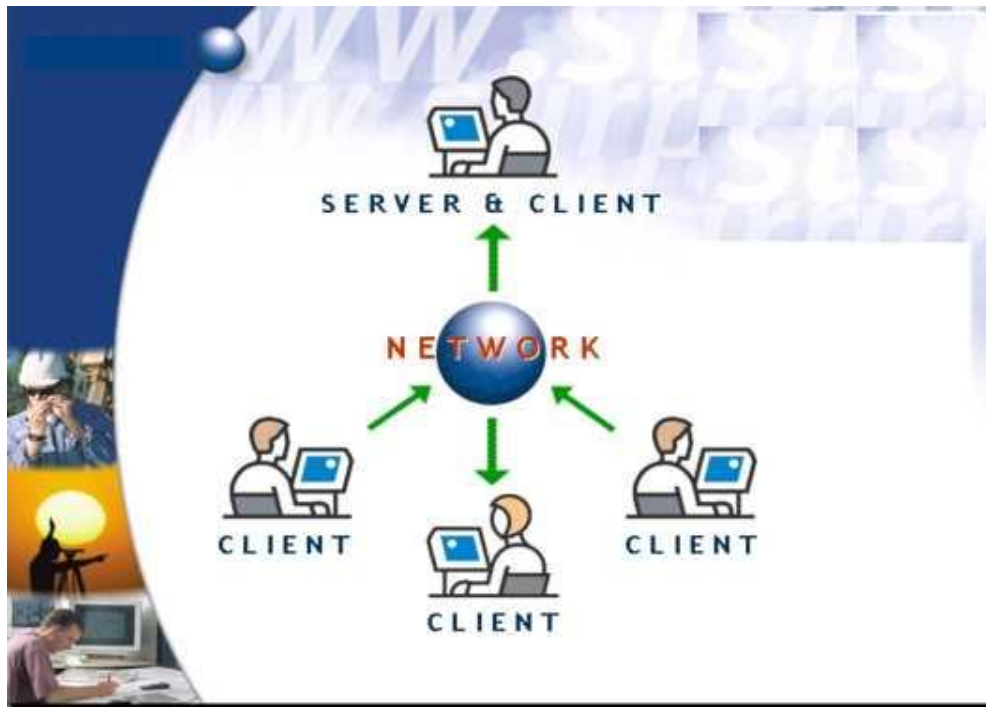
- ◆ Sắp xếp các đối tượng trong chế độ Details.
  - Nhấp chuột trên tiêu đề của cột cần sắp xếp

# Một số thao tác khác

- ◆ Thiết lập thuộc tính cho đối tượng.
  - Chọn đối tượng cần thiết lập
  - File -> Properties
  - Thiết lập các thông số trong hộp thoại
    - Read Only: Chỉ đọc
    - Hidden: Ẩn

## Chương III.

# MẠNG CƠ BẢN VÀ INTERNET



# Bài 7. Mạng cơ bản

- ◆ Những khái niệm cơ bản
- ◆ Phân loại mạng
- ◆ Các thiết bị mạng

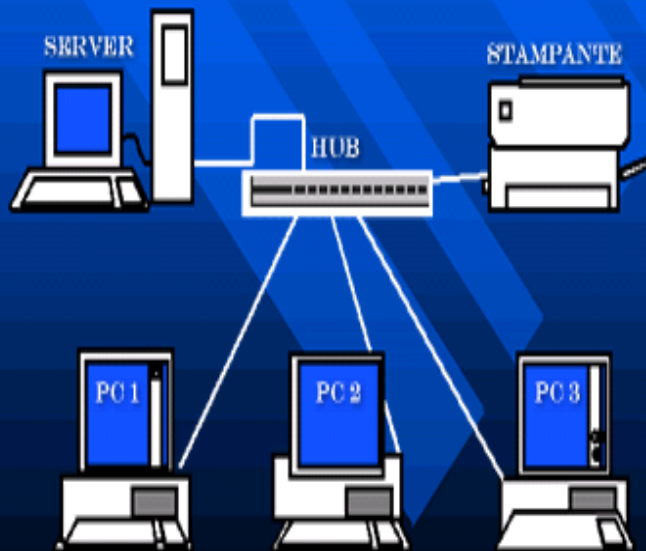
# 1. Những khái niệm cơ bản

## a. Vai trò của mạng máy tính

- ◆ Sao chép và truyền dữ liệu
- ◆ Chia sẻ tài nguyên (thông tin, thiết bị)
- ◆ Tạo thành hệ thống tính toán lớn



## b. Các thành phần trong mạng



- Các máy tính
- Các thiết bị mạng đảm bảo kết nối các máy tính với nhau.
- Phần mềm cho phép thực hiện việc giao tiếp giữa các máy tính

## c. Giao thức truyền thông của mạng máy tính

- Giao thức truyền thông là bộ các quy tắc cụ thể phải tuân thủ trong việc trao đổi thông tin trong mạng giữa các thiết bị nhận và truyền dữ liệu. Ví dụ: Giao thức về tốc độ truyền, khuôn dạng dữ liệu, kiểm soát lỗi...
- Bộ giao thức truyền thông được dùng phổ biến hiện nay trong các mạng, đặc biệt trong mạng toàn cầu Internet là TCP/IP.

## 2. Phân loại mạng máy tính

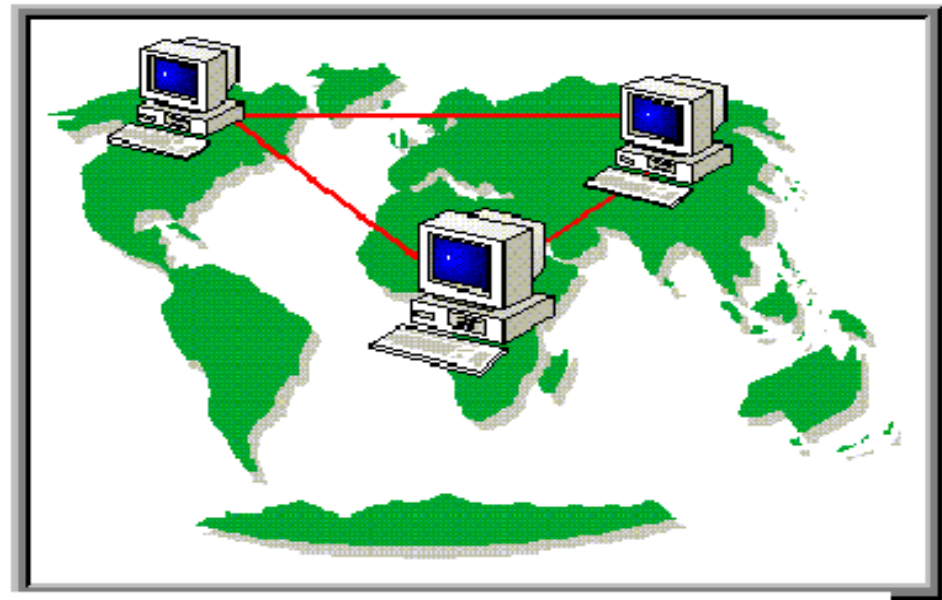
- ◆ Phân bố theo phạm vi địa lý
- ◆ Phân bố theo mô hình mạng
- ◆ Phân bố theo kỹ thuật chuyển mạch

## a. Phân bố theo phạm vi địa lý

- ◆ **Mạng cục bộ - LAN (Local Area Network):**  
Là mạng kết nối các máy tính ở gần nhau, ví dụ: trong một phòng, một toà nhà, một xí nghiệp...

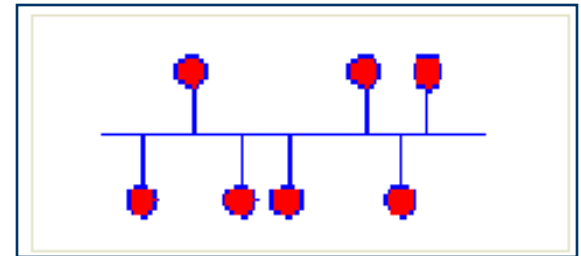
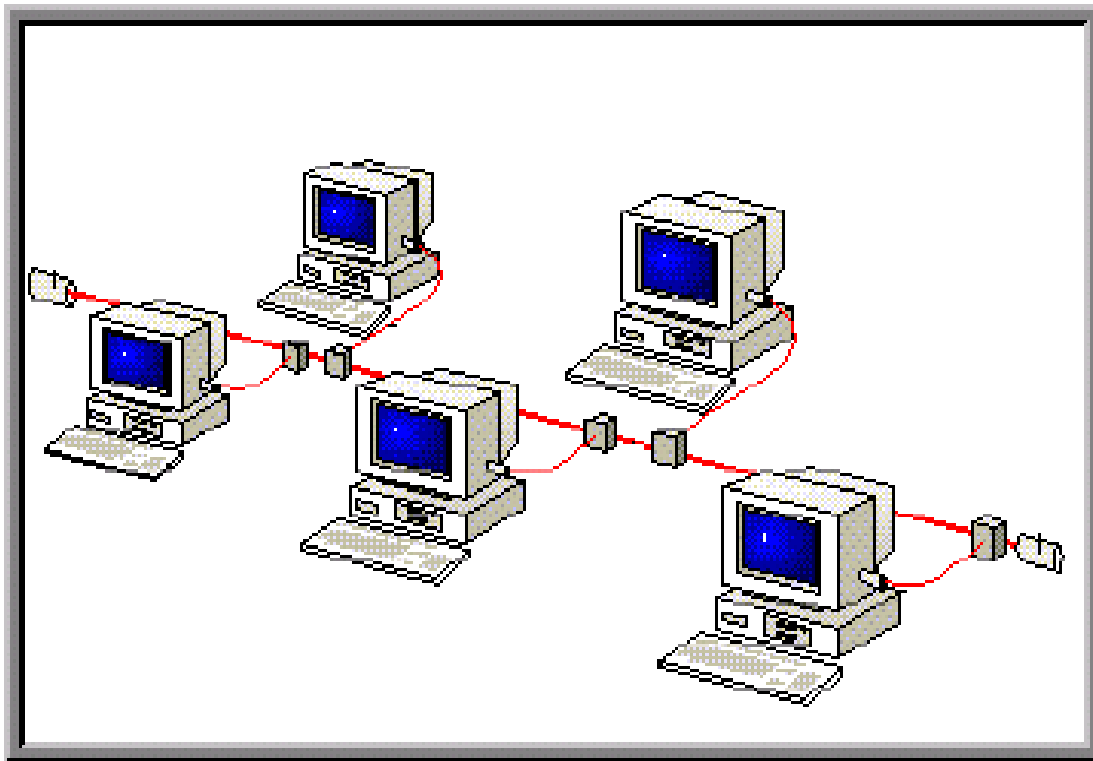
## a. Phân bố theo phạm vi địa lý

- ◆ **Mạng diện rộng - WAN (Wide Area Network):** Là mạng kết nối các máy tính cách nhau một khoảng cách lớn, mạng diện rộng thường liên kết các mạng cục bộ.



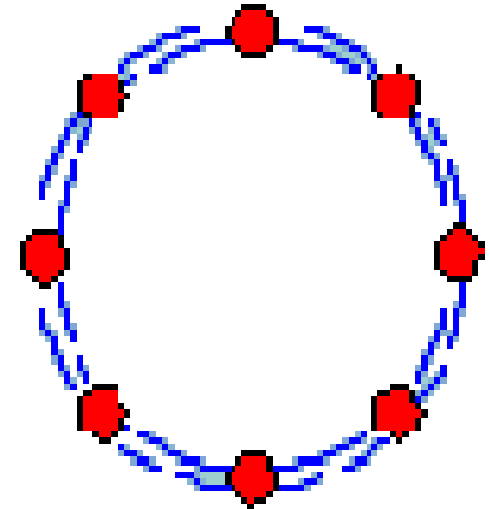
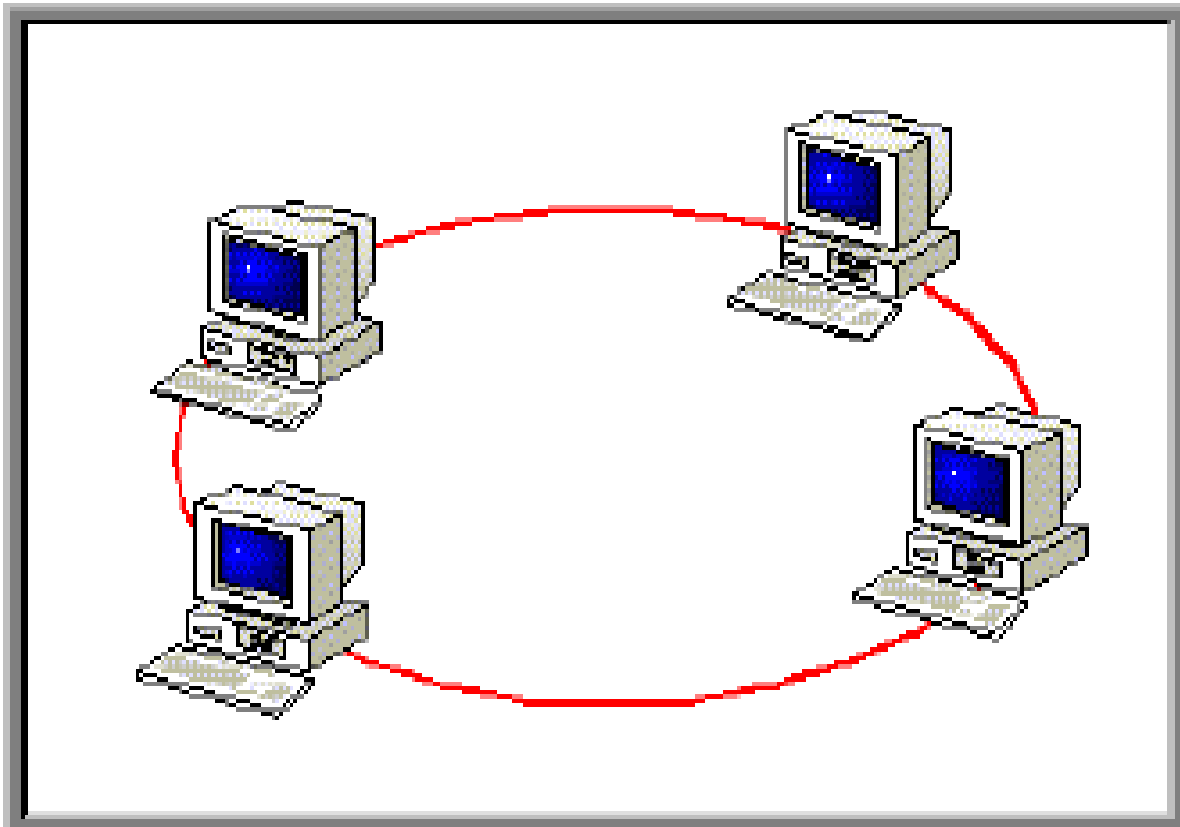
## b. Phân loại theo kỹ thuật chuyển mạch

- ◆ Kiểu đường thẳng



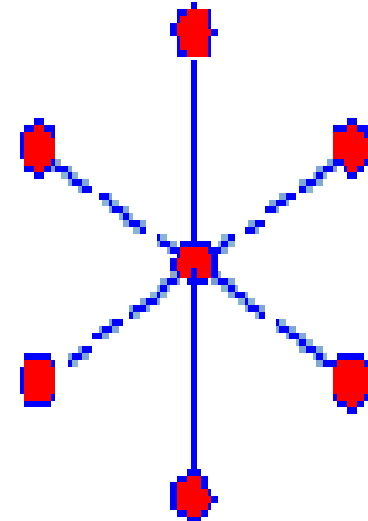
## b. Phân loại theo kỹ thuật chuyển mạch

- ◆ Kiểu đường vòng



## b. Phân loại theo kỹ thuật chuyển mạch

- ◆ Kiểu hình sao



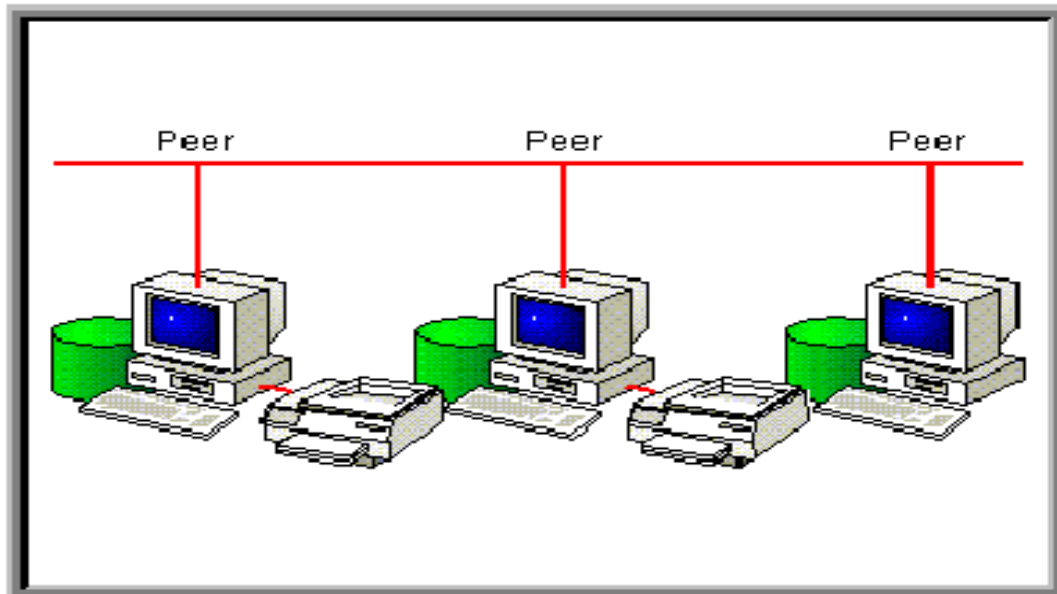


# Nhận xét

- ◆ Với mỗi kiểu cần có các loại thiết bị mạng khác nhau, chi phí xây dựng, sự phát triển mạng, cách thức quản lý của các mạng cũng khác nhau.

## c. Phân loại theo mô hình mạng

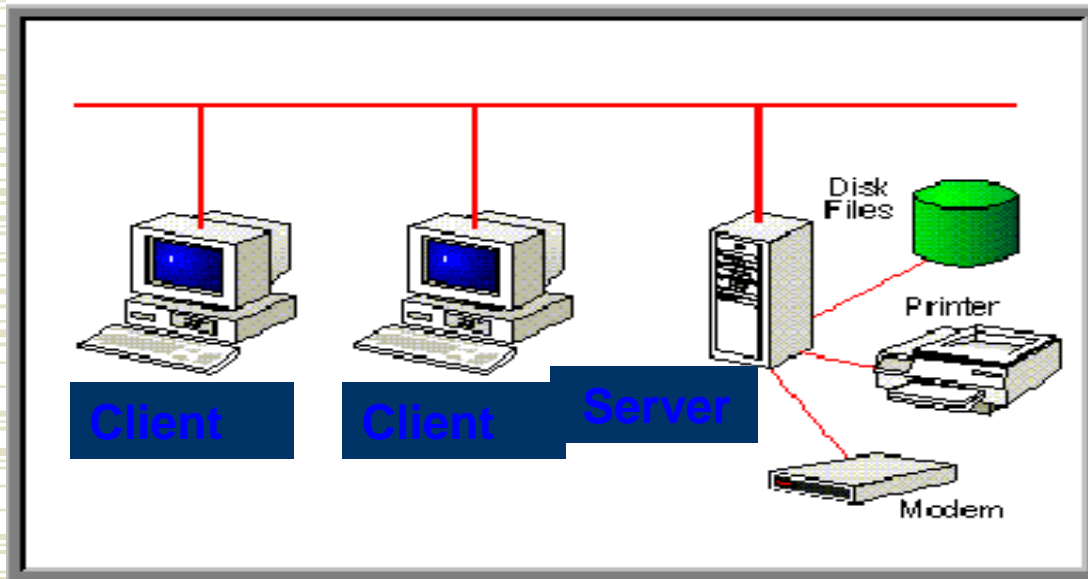
- ◆ Mô hình mạng ngang hàng (Peer to peer)



*Mô hình này có ưu điểm là xây dựng và bảo trì đơn giản, song chỉ thích hợp với các mạng có quy mô nhỏ dữ liệu phân tán.*  
dụng tài nguyên của máy khác trong mạng

## c. Phân loại theo mô hình mạng

### ◆ Mô hình khách chủ (Client Server)



**Client** – Máy khách sử dụng tài nguyên do máy chủ cung cấp.

**Server** – Máy chủ đảm bảo phục vụ các máy khác bằng cách điều khiển việc phân bố tài nguyên.

*Mô hình này có ưu điểm là dữ liệu được quản lý tập trung, chế độ bảo mật tốt thích hợp với các mạng có quy mô lớn và trung bình.*

# 3. Các thiết bị mạng

- ◆ Network Card
- ◆ Hub
- ◆ Modem
- ◆ Repeater
- ◆ Bridge
- ◆ Switch
- ◆ Router
- ◆ Gateway



Gateway

***Hub***

***Bridge***

***Router***

***Cáp mạng***

***Switch***

***Vỏ mạng***

## a. Network Card

- ◆ Là thiết bị nối máy tính với các thiết bị mạng khác
- ◆ Thiết bị được gắn trên cổng PCI



***Vĩ mạng***

## b. Repeater

- ◆ Là thiết bị dùng để khuếch đại và định thời lại tín hiệu. Giúp tín hiệu có thể truyền dẫn đi xa hơn.
- ◆ Điện tín, điện thoại, truyền thông tin qua sợi quang... và các nhu cầu truyền tín hiệu đi xa đều cần sử dụng Repeater



*Repeater*

## c. Hub

- ◆ Được coi là một Repeater có nhiều cổng. Một Hub có từ 4 đến 24 cổng và có thể còn nhiều hơn.
- ◆ Được dùng trong các mạng 10aBASE-T; 100BASE-T



***Hub***



## d. Bridge

- ◆ Bridge được sử dụng để ghép nối 2 mạng để tạo thành một mạng lớn duy nhất. Bridge được sử dụng phổ biến để làm cầu nối giữa hai mạng Ethernet
- ◆ Ưu điểm của Bridge là hoạt động trong suốt.
- ◆ Nhược điểm: Chỉ kết nối những mạng cùng loại.



**Bridge**

## e. Switch

- ◆ Được coi như một Bridge nhưng có nhiều cổng
- ◆ Switch hoạt động ở tốc độ cao hơn nhiều so với Repeater .



*switch*

## g. Router

- ◆ Router kết nối hai hay nhiều mạng IP với nhau.
- ◆ Ưu điểm: có thể kết nối các loại mạng khác lại với nhau.
- ◆ Nhược điểm: chậm hơn so với Bridge.



***Router***

# h. Gateway

- ◆ Cho phép nối ghép hai loại giao thức với nhau.
- ◆ Qua Gateway, các máy tính trong các mạng sử dụng các giao thức khác nhau có thể dễ dàng "nói chuyện" được với nhau



*Gateway*

# Bài 8

## Khái thác và sử dụng Internet

- ◆ Tổng quan về Internet
- ◆ Dịch vụ WWW (World Wide Web)
- ◆ Thư điện tử (Email)

# 1. Tổng quan về Internet



## a. Khái niệm:

- ◆ Là mạng máy tính toàn cầu, kết nối hàng triệu máy tính và mạng máy tính trên khắp thế giới và sử dụng bộ giao thức TCP/IP

## b. Vai trò của Internet

**Bách khoa toàn thư** (còn được gọi là **từ điển bách khoa** hay **bách khoa thư**; **La tinh trung cổ**: **encyclopaedia/encyclopaedia**; **tiếng Anh**: **encyclop(a)edia/cyclop(a)edia**) là bộ sách tra cứu về nhiều lĩnh vực **kiến thức** nhân loại. Từ này xuất phát ban đầu từ **tiếng Hy Lạp cổ**, viết là **εγκύκλιος παιδεία** (*enkuklios paideia*), nghĩa là "kiến thức phổ thông/đại chúng". Trong đó **εγκύκλιος** ("phổ biến rộng rãi"), **κύκλος** ("vòng tròn/phạm vi") và **παιδεία** ("giáo dục").

Bách khoa toàn thư có thể là bộ sách tra cứu kiến thức tổng quát trên nhiều lĩnh vực kiến thức khác nhau (ví dụ bộ *Encyclopædia Britannica* bằng **tiếng Anh** và bộ *Brockhaus* bằng **tiếng Đức** là những bộ khá nổi tiếng), hoặc có thể là bộ sách tra cứu chuyên sâu về một lĩnh vực nào đó (ví dụ như bách khoa toàn thư về **y học**, **triết học**, hoặc **luật**). Cũng có những bộ bách khoa toàn thư đề cập đến nhiều chủ đề khác nhau trên một góc độ kiến thức, chẳng hạn như bộ *Đại*

**WHEN IN DOUBT—"LOOK IT UP" IN**  
The  
*Encyclopaedia Britannica*  
**The Sum of Human Knowledge**  
29 volumes, 28,150 pages,  
44,000,000 words of text.  
Printed on thin, but strong  
opaque India paper, each  
volume but one inch in  
thickness.

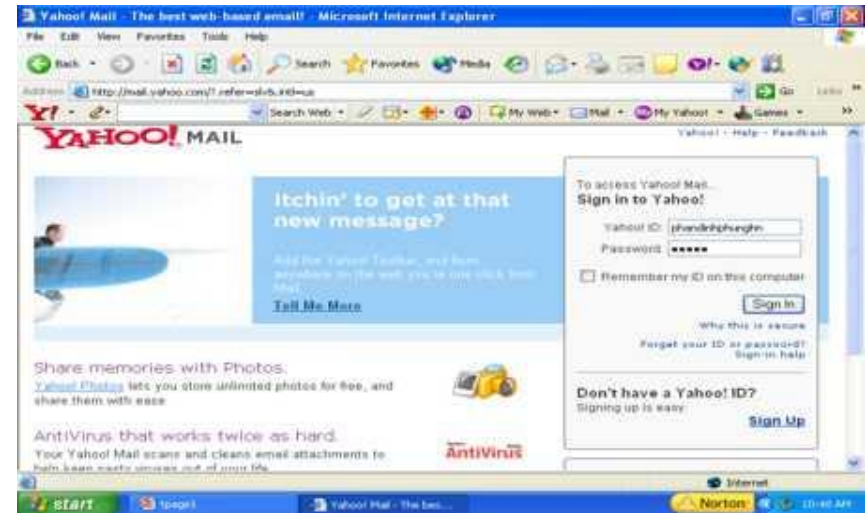
Quảng cáo năm 1913 của *Encyclopædia Britannica*.

- ◆ Cung cấp nguồn tài nguyên khổng lồ và vô tận

Cung cấp các dịch vụ, nhanh rẻ, tiết kiệm thời gian.



**Games**



**Thị trường điện tử**



# Cung cấp các dịch vụ, nhanh rẻ, tiết kiệm thời gian.



**Chat**



**Tìm kiếm thông tin**

# Đảm bảo một phương thức giao tiếp hoàn toàn mới



Mua hàng qua mạng



Điện thoại Internet

## 2. Dịch vụ World Wide Web

### a. Tổ chức thông tin:

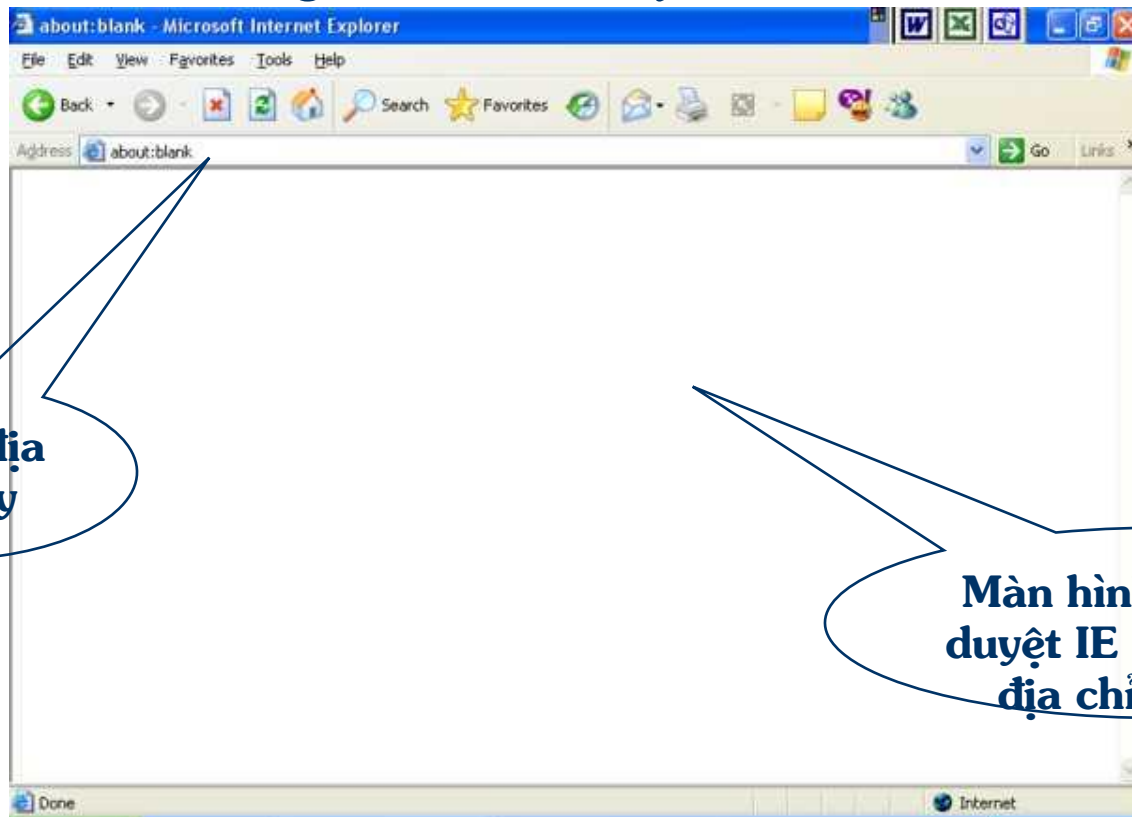
- ◆ Thông tin trên Internet được tổ chức dưới dạng siêu văn bản (là tổng thể của văn bản, hình ảnh, âm thanh, video... và có liên kết đến các siêu văn bản khác).
- ◆ Trang web là siêu văn bản đã được gán một địa chỉ cụ thể
- ◆ Để tìm kiếm các trang web và tài nguyên trên mạng Internet, ta sử dụng hệ thống WWW.
- ◆ Có 2 loại trang web: trang web động và trang web tĩnh
- ◆ Website là tập hợp của nhiều trang web trong hệ thống WWW, trong đó trang đầu tiên gọi là trang chủ (Home page)

## **b. Dịch vụ WWW**

- ◆ Là dịch vụ cho phép tìm kiếm các trang web, tài nguyên trên mạng.
- ◆ Được xây dựng dựa trên giao thức truyền tin siêu văn bản Http

## c. Trình duyệt Web

- ◆ Để truy cập vào các trang web ta phải dùng một chương trình đặc biệt gọi là trình duyệt web.



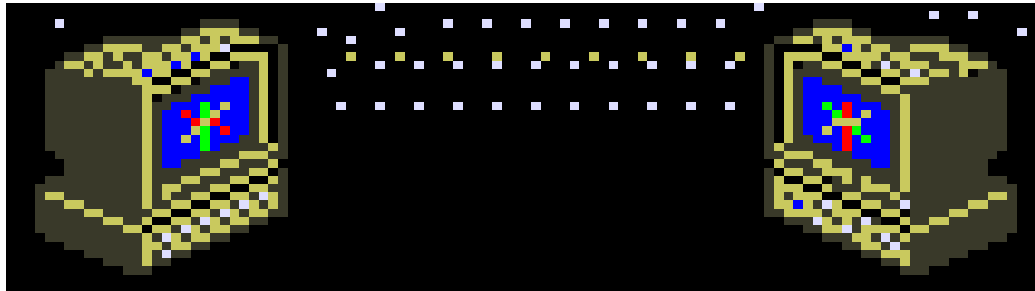
Nơi gõ địa  
chỉ truy  
cập

Màn hình của trình  
duyet IE khi ch- a có  
địa chỉ truy cập

## c. Trình duyệt Web

- ◆ Trình duyệt web: là chương trình giúp người dùng giao tiếp với hệ thống WWW: truy cập trang web, tương tác tới máy chủ trong hệ thống WWW và các tài nguyên khác của Internet
- ◆ Có nhiều loại trình duyệt khác nhau, thông dụng nhất là Internet Explorer, Netscape Navigator, Mozilla Firefox.

## 2. Dịch vụ thư điện tử (Email)



- Để nhận và gửi thư điện tử, người dùng cần đăng ký một hộp thư điện tử do nhà cung cấp dịch vụ thư cấp phát.
- Mỗi hộp thư được gắn với một địa chỉ có dạng:

**<Tên thuê bao>@<tên máy chủ của hộp thư>**

Ví dụ: **CDN\_LongBien@yahoo.com.vn**

## Đăng nhập Yahoo!



**Bạn đã được bảo vệ?**  
Hãy tạo con dấu đăng nhập.  
(Tại sao?)

Tên Truy Nhập  
Yahoo!

(ví dụ free2rhyme@yahoo.com)

Mật khẩu:

Tự động đăng nhập trong  
2 tuần nếu không đăng xuất. [Info](#)  
[Bỏ chọn nếu dùng chung máy tính]

Đăng Nhập

Nhập tên truy  
nhập

Mật khẩu đăng  
nhập

Đăng nhập vào  
hòm thư



# **HỆ SOẠN THẢO VĂN BẢN MICROSOFT WORD**

**Bài 9: Các thao tác soạn thảo, hiệu chỉnh  
và định dạng.**


**Bài 10: Làm việc với bảng**

# **Bài 9. Các thao tác soạn thảo, hiệu chỉnh, định dạng**

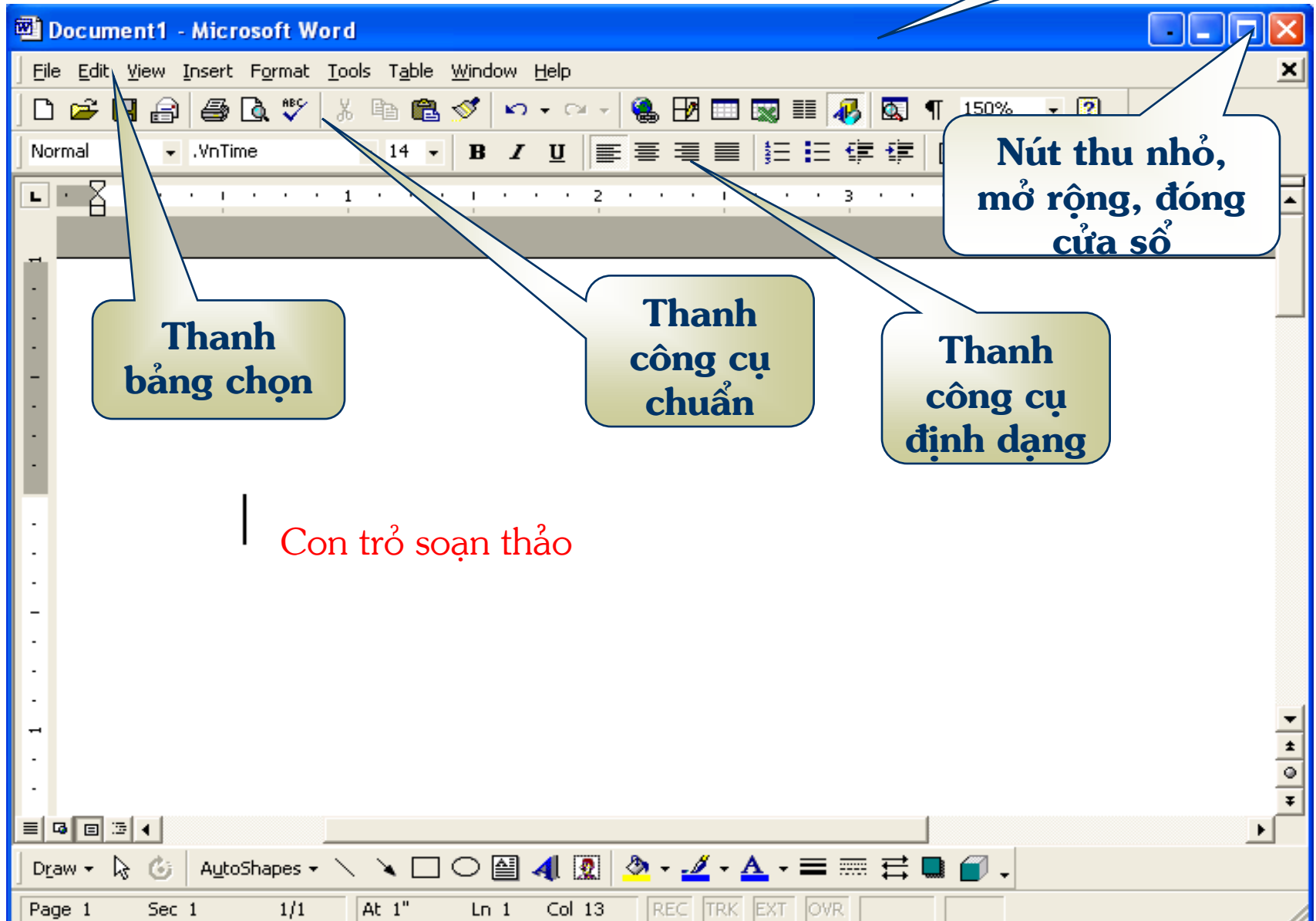
- ◆ Màn hình soạn thảo
- ◆ Các thao tác soạn thảo
- ◆ Các thao tác hiệu chỉnh
- ◆ Các thao tác định dạng

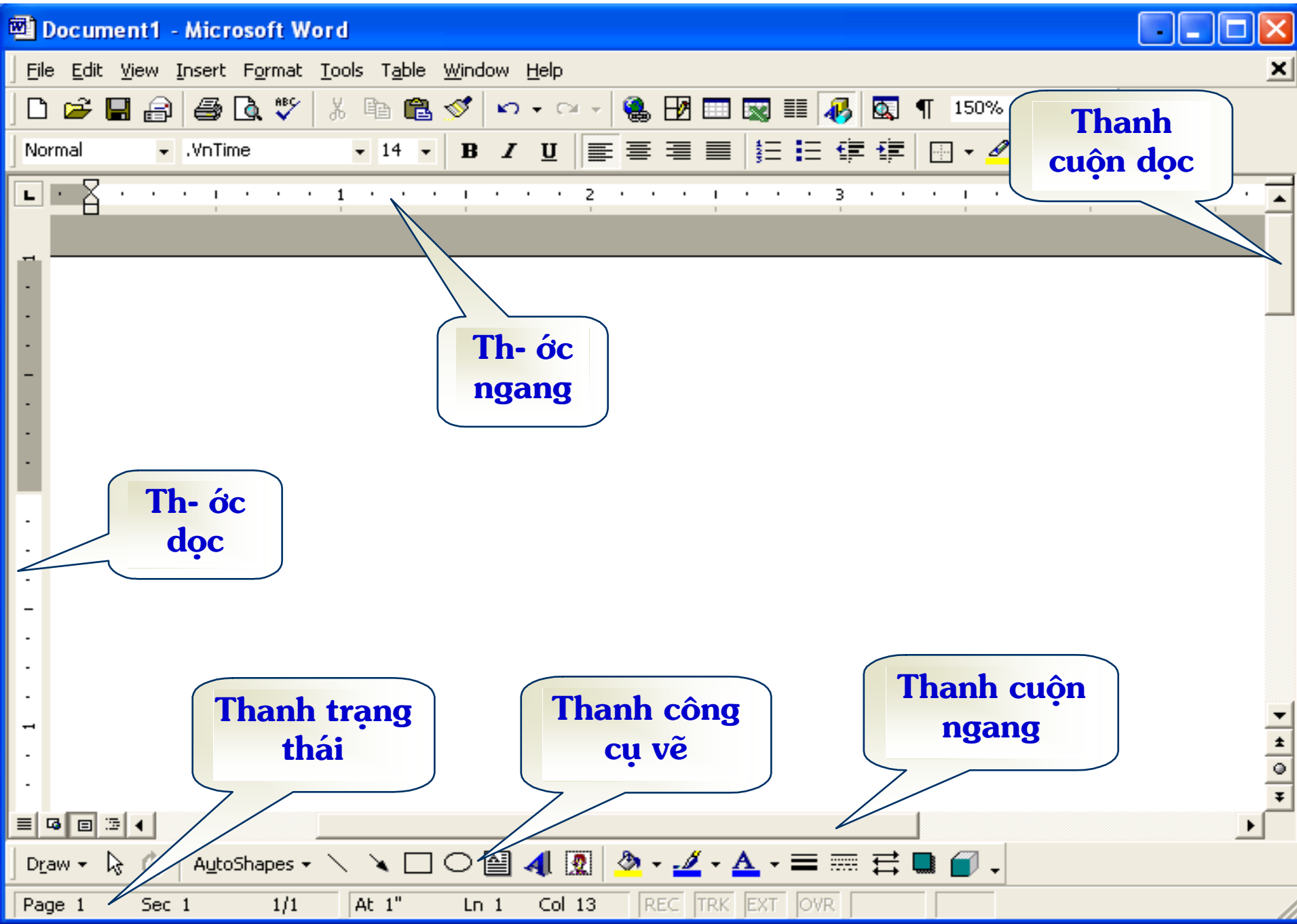
# 1. Màn hình soạn thảo.

## a. Khởi động Word

- ◆ Start -> Programs -> Microsoft Word
- ◆ Nháy đúp chuột vào biểu tượng Word trên màn hình nền 

## b. Màn hình soạn thảo





Thanh cuộn dọc

Th- ớc ngang

Th- ớc dọc

Thanh trạng thái

Thanh công cụ vẽ

Thanh cuộn ngang

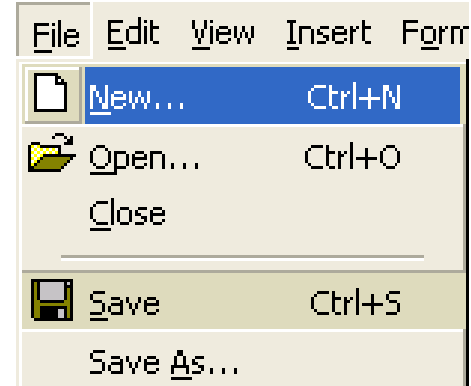
## c. Thoát khỏi Word

- ◆ Chọn File -> Exit (Alt + X)
- ◆ Ctrl + F4
- ◆ Chọn nút đóng Close

## 2. Các thao tác soạn thảo

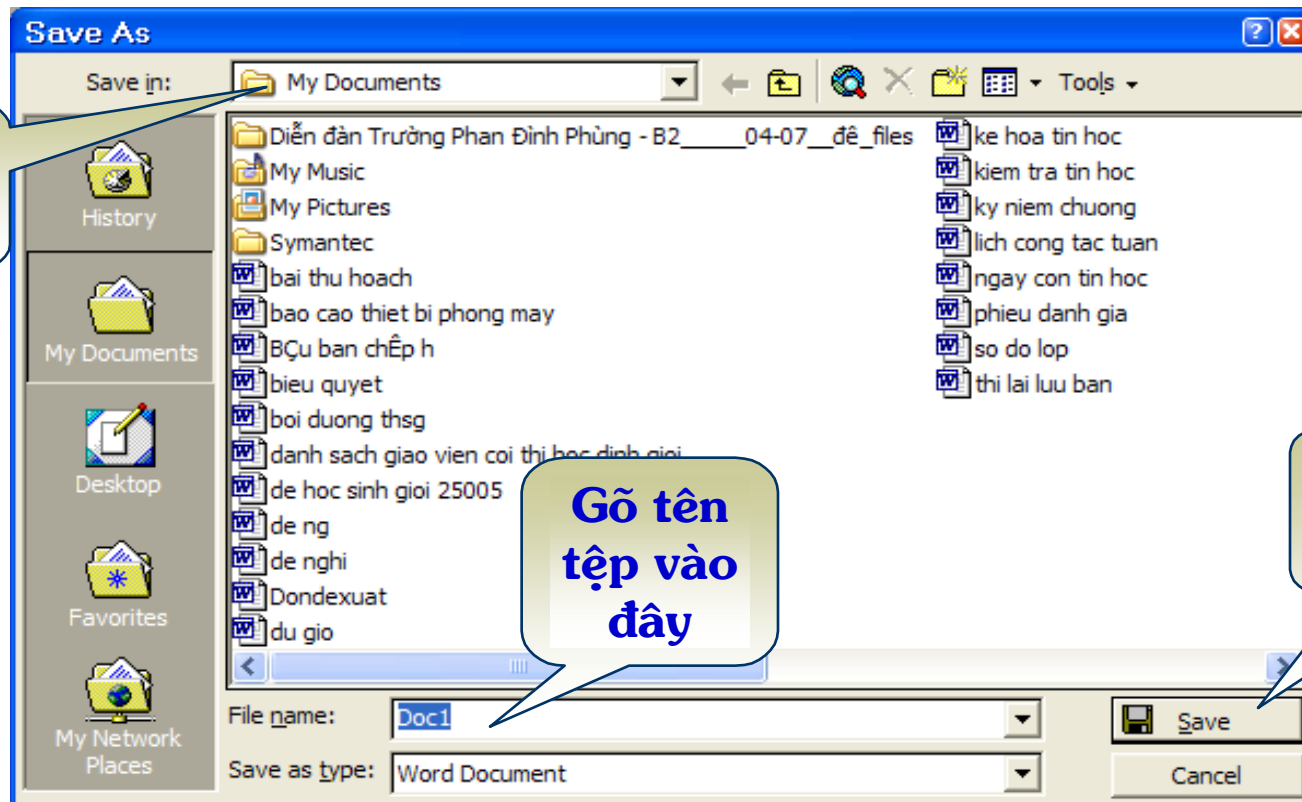
### a. Các thao tác với tệp văn bản

Mở tệp mới: Chọn **File** **New** (Ctrl+N)



Mở tệp sẵn có: Chọn **File** **Open** (Ctrl+O)

# Lưu tệp đang mở: Chọn **File** **Save** (Ctrl+S)



Th- mục  
l- u giữ tệp

Gõ tên  
tệp vào  
đây

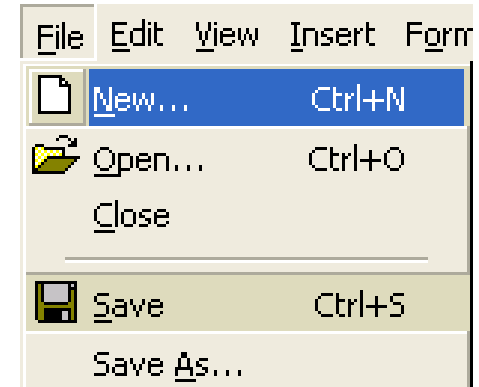
L- u tệp  
văn bản



## b. Các cách thực hiện một lệnh trong Word

1

**Chọn lệnh trong bảng chọn**



2

**Chọn nút lệnh trên thanh công cụ**



3

**Nhấn tổ hợp phím tắt. Ví dụ: Ctrl + N.**

## c. Con trỏ văn bản và con trỏ chuột

Cảnh đẹp | quê hương

Con trỏ  
chuột

Con trỏ  
văn bản

*Vịnh Hạ Long*



Các đảo trên vịnh Hạ Long chủ yếu là đảo đá vôi được hình thành cách đây trên năm trăm triệu năm. Ấn dấu trong những hòn đảo đá là hệ thống hang động vô cùng phong phú với măng, nhũ đá có quy mô, hình dáng, màu sắc huyền ảo... Một số hang động còn chứa đựng các dấu tích của người tiền sử Hạ Long là điểm hấp dẫn khách tham quan như: Đầu gỗ, Bồ Nâu, Sừng Sốt, Soi nhụ, Thiên Long, Mè cung, Tam cung...

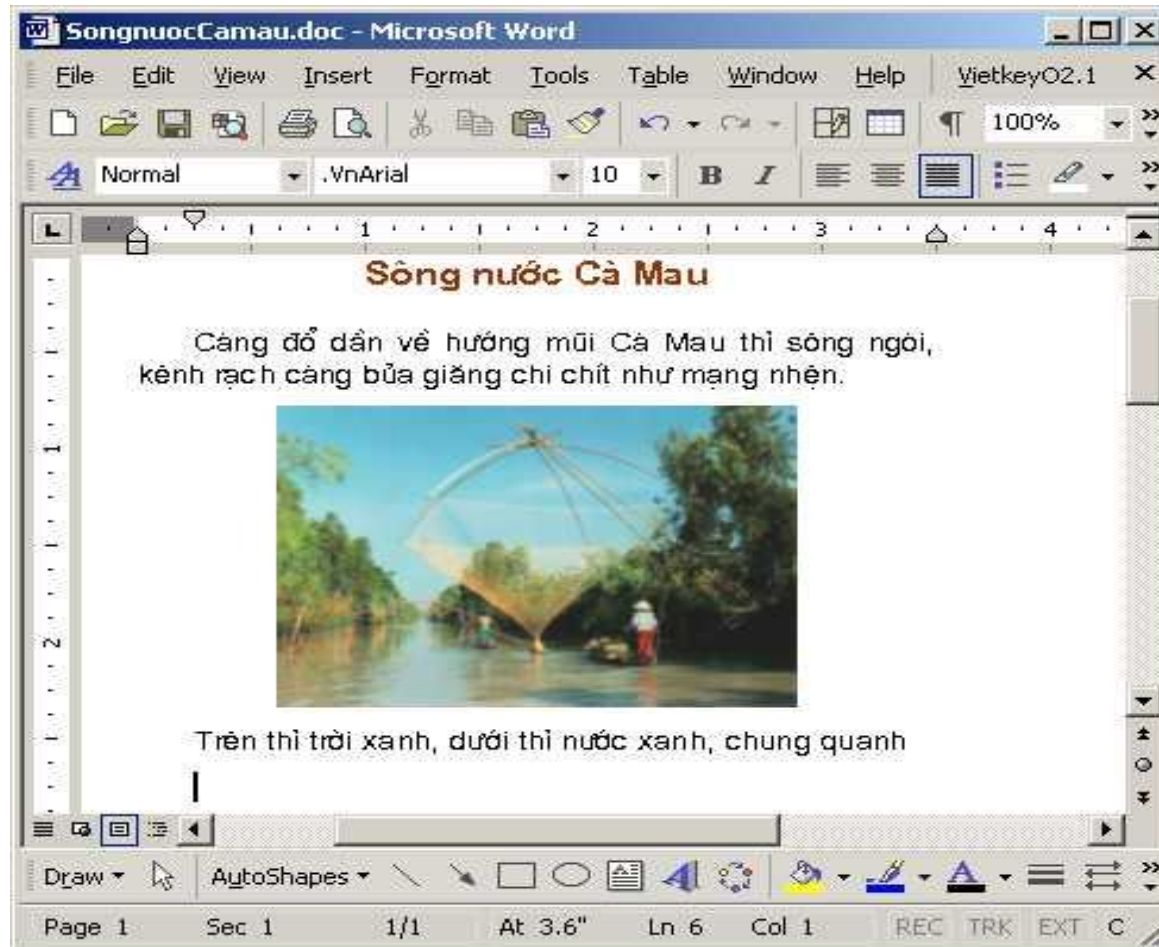
**Khi muốn di chuyển con trỏ văn bản:**

**Cách 1:** Nháy chuột tại vị trí mong muốn.

**Cách 2:** Dùng các phím **Home, End,...** trên bàn phím.

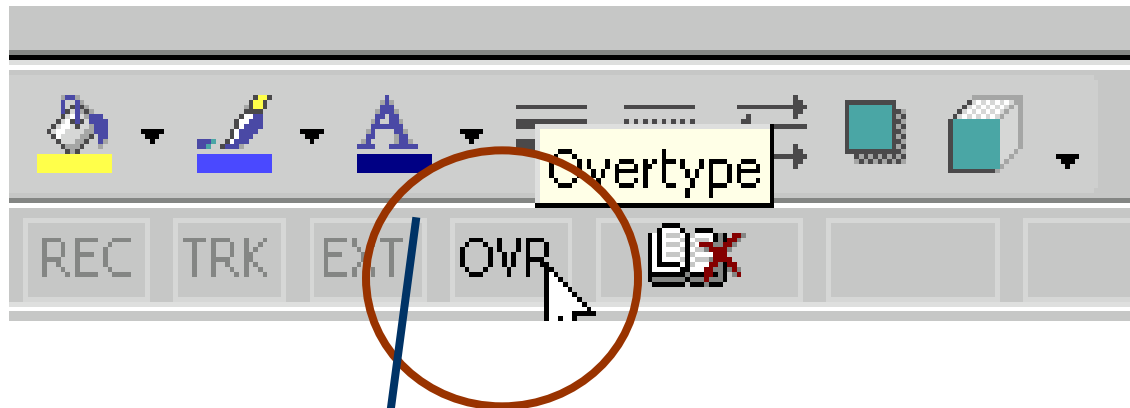
## d. Cách gõ văn bản

- Con trỏ văn bản ở vị trí cuối dòng sẽ tự động xuống dòng.
- Nhấn phím **Enter** để kết thúc một đoạn văn bản.



## Lưu ý hai chế độ gõ văn bản

- Chế độ chèn (**INSERT**): Ngâm định, nội dung văn bản gõ từ bàn phím sẽ đ- ọc chèn vào tr- ớc nội dung đã có từ vị trí con trỏ văn bản.
- Chế độ đè (**OVERTYPE**): Mỗi kí tự gõ vào từ bàn phím sẽ ghi đè, thay thế kí tự đã có tại vị trí con trỏ văn bản.



N  t OVR n  i r

*Nhấn phím INSERT hoặc nháy đúp chuột trên nút OVR để chuyển đổi chế độ chèn-đè hoặc bật/tắt trên thanh trạng thái.*

# 3. Các thao tác hiệu chỉnh

## a. Chọn văn bản

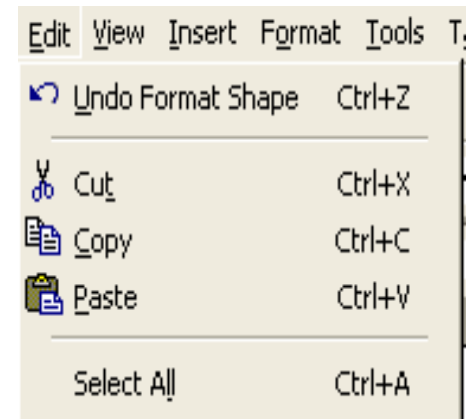
- Nhấn đồng thời phím shift và các phím Home, End.
- Kéo thả chuột
- Ctrl+A: nếu muốn chọn nhanh toàn bộ văn bản

## **b. Xoá văn bản**

- Chọn văn bản cần xoá.
- Dùng các phím xoá Delete hoặc Back space

## e. Sao chép hoặc di chuyển

- Lựa chọn văn bản
- chọn Edit -> Copy/Cut  
(Hoặc chọn nhanh biểu tượng trên thanh công cụ)
- Di chuyển con trỏ đến vị trí mới
- chọn Edit -> Paste



## 4. Các thao tác định dạng

*Định dạng văn bản là trình bày các phần văn bản theo những thuộc tính của văn bản nhằm mục đích cho văn bản đ- ọc rõ ràng và đẹp, nhấn mạnh những phần quan trọng, giúp ng- ời đ- ọc nắm bắt dễ hơn các nội dung chủ yếu của văn bản.*



## a. Định dạng ký tự

- ◆ Lựa chọn văn bản cần định dạng
- ◆ Format -> Font
- ◆ Thiết lập các thông số trong hộp thoại Font
- ◆ OK

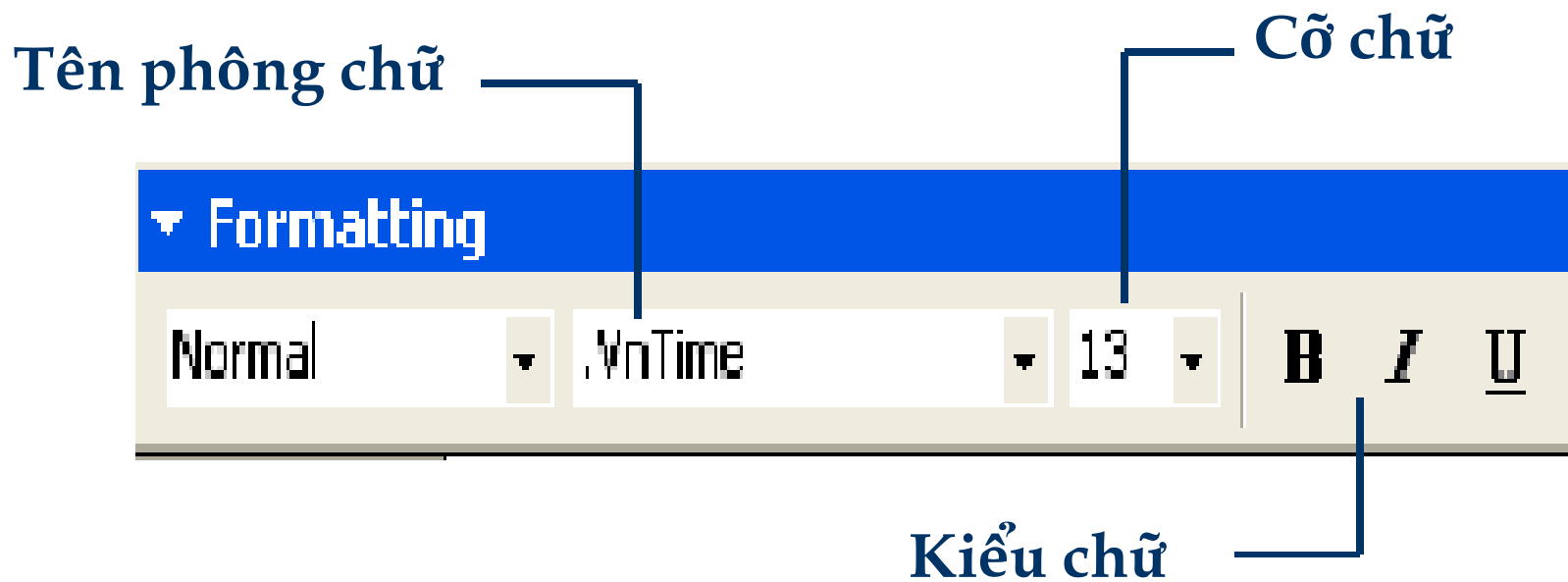
# Hộp thoại Font

The image shows a Windows 'Font' dialog box with several callout boxes in Vietnamese explaining its components:

- Chọn phông chữ** (Choose font): Points to the 'Font' list where '.VnTime' is selected.
- Kiểu chữ** (Font style): Points to the 'Font style' dropdown menu showing 'Regular'.
- Cỡ chữ** (Font size): Points to the 'Size' dropdown menu showing '13'.
- Màu chữ** (Font color): Points to the 'Font color' dropdown menu showing 'Automatic'.
- Gạch chân** (Underline): Points to the 'Underline style' dropdown menu showing '(none)'. This label is also connected to the 'Underline' checkbox in the 'Effects' section.
- Thiết lập chế độ ngầm định** (Set default): Points to the 'Default...' button at the bottom left.
- Cửa sổ hiển thị kết quả chọn** (Preview window): Points to the preview area at the bottom showing the text 'Đoàn TNCS Hồ Chí Minh'.

The dialog box itself has tabs for 'Font', 'Character Spacing', and 'Text Effects'. The 'Font' tab is active, showing a list of fonts, style options (Regular, Italic, Bold, Bold Italic), and size options (8, 9, 10, 13). The 'Effects' section includes checkboxes for Strikethrough, Double strikethrough, Superscript, Subscript, Shadow, Outline, Emboss, Engrave, Small caps, All caps, and Hidden. The preview area at the bottom displays the text 'Đoàn TNCS Hồ Chí Minh' in the selected font and size.

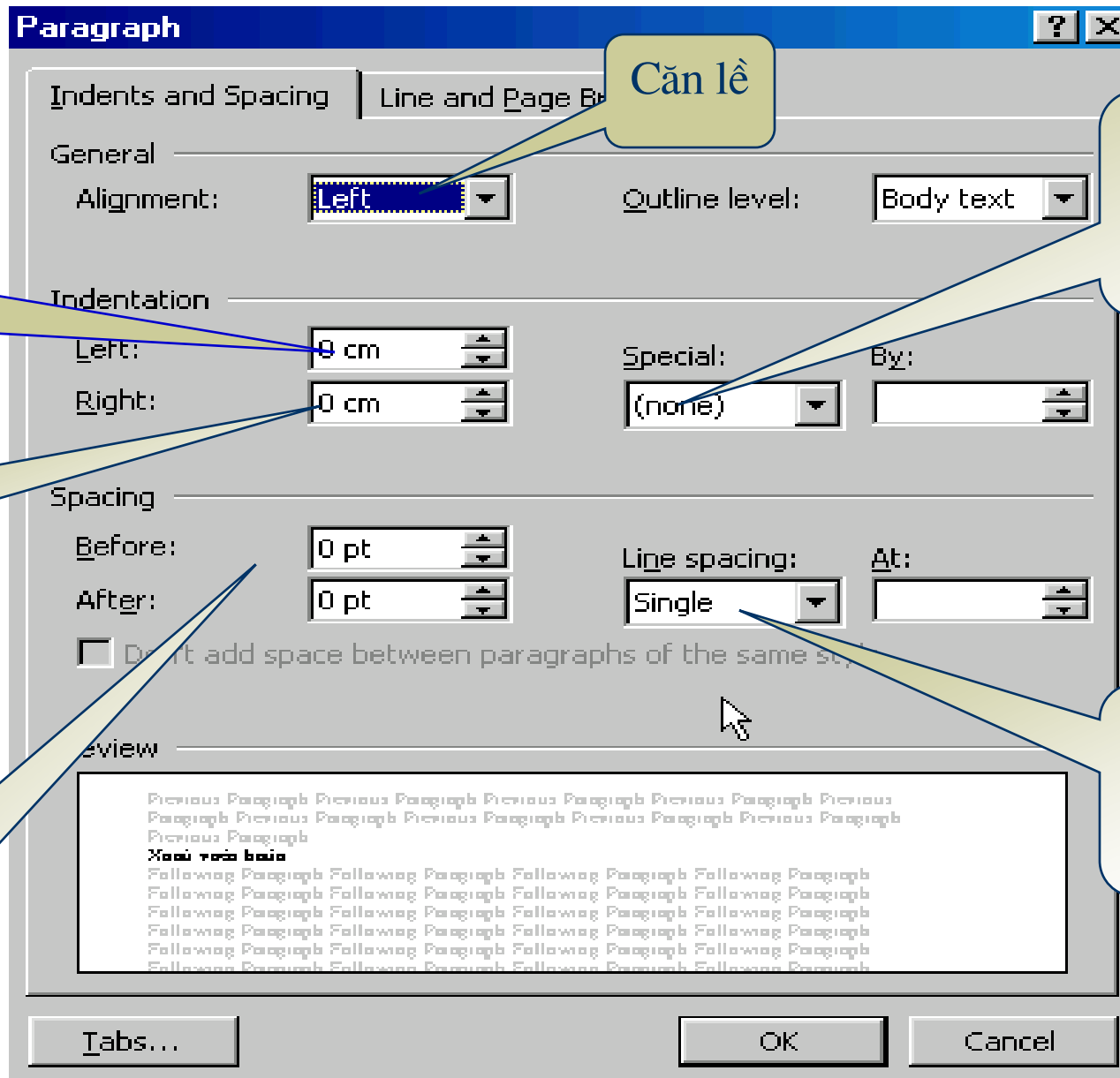
## Sử dụng các nút lệnh trên thanh công cụ định dạng



## b. Định dạng đoạn văn

- ◆ Chọn đoạn văn cần định dạng
- ◆ Format -> Paragraph
- ◆ Thiết lập các thông số trong hộp thoại Paragraph

# Hộp thoại Paragraph



Vị trí  
lề trái

Vị trí lề  
phải

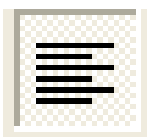
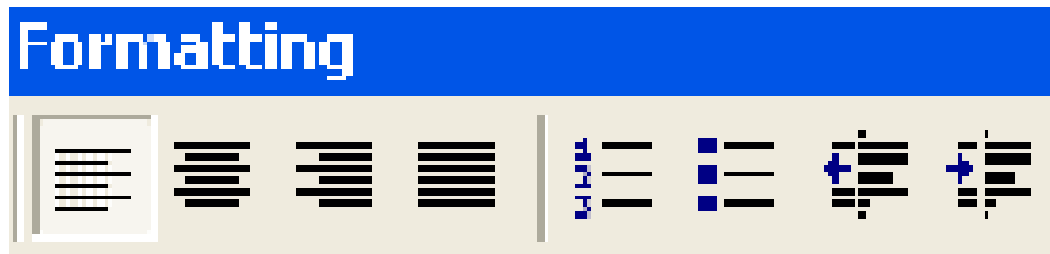
Khoảng  
cách giữa  
các đoạn  
văn

Căn lề

Định dạng  
dòng đầu  
tiên

Khoảng  
cách giữa  
các dòng

# Sử dụng các nút lệnh trên thanh công cụ định dạng



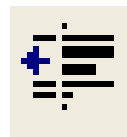
*Căn thẳng lề trái*



*Căn thẳng hai lề*



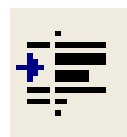
*Căn giữa*



*Giảm lề một  
khoảng nhất định*

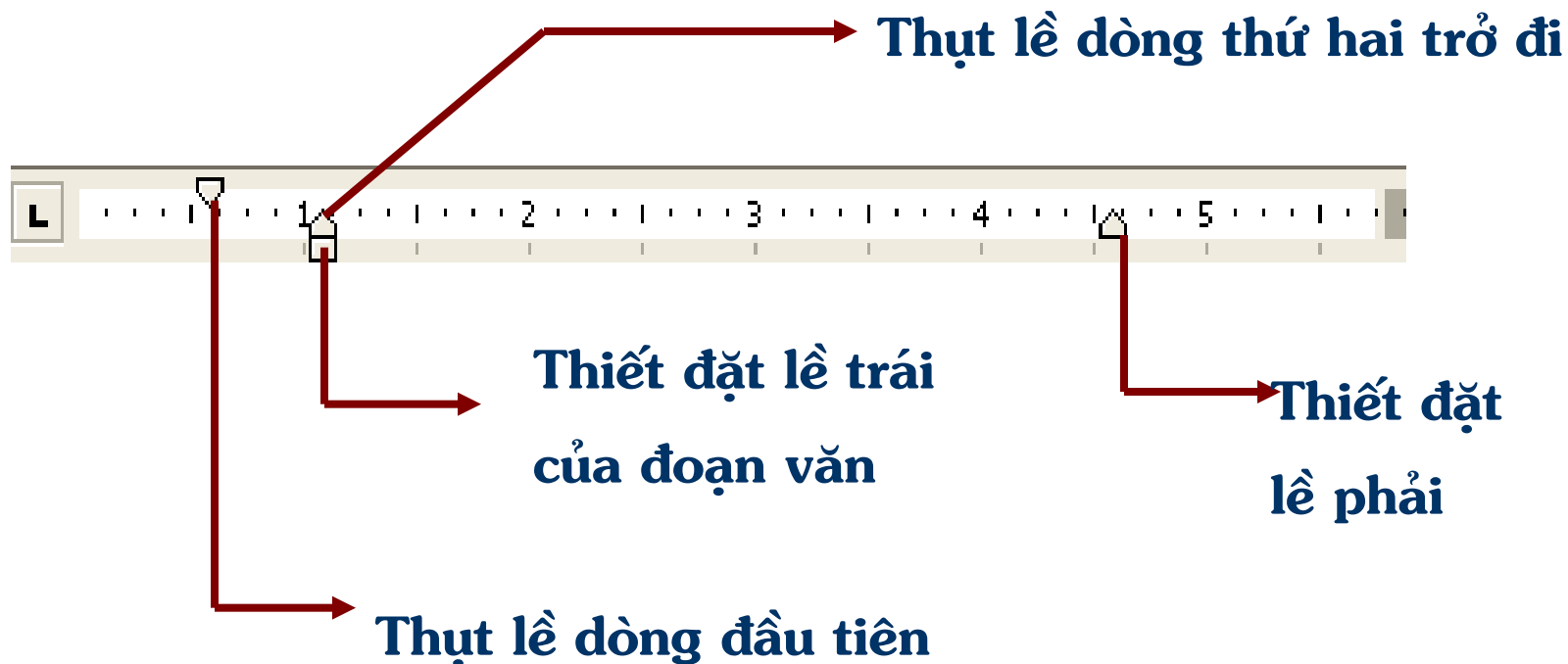


*Căn thẳng lề phải*



*Tăng lề một  
khoảng nhất định*

✎ Ngoài ra có thể dùng thước ngang để điều chỉnh một số thuộc tính của lề bằng cách đ- a con trỏ chuột lên con tr- ợt t- ơng ứng trên th- ớc và kéo thả đến vị trí cần thiết.

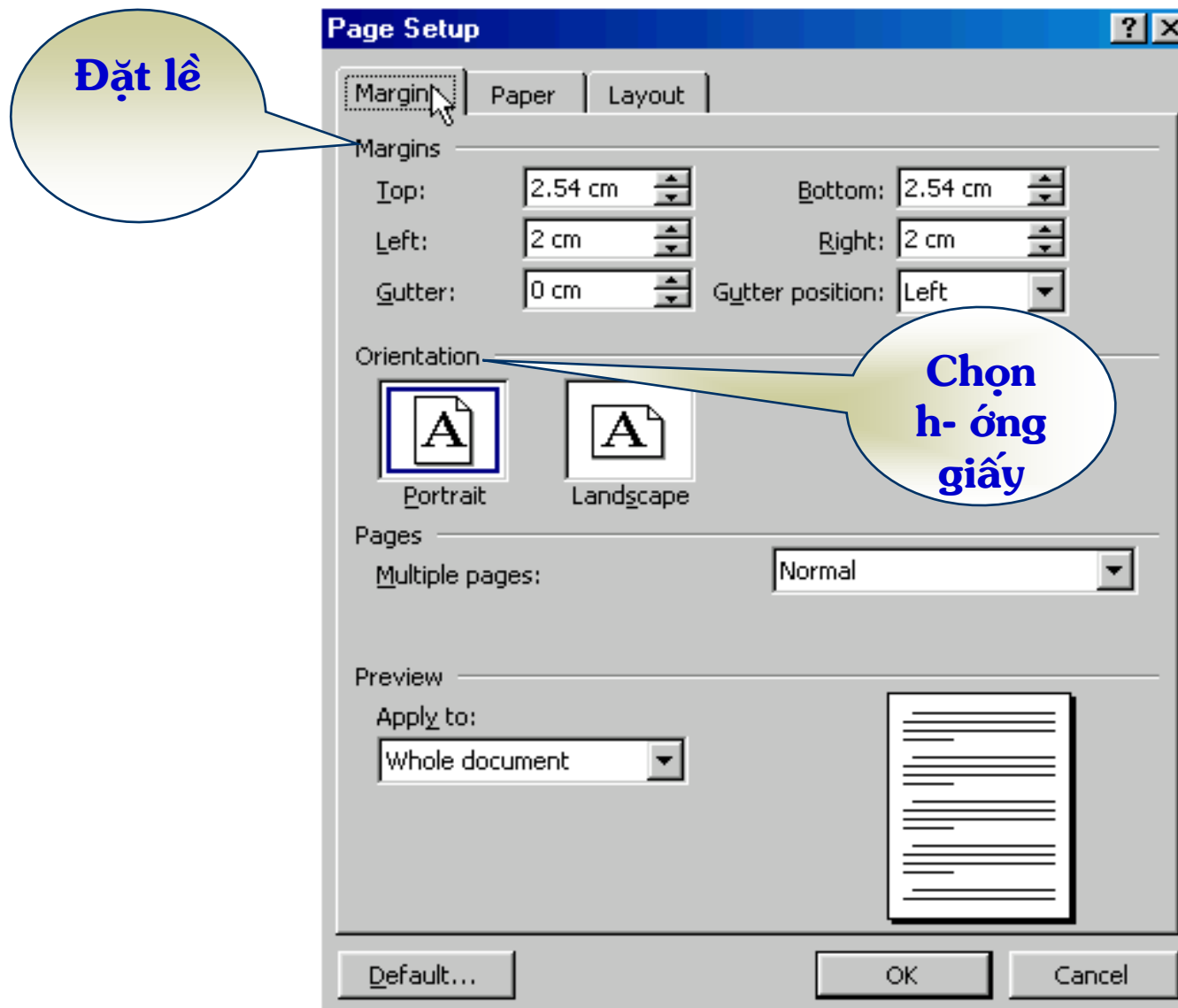


## c. Định dạng trang in

- ◆ File -> Page Setup
- ◆ Thiết lập các thông số trong hộp thoại page setup



# Hộp thoại Page Setup



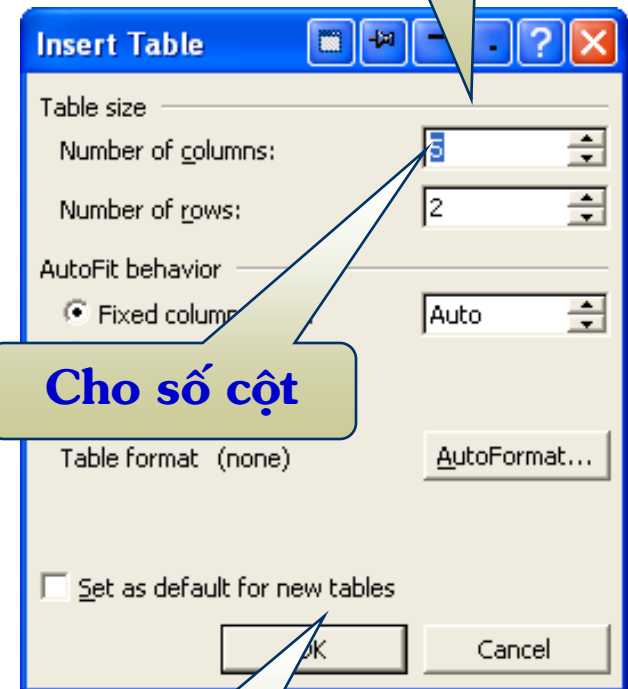
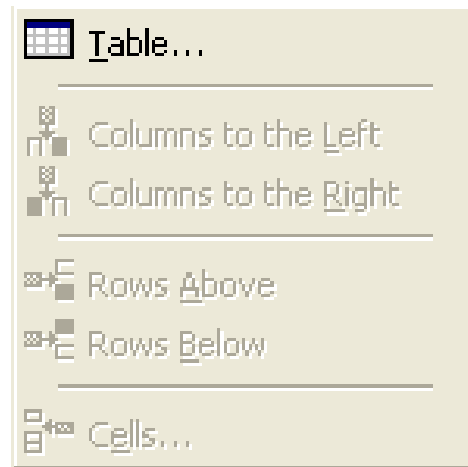
# Bài 10. Làm việc với bảng

- ◆ Tạo bảng
- ◆ Các thao tác với bảng

# 1. Tạo bảng

## a. Tạo

Cách 1: Chọn **Table** **Insert** **Table** □



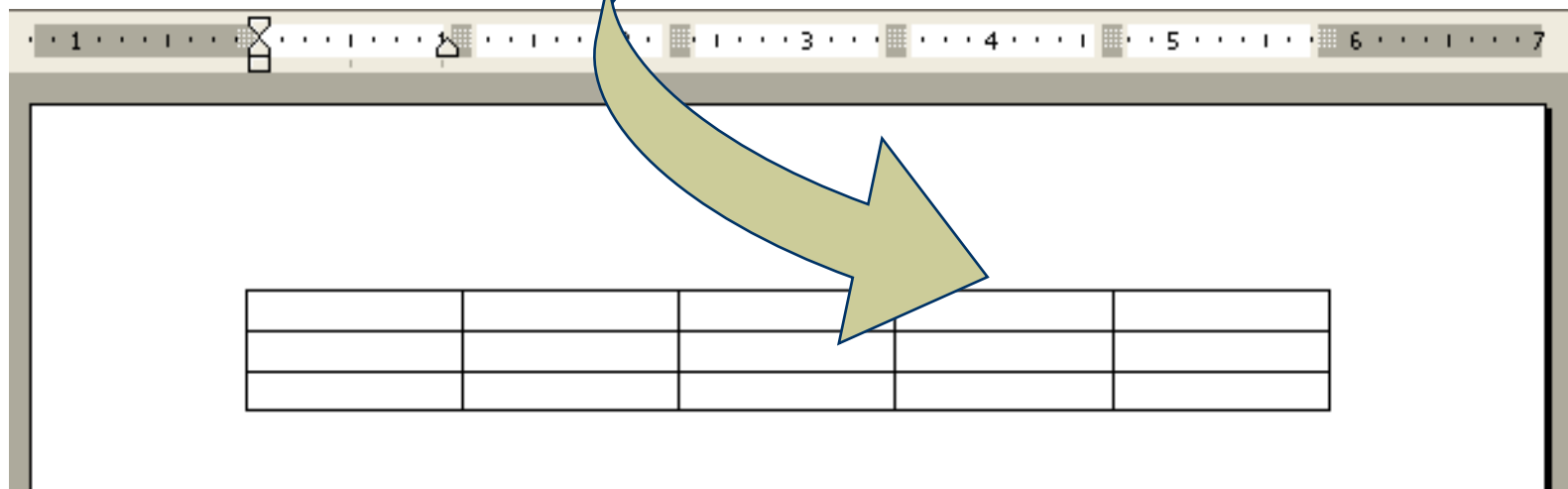
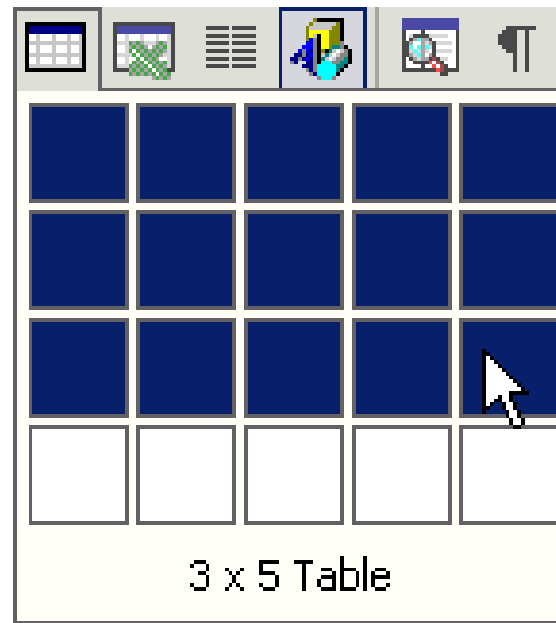
Cho số hàng

Cho số cột

Nháy OK

**Cách 2:** Nháy chọn nút lệnh  trên thanh công cụ.

*Kéo thả chuột xuống dưới và sang phải để chọn số hàng, cột cho bảng.*



## *b. Chọn thành phần của bảng*

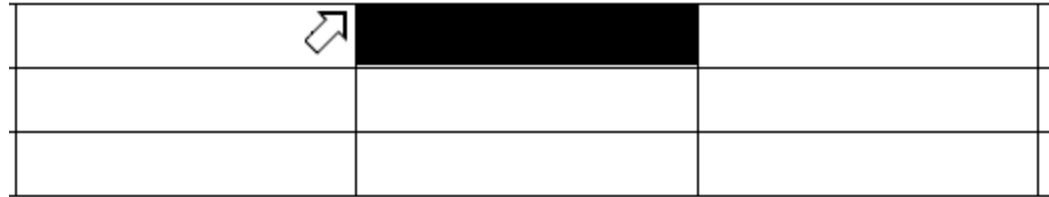
### **Cách 1: Dùng lệnh Table Select**

The image shows a screenshot of a spreadsheet application's menu system. The 'Table' menu is open, and the 'Select' option is highlighted in blue. A submenu is visible, showing options: 'Table', 'Column', 'Row', and 'Cell'. A callout box with the text 'Chọn ô' (Select cell) points to the 'Cell' option. Below the menu is a 3x5 grid representing a table. The cell at row 1, column 3 is shaded black, indicating it is selected.

|  |  |  |  |  |
|--|--|--|--|--|
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

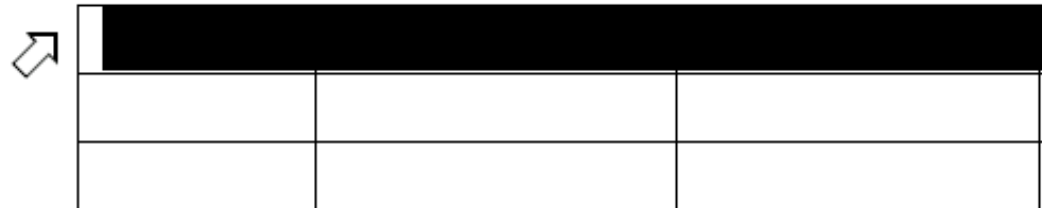
## Cách 2: Dùng chuột

- Chọn một ô, nháy chuột tại cạnh trái của ô đó.
- Chọn một hàng, nháy chuột bên trái của hàng đó.
- Chọn một cột, nháy chuột ở đ- ờng biên trên của ô trên cùng của cột đó.



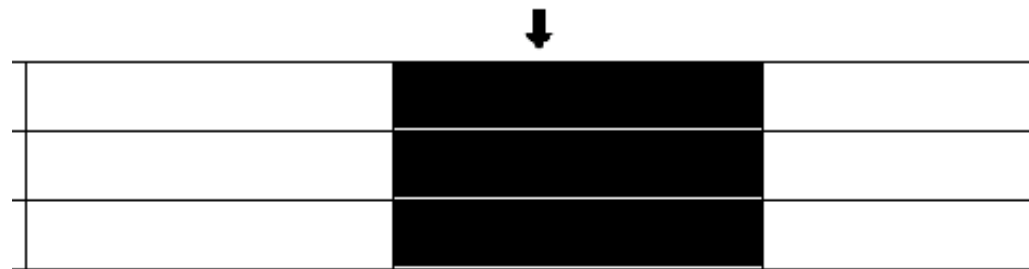
|  |  |  |  |
|--|--|--|--|
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

A 3x4 grid of cells. The second cell in the first row is shaded black. A mouse cursor icon is positioned at the left edge of this cell.



|  |  |  |  |
|--|--|--|--|
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

A 3x4 grid of cells. The first row is shaded black. A mouse cursor icon is positioned at the left edge of the first row.

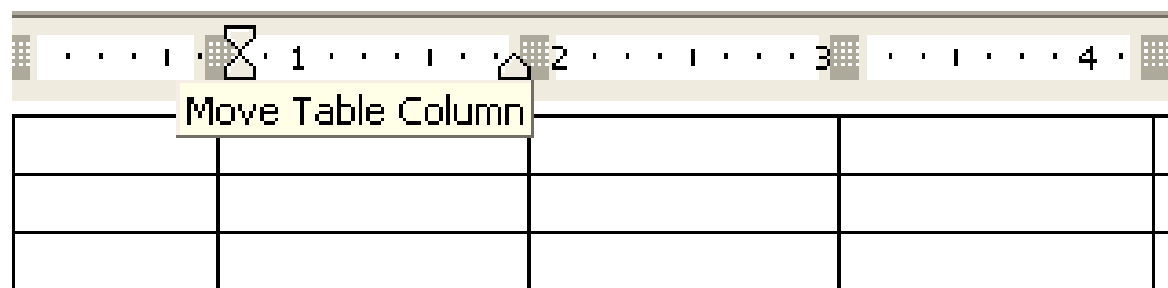


|  |  |  |  |
|--|--|--|--|
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

A 3x4 grid of cells. The second column is shaded black. A mouse cursor icon is positioned at the top edge of the second column.

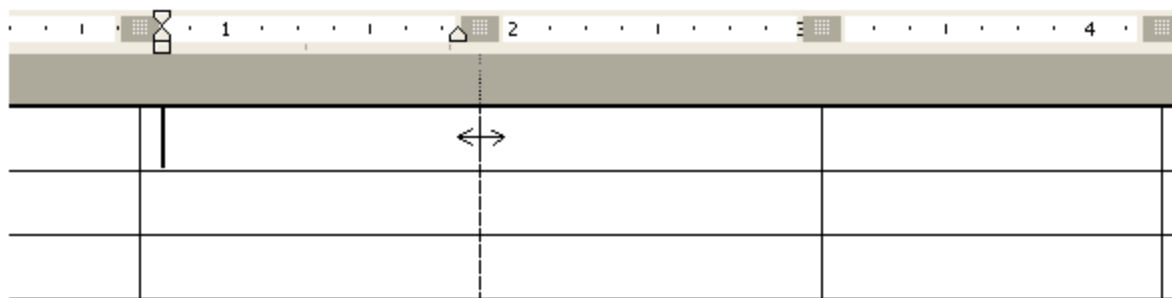
### c. Thay đổi kích thước của cột hay hàng

Do chiều rộng cột chọn tự động (Auto) nên chiều ngang trang văn bản sẽ được chia đều cho các cột.



The image shows a table with four columns and three rows. A tooltip labeled "Move Table Column" is positioned over the first column. Above the table, a horizontal bar contains four column width indicators, each labeled with the number "1", "2", "3", and "4" respectively. The first indicator is highlighted with a mouse cursor.

|  |  |  |  |
|--|--|--|--|
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |



The image shows the same table as above, but with a horizontal double-headed arrow (↔) positioned between the first and second columns, indicating the process of adjusting the column width. A vertical dashed line is also visible between the first and second columns.

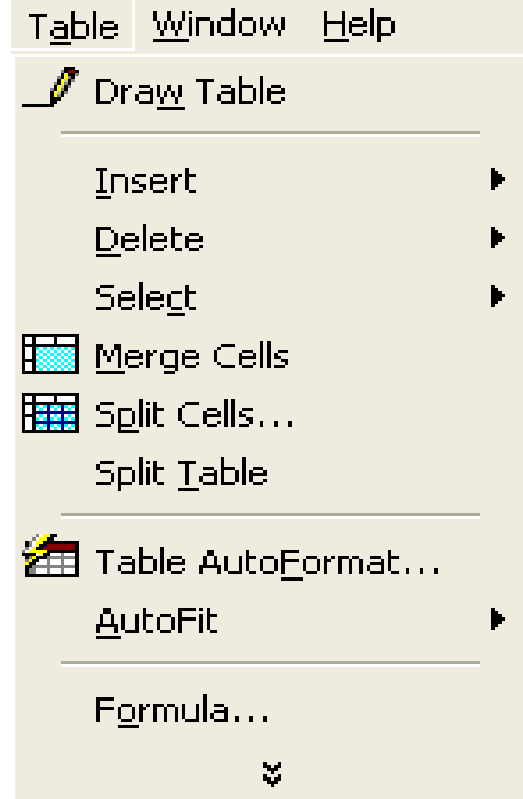
|  |  |  |  |
|--|--|--|--|
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## 2. Các thao tác với bảng

### a. Chèn thêm, xoá ô hàng hoặc cột

**1** Chọn ô hay cột, hàng tương ứng cần chèn hoặc xoá.

**2** Dùng lệnh **Table Insert** (chèn) hoặc **Table Delete** (xoá).





# b. Tách một ô thành nhiều ô

**1** Chọn ô cần tách

|  |  |
|--|--|
|  |  |
|  |  |
|  |  |
|  |  |

**2** Sử dụng lệnh **Table** **Split Cells**

Số cột

Split Cells

Number of columns: 3

Number of rows: 3

Merge cells before split

OK Cancel

Hoặc chọn biểu tượng  trên thanh **Tables and Borders**.

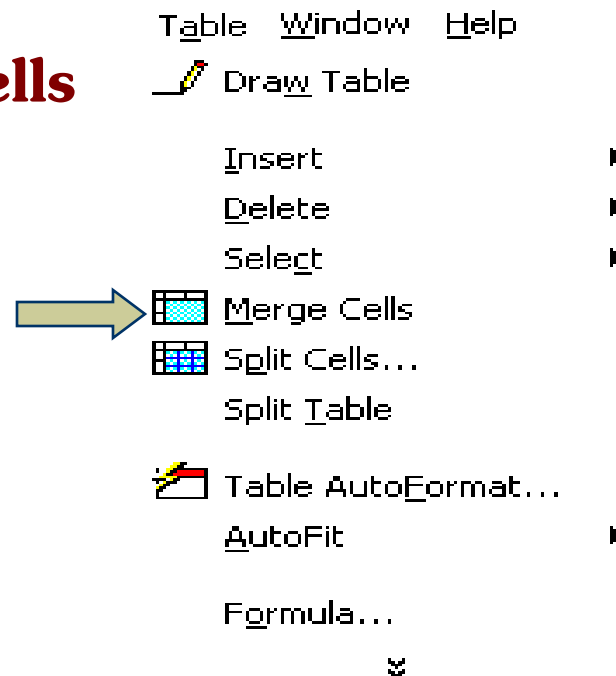
|  |  |  |  |
|--|--|--|--|
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## c. Gộp nhiều ô thành một ô

**1** Chọn các ô cần gộp

|  |  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|--|
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**2** Sử dụng lệnh **Table Merge Cells**

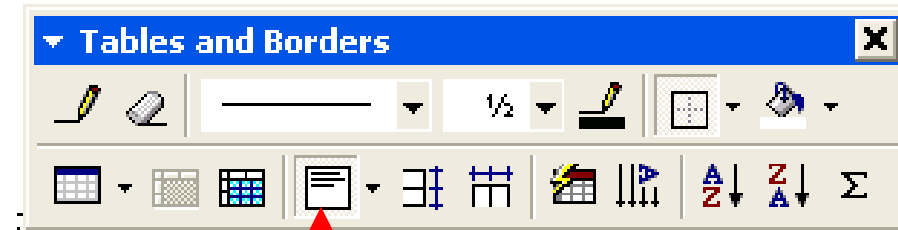


Hoặc chọn biểu tượng  
trên thanh **Tables and Borders**.



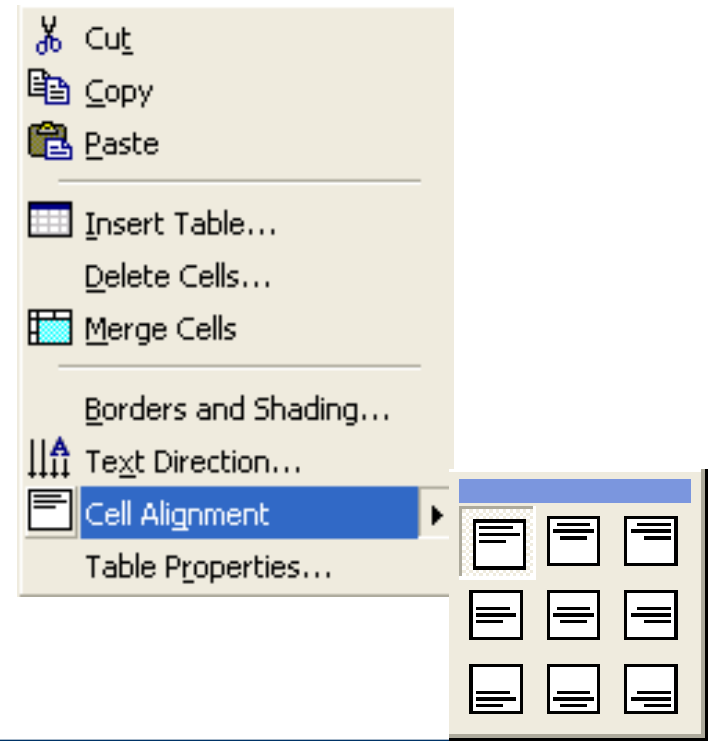
## d. Định dạng văn bản trong ô

### Cách 1: Chọn biểu tượng



### Cách 2:

- Nháy nút phải chuột.
- Chọn **Cell Alignment**.



## Chương V:

# BẢNG TÍNH EXCEL

**Bài 11: Giới thiệu về Excel**

**Bài 12: Làm việc với bảng tính**

**Bài 13: Tính toán dữ liệu trên bảng tính**

**Bài 14: Các hàm cơ sở dữ liệu**

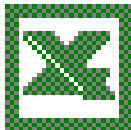
**Bài 15: Làm việc với WorkSheet**

# Bài 11: Giới thiệu về Excel


- ◆ Khởi động và thoát
- ◆ Mở một bảng tính mới
- ◆ Cửa sổ Excel
- ◆ Hộp hội thoại
- ◆ Nhập dữ liệu
- ◆ Sắp xếp dữ liệu đơn giản
- ◆ Thêm dòng và cột
- ◆ Xoá dòng và cột,
- ◆ Hiệu chỉnh kích thước ô dòng
- ◆ Lưu bảng tính
- ◆ Mở một file có sẵn
- ◆ Tìm kiếm file
- ◆ Đóng file

# 1. Khởi động và thoát khỏi Excel


## a. Khởi động

- ◆ Start -> Programs -> Microsoft Excel
- ◆ Nhấp đúp vào biểu tượng  trên màn hình nền

## b. Thoát khỏi Excel

- ◆ *Thực hiện một trong các cách sau:*
  - Chọn File -> Exit
  - Nhấn tổ hợp phím Alt + X, hoặc Ctrl + F4
  - Chọn nút Close  trên thanh tiêu đề

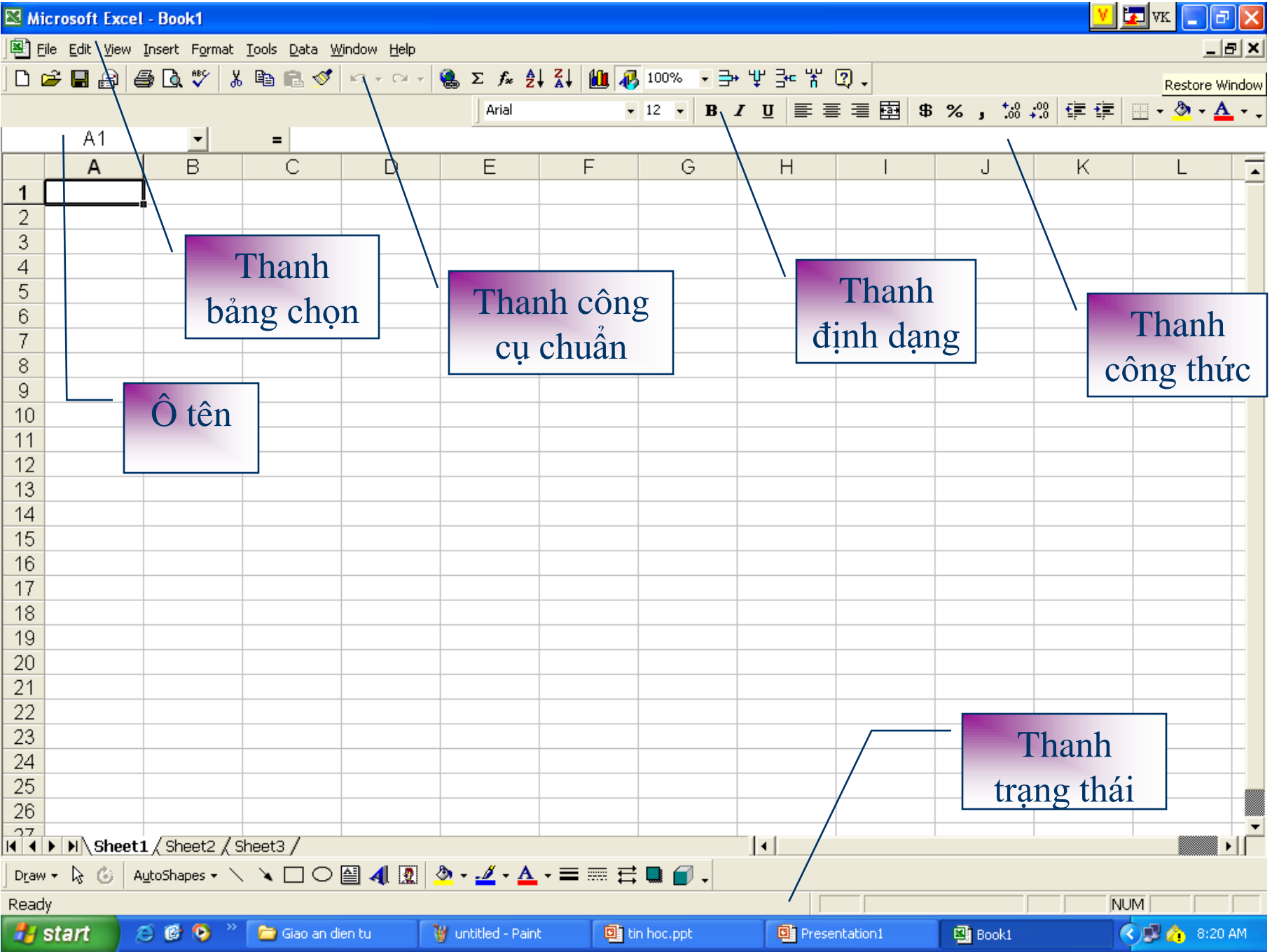
## 2. Mở một bảng tính mới

- ◆ Để mở một bảng tính mới ta thực hiện một trong các cách sau:
  - Chọn **File** -> **New** (Ctrl +N) -> OK
  - Chọn chức năng new  trên thanh công cụ



# 3. Cửa sổ Excel

- ◆ Sau khi khởi động vào Excel hoặc mở một file mới ta thấy cửa sổ làm việc của Excel có dạng như sau:



Thanh  
bảng chọn

Thanh công  
cụ chuẩn

Thanh  
định dạng

Thanh  
công thức

Ô tên

Thanh  
trạng thái

|    | A | B            | C                | D               | E | F | G | H | I | J | K |
|----|---|--------------|------------------|-----------------|---|---|---|---|---|---|---|
| 1  |   |              |                  |                 |   |   |   |   |   |   |   |
| 2  |   | <b>Mã SV</b> | <b>Họ và tên</b> | <b>Năm sinh</b> |   |   |   |   |   |   |   |
| 3  |   | T001         | Nguyễn Văn An    | 1987            |   |   |   |   |   |   |   |
| 4  |   | T002         | Lê Tuyết Nhi     | 1989            |   |   |   |   |   |   |   |
| 5  |   | T003         | Trần Bình Minh   | 1990            |   |   |   |   |   |   |   |
| 6  |   |              |                  |                 |   |   |   |   |   |   |   |
| 7  |   |              |                  |                 |   |   |   |   |   |   |   |
| 8  |   |              |                  |                 |   |   |   |   |   |   |   |
| 9  |   |              |                  |                 |   |   |   |   |   |   |   |
| 10 |   |              |                  |                 |   |   |   |   |   |   |   |
| 11 |   |              |                  |                 |   |   |   |   |   |   |   |
| 12 |   |              |                  |                 |   |   |   |   |   |   |   |
| 13 |   |              |                  |                 |   |   |   |   |   |   |   |
| 14 |   |              |                  |                 |   |   |   |   |   |   |   |
| 15 |   |              |                  |                 |   |   |   |   |   |   |   |
| 16 |   |              |                  |                 |   |   |   |   |   |   |   |
| 17 |   |              |                  |                 |   |   |   |   |   |   |   |
| 18 |   |              |                  |                 |   |   |   |   |   |   |   |
| 19 |   |              |                  |                 |   |   |   |   |   |   |   |
| 20 |   |              |                  |                 |   |   |   |   |   |   |   |
| 21 |   |              |                  |                 |   |   |   |   |   |   |   |
| 22 |   |              |                  |                 |   |   |   |   |   |   |   |
| 23 |   |              |                  |                 |   |   |   |   |   |   |   |
| 24 |   |              |                  |                 |   |   |   |   |   |   |   |
| 25 |   |              |                  |                 |   |   |   |   |   |   |   |
| 26 |   |              |                  |                 |   |   |   |   |   |   |   |
| 27 |   |              |                  |                 |   |   |   |   |   |   |   |

Tiêu đề cột

Vùng chọn

Tiêu đề dòng

Ô

Tab điều khiển Sheet

## 4. Nhập dữ liệu

### a. Các kiểu dữ liệu trong Excel

- ◆ **Kiểu dữ liệu số:** là tập hợp các kí tự số, có thể chứa dấu phẩy (,), dấu chấm (.), và phần trăm (%).
- ◆ **Kiểu chuỗi:** có thể chứa các ký tự số và chữ cái. Riêng trường hợp đối với ô kiểu chuỗi ở dạng số (chuỗi số) thì kí tự đầu tiên là dấu nháy đơn (')

## a. Các kiểu dữ liệu

- ◆ **Kiểu ngày, giờ:** Dữ liệu có dạng ngày, tháng, năm, giờ, phút, giây. Khi nhập liệu cho kiểu ngày giờ, chú ý định dạng kiểu ngày tháng trong hệ thống máy tính (trong mục Regional Setting).
- ◆ **Kiểu công thức:** có thể chứa các biểu thức, các hàm đã được định nghĩa sẵn để tính toán. Công thức trong bảng tính phải bắt đầu bởi dấu bằng (=)

## b. **Chỉnh sửa hệ thống trước khi nhập liệu**

- ◆ Để hiệu chỉnh các thông tin trong hệ thống ta thực hiện các bước sau:
  - Mở cửa sổ **Regional and Language Option**
  - Trong cửa sổ này, chọn tab **Regional Option**, sau đó chọn **Customize**.
  - Trong cs xuất hiện, chọn tab **Date**
  - Trong mục **Short date format** nhập định dạng phù hợp. ví dụ: d/m/yy



## c. Dữ liệu nhập không hợp lệ

- ◆ Trong quá trình nhập liệu, đôi khi Excel không có khả năng nhận biết ra dữ liệu nhập vào và do đó không có khả năng hiển thị
- ◆ Khi đó, Excel sẽ hiển thị các ký hiệu để phản hồi đến cho người dùng.
- ◆ Bảng các ký hiệu lỗi thường gặp:

| Ký hiệu | Mô tả                                                                                                                                            |
|---------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| #       | Hiển thị khi dữ liệu kiểu số, ngày chứa trong ô không có đủ kích thước để hiển thị dữ liệu. Trong trường hợp này, ta cần tăng độ rộng của cột    |
| #VALUE  | Ký hiệu này sẽ hiển thị khi kiểu của các tham số trong hàm, công thức không hợp lệ.<br>Ví dụ: A1="ABC", A2=A1*10                                 |
| #Name?  | Ký hiệu này sẽ hiển thị khi hàm, công thức mà Excel không hiểu hay tham chiếu địa chỉ ô mà Excel không biết. Ví dụ: B1=ABC (A1:A4); B2=Sum(A1A4) |
| #REF!   | Ký hiệu này sẽ hiển thị khi có sự tham chiếu không hợp lệ do trong ô công thức có tham chiếu đến 1 ô đã bị xoá                                   |
| #DIV/0  | Ký hiệu này sẽ hiển thị khi có một phép tính chia cho 0                                                                                          |
| #N/A    | Ký hiệu này sẽ xuất hiện khi giá trị không tìm thấy trong hàm hay công thức                                                                      |



# 5. Sắp xếp dữ liệu đơn giản

- ◆ Việc sắp xếp dữ liệu sẽ làm cho thông tin dễ quan sát hơn
- ◆ Để sắp xếp dữ liệu ta thực hiện như sau:
  - Chọn vùng dữ liệu cần sắp xếp
  - Chọn chức năng sắp xếp trên thanh công cụ
    -  Sắp tăng dần
    -  Sắp giảm dần

# 6. Chọn đối tượng

## a. Chọn ô

- ◆ Nhấp chuột tại ô cần chọn
- ◆ Di chuyển con trỏ đến ô cần chọn

## b. Chọn khối

- ◆ Di chuyển con trỏ đến ô bắt đầu của vùng, nhấn và rê chuột để tạo thành vùng cần chọn

## c. Chọn dòng, cột, vùng

### ➤ Chọn dòng

- Nhấp chuột vào tiêu đề của dòng cần chọn

### ➤ Chọn cột

- Nhấp chuột vào tiêu đề của cột cần chọn

### ➤ Chọn nhiều vùng nằm rời rạc

- Chọn một vùng đầu tiên
- Nhấn giữ phím Ctrl, tiếp tục thực hiện thao tác chọn các vùng khác.

# 7. Thêm dòng và cột

## a. Thêm dòng

- Chọn dòng muốn chèn thêm vào một dòng mới
- Chọn từ đơn **Insert/Rows**

## b. Thêm cột

- Chọn cột muốn chèn thêm vào một cột mới
- Chọn từ đơn **Insert/Column**

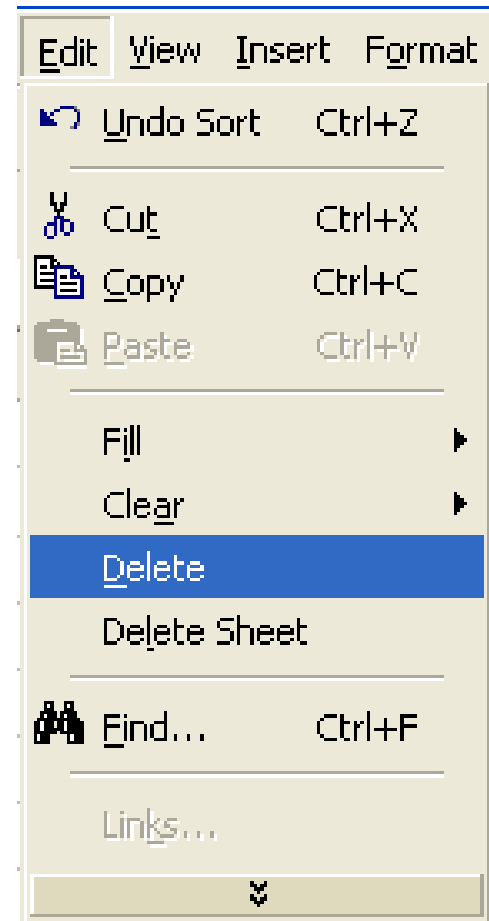
# 8. Xoá dòng và cột

## a. Xoá dòng

- Chọn dòng cần xoá
- Chọn từ đơn **Edit/Delete**

## b. Xoá cột

- Chọn cột muốn xoá
- Chọn từ đơn **Edit/Delete**



# 9. Hiệu chỉnh kích thước của dòng, cột

## a. Thay đổi bằng thao tác trực tiếp

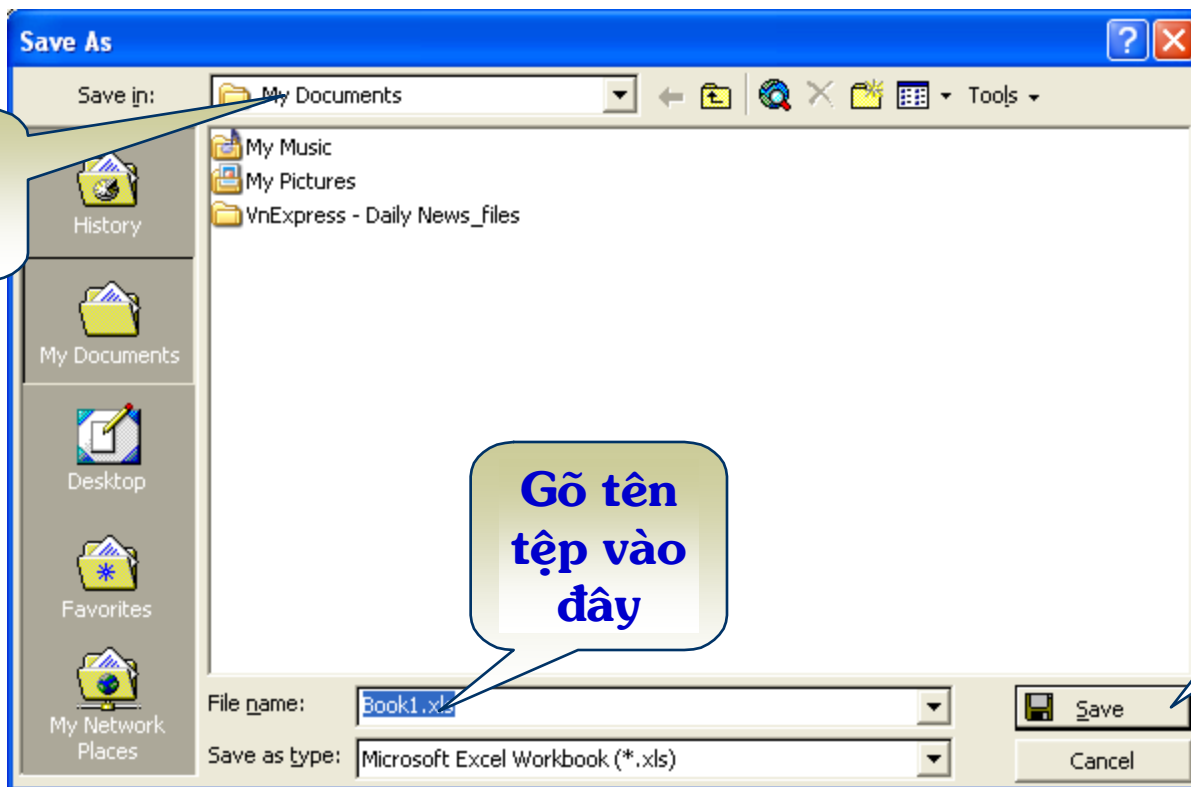
- ◆ Di chuyển chuột đến biên phải của tiêu đề ô (biên dưới của tiêu đề dòng) cần thay đổi
- ◆ Nhấn và rê chuột đến khi kích thước của dòng (cột) đạt như ý muốn

## b. Thay đổi bằng thao tác gián tiếp

- ◆ Di chuyển con trỏ đến ô cần thay đổi.
- ◆ Chọn thực đơn **Format**
  - Chọn **Row/Height**: để thay đổi chiều cao dòng
  - Chọn **Column/Width**: để thay đổi độ rộng cột

# 10. Lưu bảng tính trên đĩa

- ◆ Từ thực đơn **File -> Save**



Th- mục  
l- u giữ tệp

Gõ tên  
tệp vào  
đây

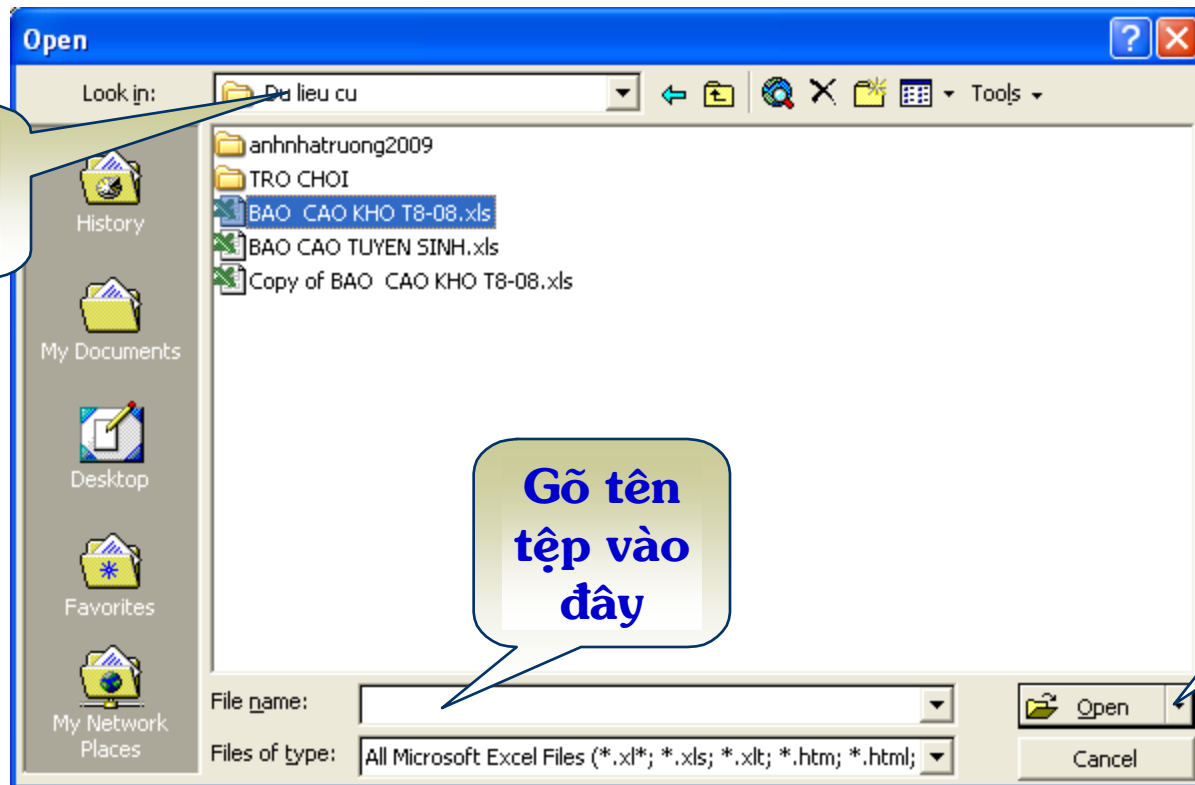
L- u tệp  
văn bản



# 11. Mở tệp tin

- ◆ Từ thực đơn **File -> Open**

Th-  
l- u giữ tệp



Gõ tên  
tệp vào  
đây

Mở văn  
bản

# 12. Đóng File

- ◆ **File -> Close**
- ◆ Trong hộp thoại xuất hiện
  - Chọn **Yes** để lưu và đóng file
  - Chọn **No** để không lưu và đóng file

# Bài 12: Làm việc với bảng tính

- ◆ Workbook và Worksheet
- ◆ Các thao tác với bảng tính
- ◆ Định dạng bảng tính
- ◆ Tìm kiếm và thay thế
- ◆ Sao chép dữ liệu, xoá dữ liệu
- ◆ In bảng tính

# 1. Workbook và Worksheet

## a. Giới thiệu Workbook và Worksheet.

- ◆ Workbook được ví như một quyển sổ chứa nhiều trang kẻ ô giống nhau, Worksheet là một trang trong sổ.
- ◆ Worksheet (Bảng tính): dùng để lưu trữ và xử lý số liệu, thực hiện các tính toán về toán học, kinh tế hay tính toán thống kê.
- ◆ Bảng tính được trình bày dữ liệu dưới dạng bảng được tạo bởi tập hợp của các dòng và cột

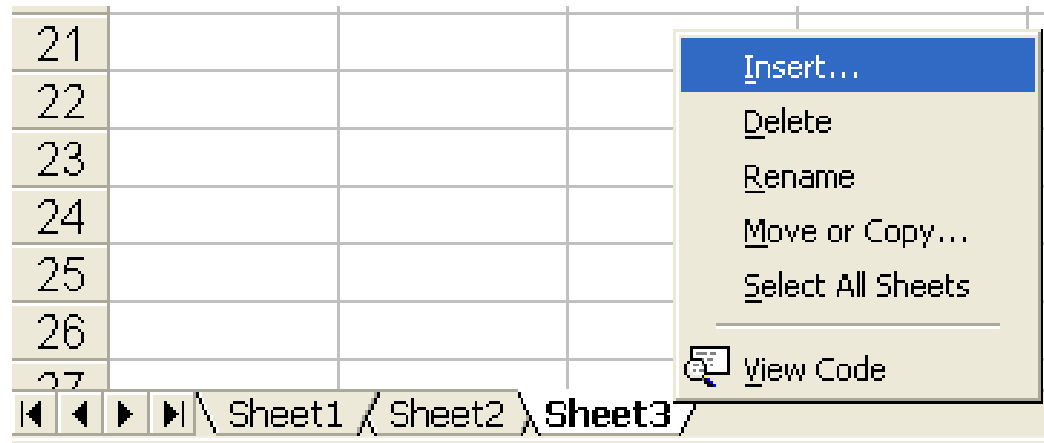
## b. Đặc điểm của Worksheet

- ◆ Dữ liệu được sắp xếp thành bảng hình chữ nhật
- ◆ Dữ liệu có thể dùng để tính toán một cách dễ dàng
- ◆ Các phép tính toán được thực hiện với bộ hàm thư viện định nghĩa rất phong phú
- ◆ Giá trị có được từ ô công thức được xử lý một cách tự động
- ◆ Có thể lọc rút trích dữ liệu theo một điều kiện.
- ◆ Có thể biểu diễn dữ liệu dưới dạng đồ thị

## 2. Các thao tác trên bảng tính

### a. Tạo thêm bảng tính

- ◆ Nhấp phải chuột vào vị trí sheet cần tạo
- ◆ Chọn **Insert**

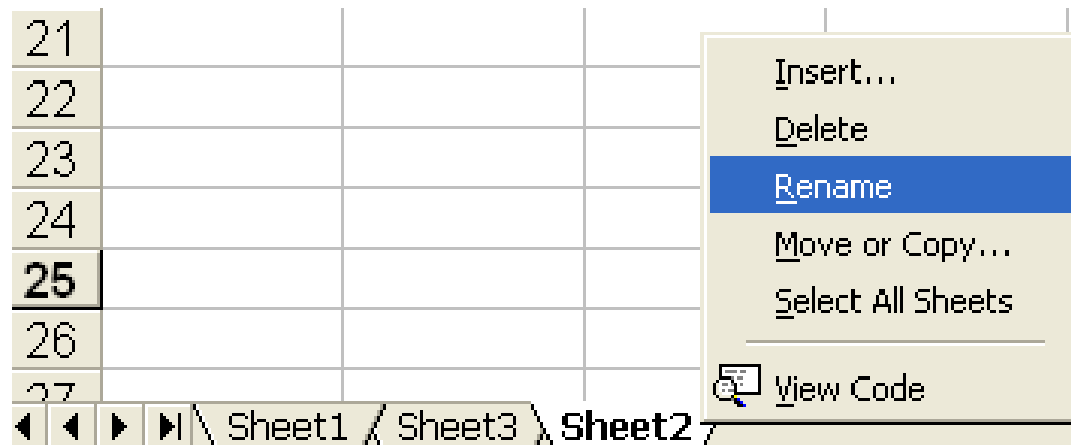


## **b. Di chuyển vị trí giữa các trang bảng tính**

- ◆ Nhấp và rê bảng tính cần di chuyển đến vị trí mới.

## c. Thay đổi tên bảng tính

- ◆ Nhấp phải chuột vào bảng tính cần đổi tên
- ◆ Chọn **Rename**
- ◆ Nhập tên mới, gõ Enter

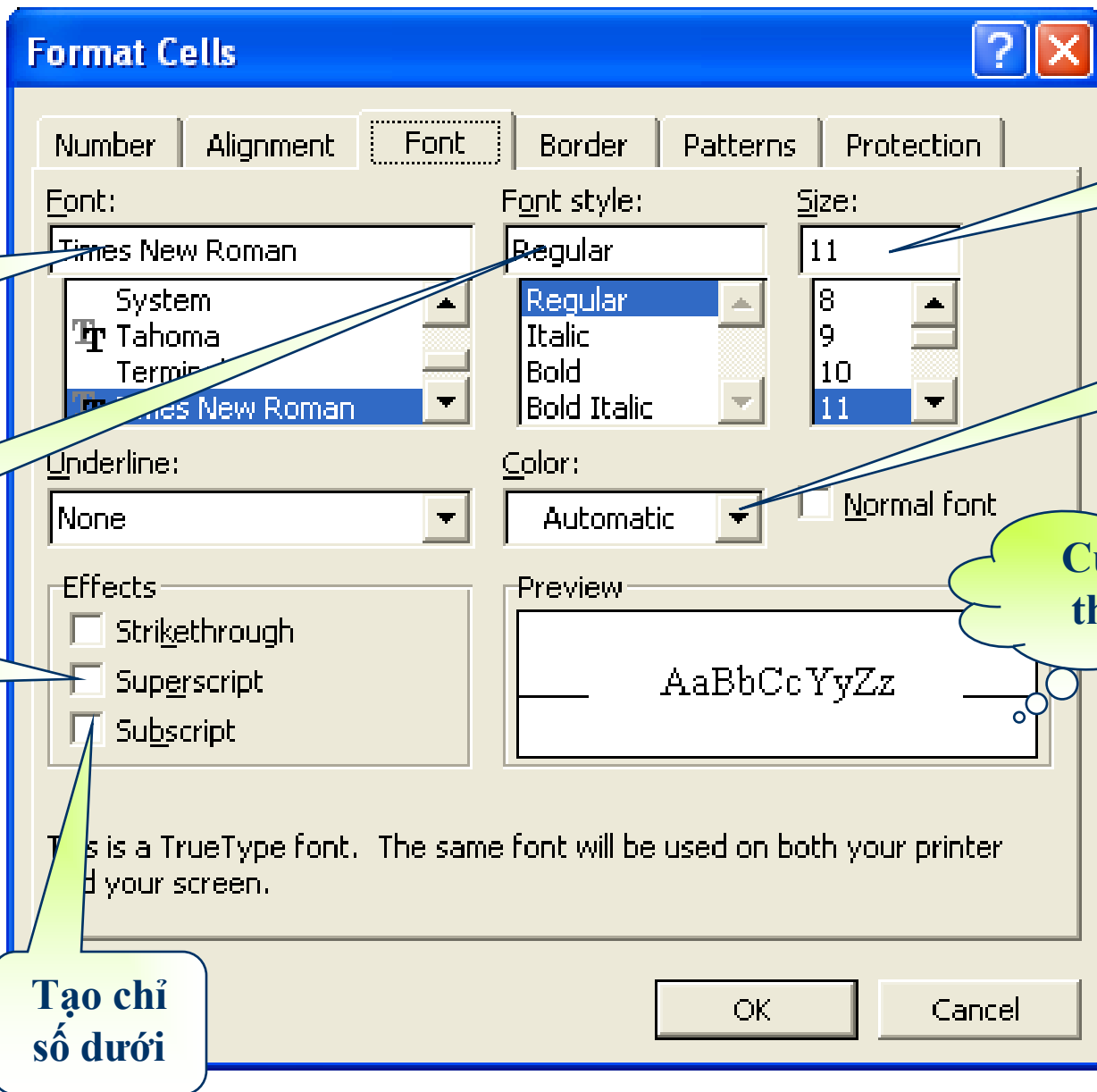




# 3. Định dạng bảng tính

## a. Định dạng văn bản trong ô

- ◆ Chọn vùng dữ liệu cần định dạng
- ◆ **Format -> cells** (Ctrl+1)
- ◆ Chọn trang **Font**
- ◆ Thiết lập các thông số sau:



**Chọn font chữ**

**Chọn kiểu chữ**

**Tạo chỉ số trên**

**Tạo chỉ số dưới**

**Chọn cỡ chữ**

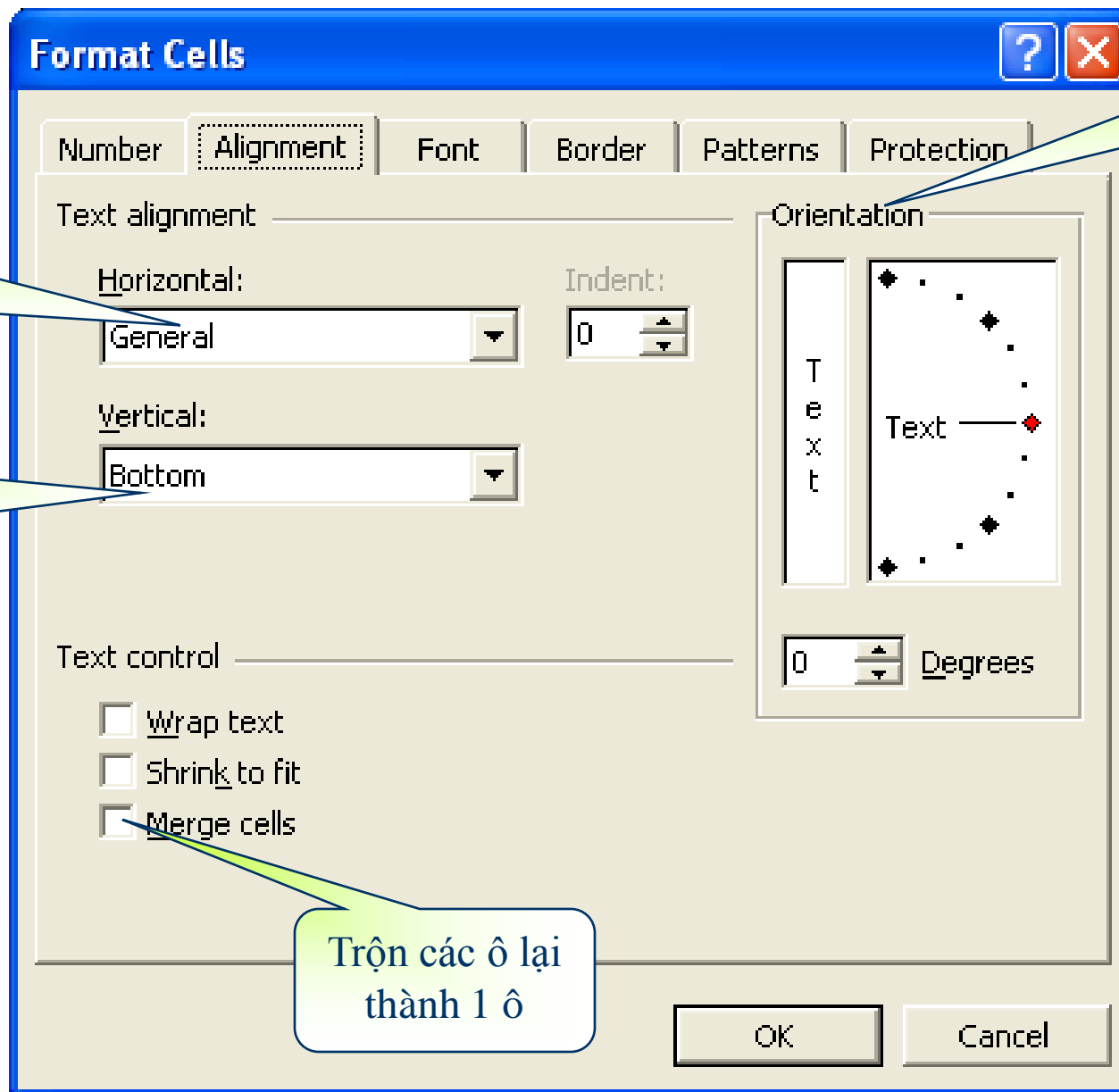
**màu chữ**

**Cửa sổ hiển thị kết quả chọn**

**Hộp thoại định dạng văn bản trong ô**

## b. Căn lề cho văn bản trong ô

- ◆ Chọn vùng dữ liệu cần định dạng
- ◆ **Format -> cells** (Ctrl+1)
- ◆ Chọn trang **Alignment**
- ◆ Thiết lập các thông số sau:



Căn theo chiều ngang

Căn theo chiều dọc

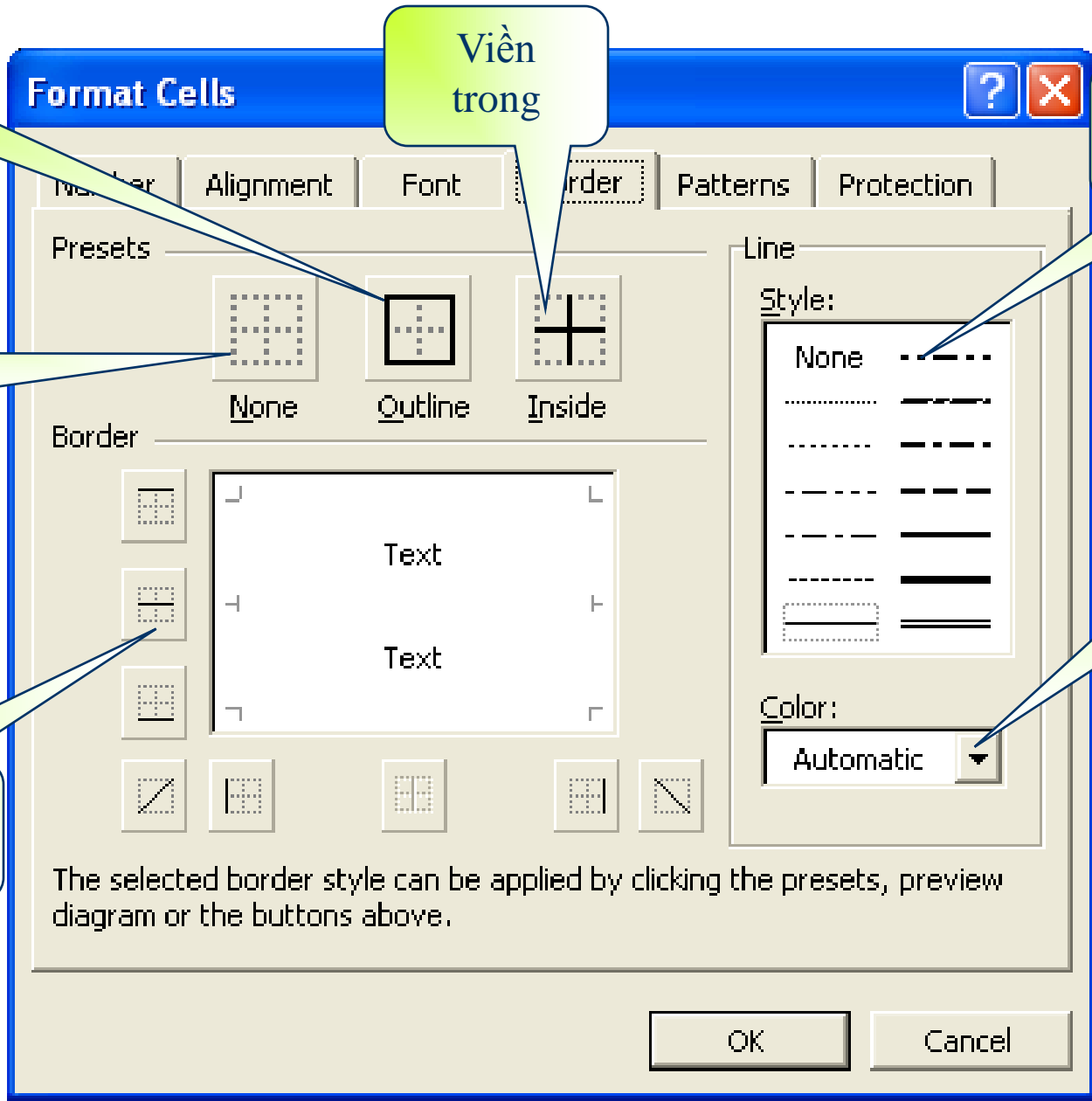
Trộn các ô lại thành 1 ô

Chọn hướng chữ

Hộp thoại căn lề văn bản trong ô

## c. Đường viền khung

- ◆ Chọn vùng dữ liệu cần định dạng
- ◆ **Format** -> **cells** (Ctrl+1)
- ◆ Chọn trang **Border**
- ◆ Thiết lập các thông số sau:



Viền bao ngoài

Viền trong

Chọn kiểu đường viền

Không tạo viền

Chọn màu đường viền

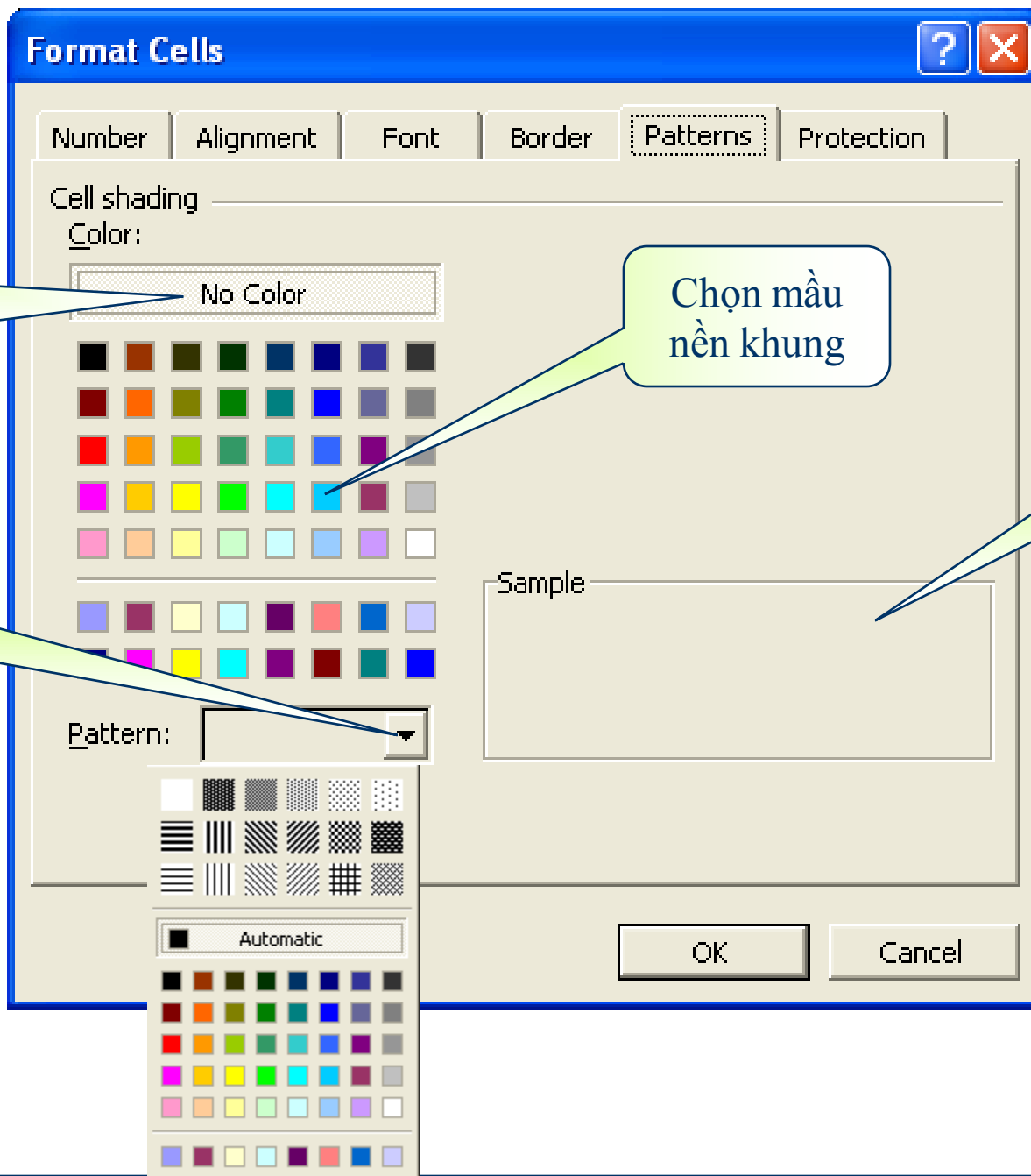
Chọn vị trí cần tạo viền

Hộp thoại tạo đường viền khung

## d. Màu nền khung

- ◆ Chọn vùng dữ liệu cần định dạng
- ◆ **Format -> cells** (Ctrl+1)
- ◆ Chọn trang **Patterns**
- ◆ Thiết lập các thông số sau:

# Hộp thoại tạo màu nền khung



Không màu nền khung

Chọn màu nền khung

Kiểu đồ nền

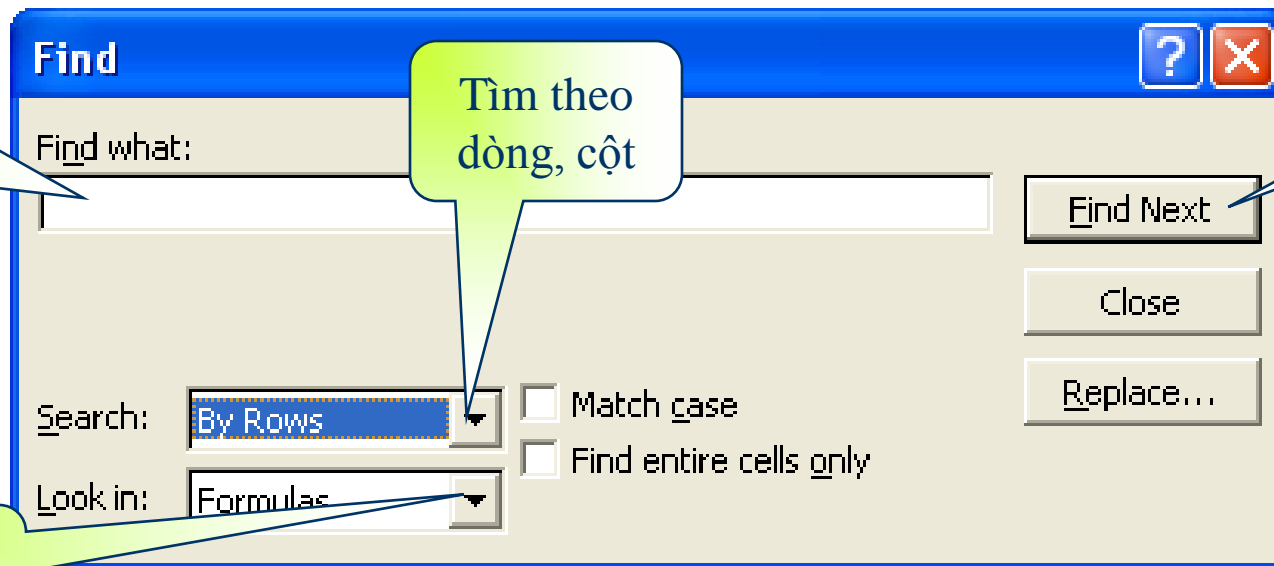
Cửa sổ hiển thị lựa chọn



# 4. Tìm kiếm và thay thế

## a. Tìm kiếm

- ◆ Mở hộp thoại tìm kiếm: Chọn Edit ->Find
- ◆ Thiết lập các thông số trong hộp thoại Find



Thực hiện tìm

Tìm theo dòng, cột

Nhập từ, nhóm từ cần tìm

Tìm công thức, giá trị, chú thích

# 5. Sao chép, di chuyển, xoá dữ liệu

## a. Sao chép dữ liệu

- ◆ Chọn vùng dữ liệu cần sao chép
- ◆ Nhấn tổ hợp phím Ctrl + C
- ◆ Di chuyển con trỏ đến vị trí mới
- ◆ Nhấn tổ hợp phím Ctrl + V

## b. Di chuyển vùng dữ liệu

- ◆ **Cách dùng bàn phím**
  - Chọn vùng dữ liệu cần di chuyển
  - Nhấn tổ hợp phím Ctrl + X
  - Di chuyển con trỏ đến vị trí mới
  - Nhấn tổ hợp phím Ctrl + V
- ◆ **Cách dùng chuột**
  - Chọn vùng dữ liệu cần di chuyển
  - Nhấn giữ phím chuột, kéo rê đến vị trí mới, rồi thả phím chuột

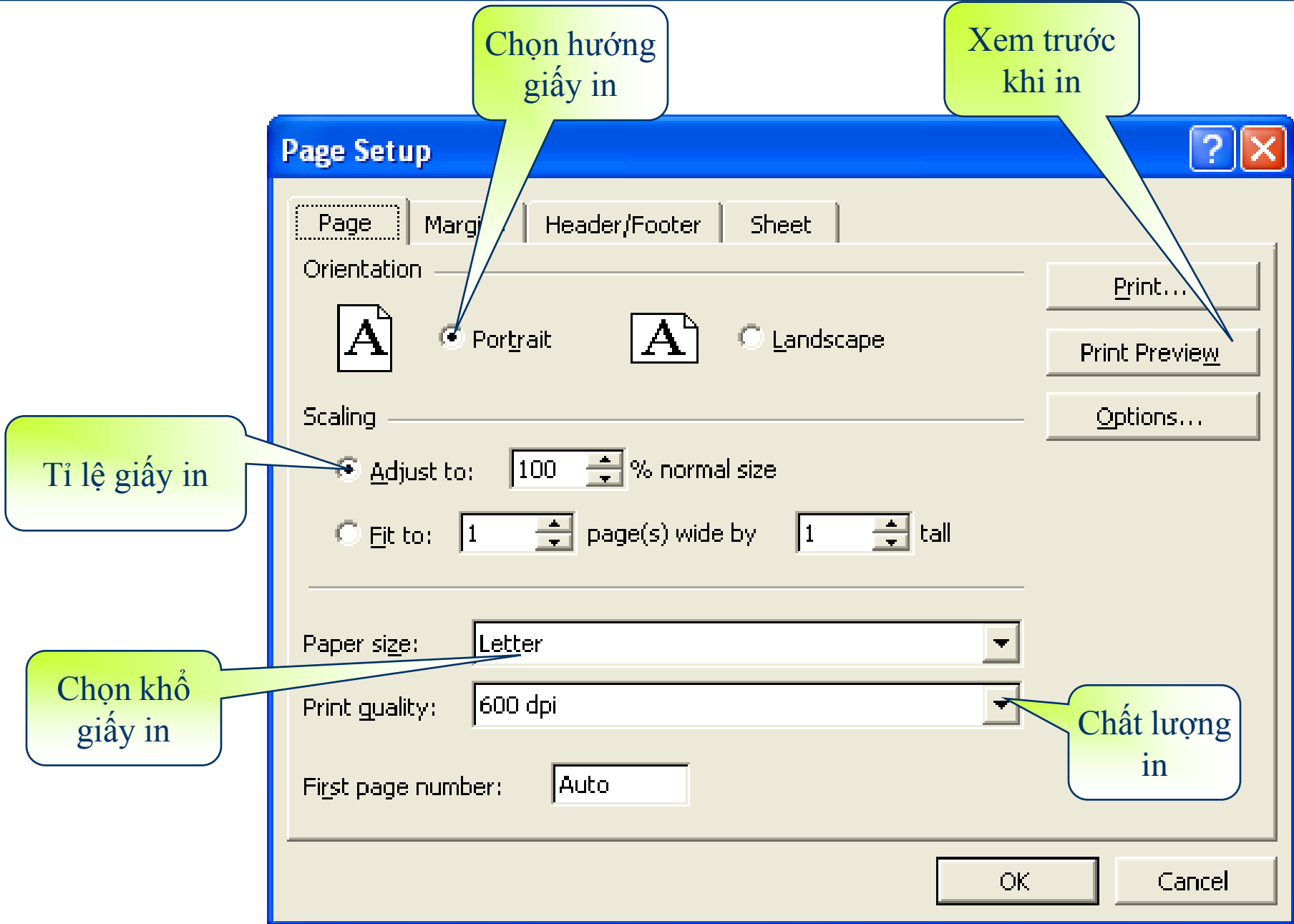
## c. Xoá dữ liệu

- ◆ Chọn vùng dữ liệu cần xoá
- ◆ **Edit -> Clear**
  - **All**: xoá tất
  - **Format**: xoá định dạng
  - **Contents**: xoá nội dung
  - **Comments**: xoá chú thích

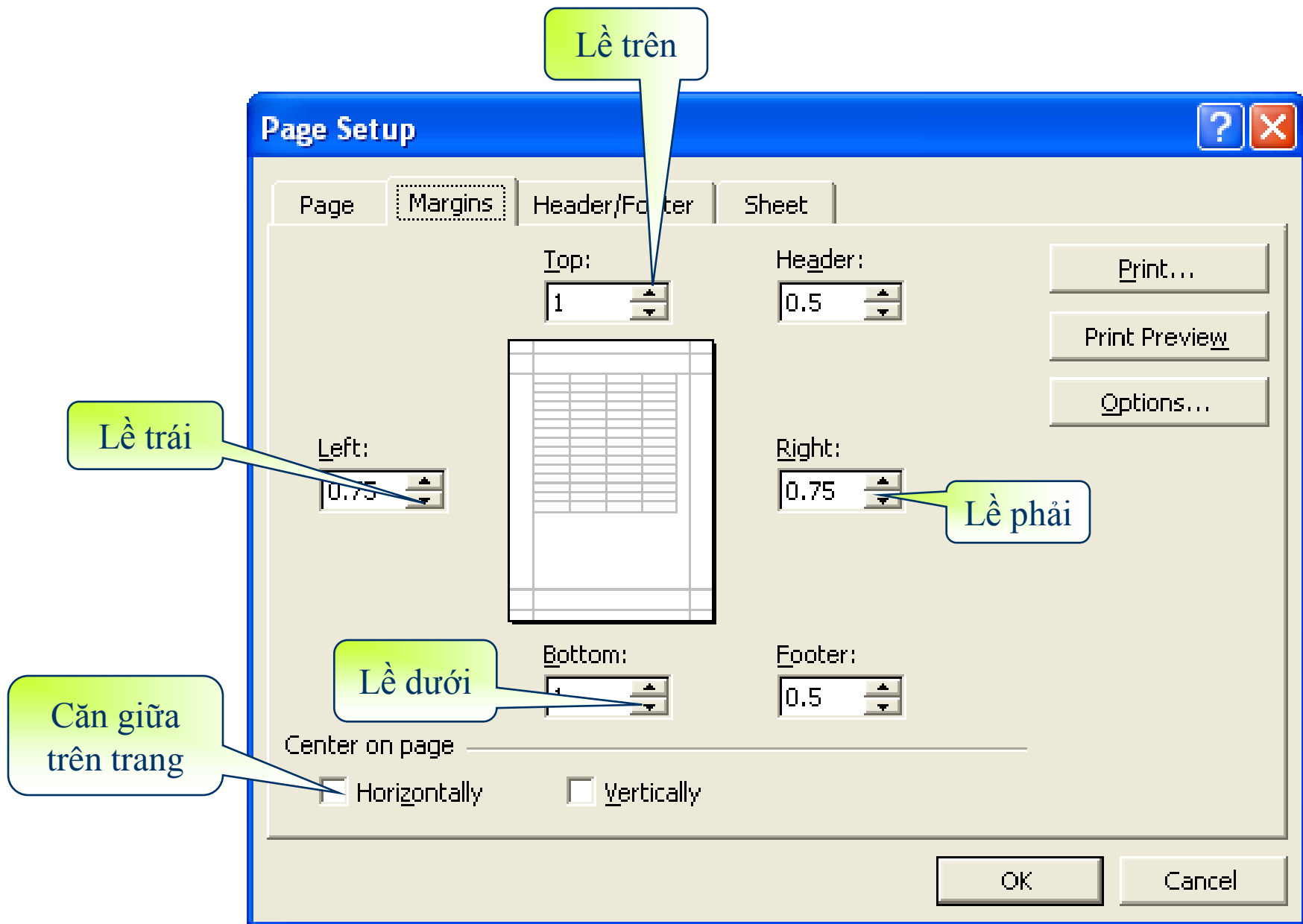
## 6. In ấn bảng tính

### a. Định dạng trang trước khi in

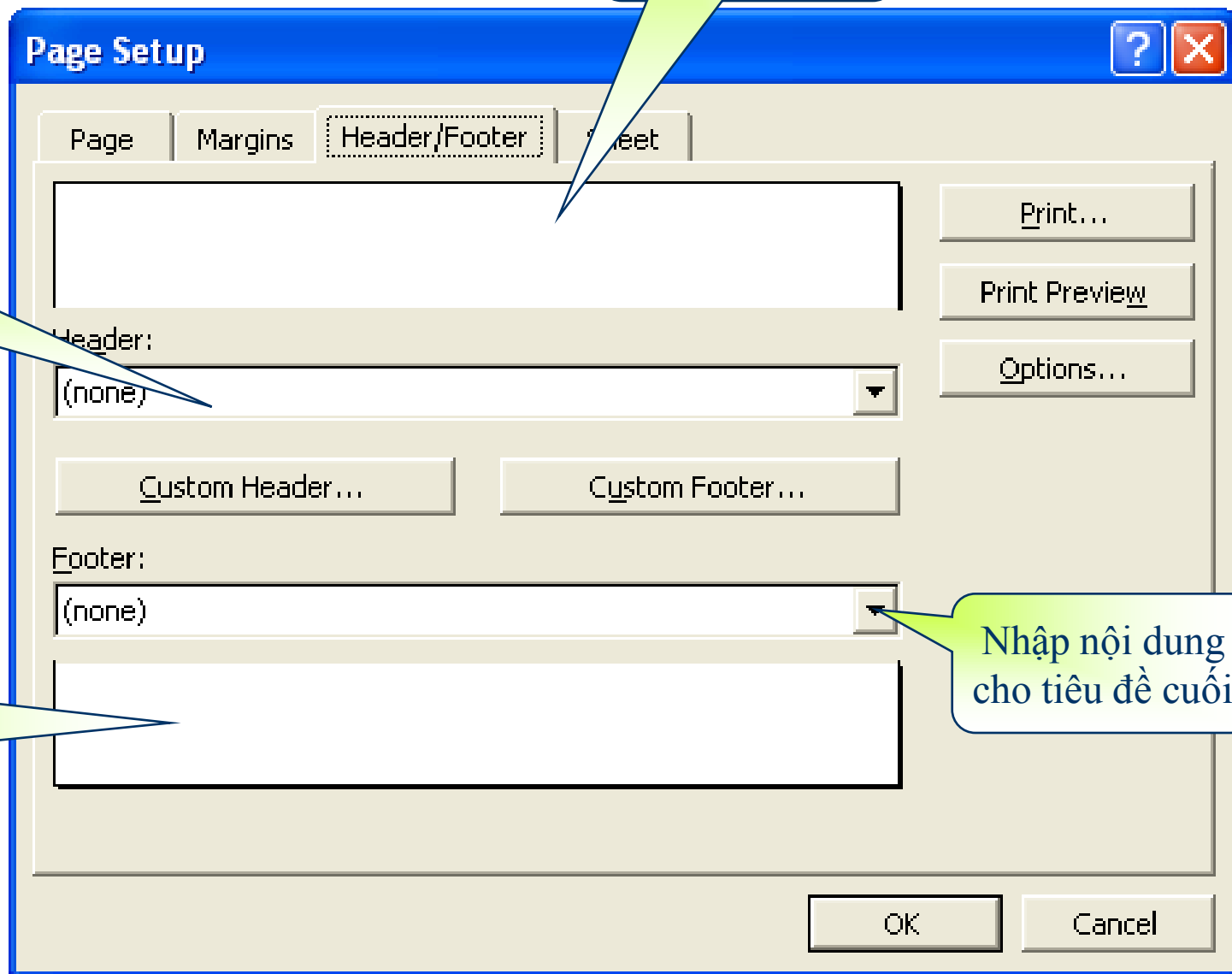
- ◆ Chọn thực đơn **File -> Page Setup**
- ◆ Thiết lập các thông số trong hộp thoại Page Setup



**Hộp thoại định dạng trang in**



**Hộp thoại định dạng lề cho trang in**



Cửa sổ hiển thị Header vừa nhập

Nhập nội dung cho tiêu đề đầu

Nhập nội dung cho tiêu đề cuối

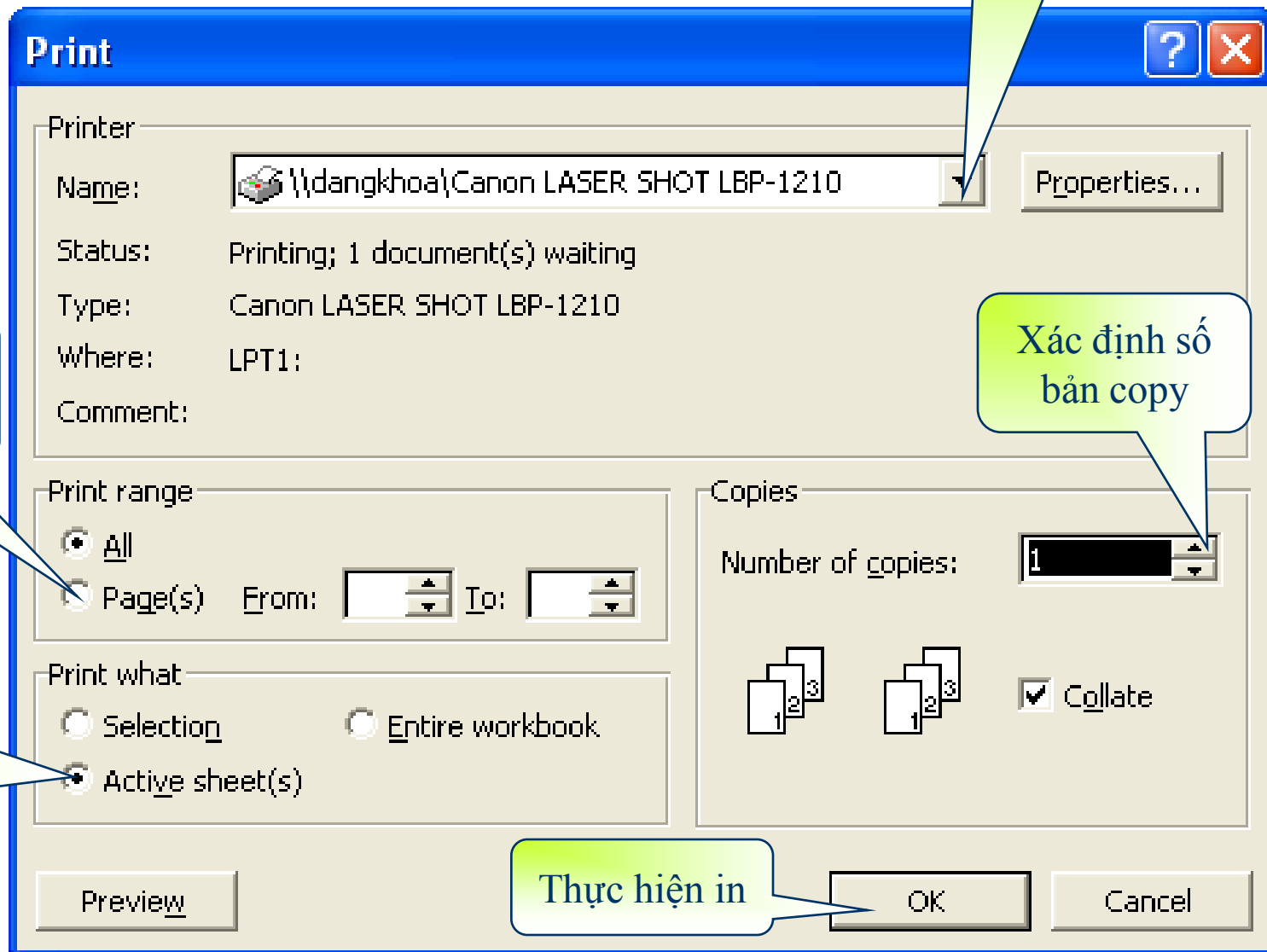
Cửa sổ hiển thị Footer vừa nhập

**Hộp thoại định dạng Header/Footer**



## b. Thực hiện in bảng tính

- ◆ Từ thực đơn **File -> Print**
- ◆ Thiết lập các thông số trong hộp thoại Print



**Hộp thoại In ấn**

# Bài 13. Tính toán dữ liệu trên bảng tính

- ◆ Tham chiếu địa chỉ
- ◆ Các hàm cơ bản

# I. Cách tham chiếu địa chỉ

- ◆ Sao chép công thức
- ◆ Tham chiếu địa chỉ

# 1. Sao chép công thức

- ◆ Khi thiết lập công thức trong bảng tính, ta có thể sao chép công thức đó cho nhiều ô với mục đích giống nhau
- ◆ Để thực hiện sao chép ta làm như sau:
  - Chọn ô chứa công thức cần sao chép
  - Đưa chuột đến góc dưới phải của ô vừa chọn
  - Nhấp và kéo đến ô cần tính và thả chuột

## 2. Tham chiếu địa chỉ

- ◆ Trong quá trình sử dụng dữ liệu trên bảng tính để lập những công thức, ta cần tham chiếu đến một hay nhiều ô chứa dữ liệu trên bảng tính để lấy được dữ liệu cần thiết cho công thức tính toán.
- ◆ Mỗi ô trên bảng tính Worksheet đều mang một địa chỉ cụ thể không trùng lặp.
- ◆ Địa chỉ của một ô gồm 2 thành phần
  - **Địa chỉ Cột:** cho biết ô đó nằm ở cột nào trên bảng tính
  - **Địa chỉ dòng:** cho biết ô đó nằm ở dòng nào trên bảng tính

## a. Tham chiếu địa chỉ tương đối

- ◆ Là cách sử dụng địa chỉ mà mỗi khi sao chép công thức sang vị trí mới trên Worksheet thì địa chỉ cũ sẽ tự động thay đổi thành địa chỉ mới tùy theo vị trí của ô được sao chép đến.

- ◆ Cú pháp

**<địa chỉ cột><địa chỉ dòng>**

- ◆ Ví dụ: **B10; D5**

## b. Tham chiếu địa chỉ tuyệt đối

- ◆ Tham chiếu địa chỉ tuyệt đối là cách cố định địa chỉ, trong công thức tính toán bất chấp việc sao chép công thức từ địa chỉ cũ sang địa chỉ mới.
- ◆ Cú pháp  
 $\$<\text{địa chỉ cột}>\$<\text{địa chỉ dòng}>$
- ◆ Ví dụ:  $\$C\$8$ ;  $\$E\$10$

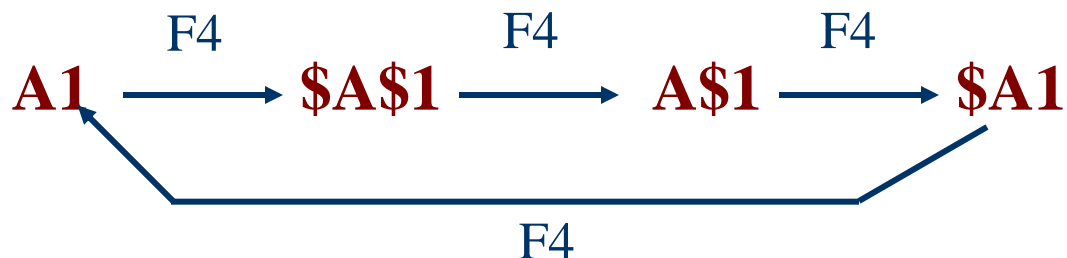


## c. Tham chiếu địa chỉ hỗn hợp

- ◆ Tham chiếu địa chỉ hỗn hợp là cách kết hợp 2 dạng tham chiếu địa chỉ tương đối và tuyệt đối nhưng được áp dụng cho từng thành phần của địa chỉ.
- ◆ Cú pháp:  
**<địa chỉ cột>\$<địa chỉ dòng>**  
**\$<địa chỉ cột><địa chỉ dòng>**
- ◆ Ví dụ: **A\$5; \$D8**

## d. Thay đổi cách tham chiếu địa chỉ

- ◆ Quét chọn địa chỉ cần thay đổi trong biểu thức.
- ◆ Nhấn phím F4 để thay đổi cách tham chiếu trong biểu thức.
  - Thứ tự thay đổi như sau: địa chỉ tương đối địa chỉ tuyệt đối địa chỉ tương đối cột tuyệt đối dòng địa chỉ tuyệt đối cột tương đối dòng
- ◆ Nhấn phím Enter để cập nhật giá trị cho biểu thức



## e. Tham chiếu vùng

- ◆ Trong trường hợp muốn tham chiếu đến các ô kế cận nhau, gọi là tham chiếu vùng.
- ◆ Cú pháp

**<địa chỉ ô bắt đầu><địa chỉ ô kết thúc>**

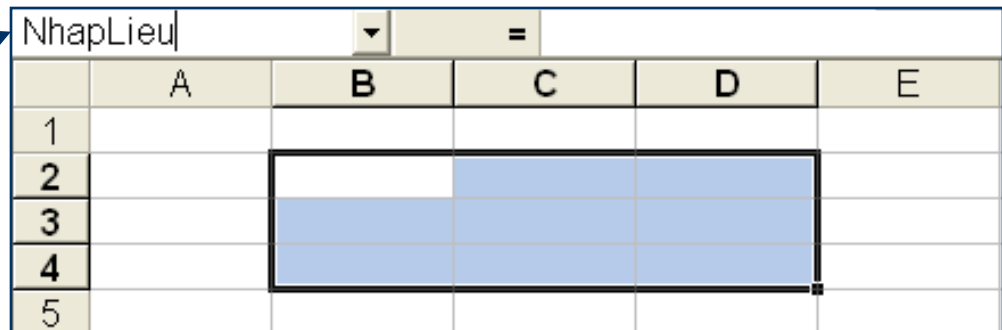
**B2:D4** →

|   | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 |   |   |   |   |   |
| 2 |   |   |   |   |   |
| 3 |   |   |   |   |   |
| 4 |   |   |   |   |   |
| 5 |   |   |   |   |   |

## g. Tham chiếu theo tên vùng

- ◆ Tham chiếu theo tên vùng chỉ dùng cho địa chỉ tuyệt đối.
- ◆ Các bước đặt tên vùng:
  - Quét chọn vùng muốn đặt tên
  - Nhập tên vùng vào ô Name Box trên thanh công cụ
  - Sử dụng tên vùng vừa tạo trong biểu thức hay hàm

Nhập tên vùng



|   | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 |   |   |   |   |   |
| 2 |   |   |   |   |   |
| 3 |   |   |   |   |   |
| 4 |   |   |   |   |   |
| 5 |   |   |   |   |   |

## II. CÁC HÀM CƠ BẢN

- ◆ Hàm Logic
- ◆ Hàm về số
- ◆ Hàm về chuỗi
- ◆ Hàm Ngày giờ
- ◆ Hàm thống kê
- ◆ Hàm điều kiện
- ◆ Hàm tìm kiếm

# 1. Các hàm LOGIC

- ◆ Trong Excel các hàm Logic thường trả về 2 giá trị:
  - FALSE: tương ứng với 0
  - TRUE: tương ứng với giá trị  $\neq 0$  (thường là  $-1$ )

# a. Hàm AND

- ◆ Cú pháp:

**AND(<biểu thức logic 1>,<biểu thức logic 2>,...)**

- ◆ Ý nghĩa:

- Hàm chỉ trả về giá trị TRUE khi các tham số truyền vào có giá trị là TRUE, ngược lại trả về giá trị FALSE.

- ◆ Ví dụ:

- $\text{AND}(3 > 6, 5 = 5) \rightarrow \text{FALSE}$
- $\text{AND}(3 = 3, 6 > 5) \rightarrow \text{TRUE}$

## b. Hàm OR

- ◆ Cú pháp:

**OR(<biểu thức logic 1>,<biểu thức logic 2>,...)**

- ◆ Ý nghĩa:

- Hàm chỉ trả về giá trị FALSE khi các tham số truyền vào có giá trị là FALSE, ngược lại trả về giá trị TRUE

- ◆ Ví dụ:

- $OR(3 > 6, 5 = 5) \rightarrow TRUE$
- $AND(3 > 3, 6 < 5) \rightarrow FALSE$



## 2. Hàm về số

- ◆ Hàm ABS
- ◆ Hàm INT
- ◆ Hàm ROUND
- ◆ Hàm MOD

# a. Hàm ABS

- ◆ Cú pháp:

**ABS(<biểu thức>)**

- ◆ Ý nghĩa:

- Hàm trả về giá trị tuyệt đối của biểu thức truyền vào

- ◆ Ví dụ:

- $ABS(-7)=7$
- $ABS(8)=8$

## b. Hàm INT

- ◆ Cú pháp:

**INT(<biểu thức>)**

- ◆ Ý nghĩa:

- Hàm trả về số nguyên có giá trị nhỏ hơn giá trị của biểu thức truyền vào.

- ◆ Ví dụ:

- $\text{INT}(7.4)=4$
- $\text{INT}(8.76)=8$

## c. Hàm MOD

- ◆ Cú pháp:

**MOD(<biểu thức 1>, <biểu thức 2>)**

- ◆ Ý nghĩa:

- Hàm trả về phần dư của phép chia <biểu thức 1>/<biểu thức 2>

- ◆ Ví dụ:

- $\text{MOD}(7,4)=3$
- $\text{MOD}(8,7)=1$

## d. Hàm ROUND

- ◆ Cú pháp:

**ROUND(<biểu thức>, <vị trí làm tròn>)**

- ◆ Ý nghĩa:

- Hàm trả về giá trị được làm tròn của <biểu thức> tại vị trí làm tròn do tham số thứ 2 chỉ định

- ◆ Nguyên tắc:

- Nếu giá trị tại vị trí tham số thứ hai +1 lớn hơn 5, giá trị tại vị trí tham số thứ hai được tăng lên một, ngược lại giữ nguyên

- ◆ Ví dụ

- $\text{ROUND}(74.32,1)=74.3$
- $\text{ROUND}(74.56,1)=74.6$

# 3. Các hàm về chuỗi

- ◆ Hàm LEFT
- ◆ Hàm RIGHT
- ◆ Hàm MID
- ◆ Hàm LEN

# a. Hàm LEFT

- ◆ Cú pháp:

**LEFT (<biểu thức chuỗi>, <số ký tự>)**

- ◆ Ý nghĩa:

- Hàm trả về các ký tự từ bên trái của một chuỗi. Số ký tự trả về được chỉ định trong tham số thứ 2.

- ◆ Ví dụ:

- LEFT('Công ty may 10,7)='Công ty'
- LEFT('CĐN Long Biên',3)='CĐN'

## b. Hàm RIGHT

- ◆ Cú pháp:

**RIGHT (<biểu thức chuỗi>, <số ký tự>)**

- ◆ Ý nghĩa:

- Hàm trả về các ký tự từ bên phải của một chuỗi. Số ký tự trả về được chỉ định trong tham số thứ 2.

- ◆ Ví dụ:

- RIGHT('Công ty may 10',6)='may 10'
- RIGHT('CĐN Long Biên',9)='Long Biên'



## c. Hàm MID

- ◆ Cú pháp:

**MID (<biểu thức chuỗi>, <vị trí bắt đầu>, <số ký tự>)**

- ◆ Ý nghĩa:

- Hàm trả về các ký nằm ở giữa của một chuỗi. Ký tự bắt đầu được chỉ định trong tham số thứ 2, Số ký tự trả về được chỉ định trong tham số thứ 3.

- ◆ Ví dụ:

- MID('Công ty may 10',9,3)='may'
- MID('CĐN Long Biên',5,4)='Long'

## d. Hàm LEN

- ◆ Cú pháp:

**LEN (<biểu thức chuỗi>)**

- ◆ Ý nghĩa:

- Hàm trả về số ký tự có trong <biểu thức chuỗi>.

- ◆ Ví dụ:

- LEN('Công ty may 10')=15
- MID('CĐN Long Biên')=13

## 4. Hàm về ngày, giờ

- ◆ Hàm DATE
- ◆ Hàm TIME
- ◆ Hàm DAY
- ◆ Hàm MONTH
- ◆ Hàm YEAR
- ◆ Hàm NOW
- ◆ Hàm HOUR
- ◆ Hàm MINUTE
- ◆ Hàm SECOND
- ◆ Hàm WEEKDAY
- ◆ Hàm DAY360
- ◆ Hàm DATEVALUE

# a. Hàm DATE

- ◆ Cú pháp:

**DATE (<năm>, <tháng>, <ngày>)**

- ◆ Ý nghĩa:

- Hàm trả về giá trị kiểu ngày dựa trên các tham số truyền vào.

- ◆ Ví dụ:

- DATE(2009,2,12)=12/9/2009

## b. Hàm TIME

- ◆ Cú pháp:

**TIME (<giờ>,<phút>,<giây>)**

- ◆ Ý nghĩa:

- Hàm trả về giá trị kiểu giờ dựa trên các tham số truyền vào.

- ◆ Ví dụ:

- TIME(4,20,12)=4:20:12

## c. Hàm DAY

- ◆ Cú pháp:

**DAY (<tham số kiểu ngày, giờ>)**

- ◆ Ý nghĩa:

- Hàm trả về ngày trong tháng dựa trên các tham số truyền vào.

- ◆ Ví dụ:

- DAY(31/12/2008)=31

## d. Hàm MONTH

- ◆ Cú pháp:

**DAY (<tham số kiểu ngày, giờ>)**

- ◆ Ý nghĩa:

- Hàm trả về tháng trong năm dựa trên các tham số truyền vào.

- ◆ Ví dụ:

- MONTH(31/12/2008)=12

## e. Hàm YEAR

- ◆ Cú pháp:

**YEAR (<tham số kiểu ngày, giờ>)**

- ◆ Ý nghĩa:

- Hàm trả về năm dựa trên các tham số truyền vào.

- ◆ Ví dụ:

- YEAR (31/12/2008)=2008



## e. Hàm NOW

- ◆ Cú pháp:

**NOW ()**

- ◆ Ý nghĩa:

- Hàm trả về ngày giờ hiện hành của hệ thống.

- ◆ Ví dụ:

- NOW()=8/10/2009 15:28

## g. Hàm HOUR

- ◆ Cú pháp:

**HOUR (<tham số kiểu ngày, giờ>)**

- ◆ Ý nghĩa:

- Hàm trả về giờ trong ngày dựa trên tham số truyền.

- ◆ Ví dụ:

- HOUR (15:25:5)=15

## h. Hàm MINUTE

- ◆ Cú pháp:

**MINUTE (<tham số kiểu ngày, giờ>)**

- ◆ Ý nghĩa:

- Hàm trả về phút dựa trên tham số truyền vào.

- ◆ Ví dụ:

- MINUTE (15:25:5)=25

# i. Hàm SECOND

- ◆ Cú pháp:

**SECOND (<tham số kiểu ngày, giờ>)**

- ◆ Ý nghĩa:

- Hàm trả về giây dựa trên tham số truyền vào.

- ◆ Ví dụ:

- SECOND (15:25:5)=5

# k. Hàm WEEKDAY

## ◆ Cú pháp:

**WEEKDAY (<tham số kiểu ngày, giờ>,<thứ đầu tuần>)**

## ◆ Ý nghĩa:

- Hàm trả về thứ trong tuần dựa trên tham số truyền vào và phụ thuộc vào giá trị của <thứ đầu tuần>.
  - 1: giá trị trả về là CN;T2;...,T7 tương ứng giá trị 1-7
  - 2: giá trị trả về là T2;...,T7;CN tương ứng giá trị 1-7
  - 3: giá trị trả về là T2;...,T7;CN tương ứng giá trị 0-6

## ◆ Ví dụ:

- WEEKDAY (8/10/2009,1)=5
- WEEKDAY (8/10/2009,2)=4

# 1. Hàm DAYS360

- ◆ Cú pháp:

**DAYS360 (<ngày bắt đầu>, <ngày kết thúc>)**

- ◆ Ý nghĩa:

- Hàm trả về số ngày là khoảng cách giữa ngày bắt đầu và ngày kết thúc.

- ◆ Ví dụ:

- DAYS360 (“10/8/2009”, “10/30/2009”)=22

## m. Hàm DATEVALUE

- ◆ Cú pháp:

**DATEVALUE (<chuỗi kiểu ngày>)**

- ◆ Ý nghĩa:

- Hàm trả về biểu thức dạng ngày, giờ dựa trên tham số truyền vào.

- ◆ Ví dụ:

- DATEVALUE (“10/8/2009”)=10/8/2009

# 5. Các hàm thống kê

- ◆ Hàm SUM
- ◆ Hàm COUNT
- ◆ Hàm AVERAGE
- ◆ Hàm MAX
- ◆ Hàm MIN



# a. Hàm SUM

- ◆ Cú pháp:

**SUM (<số thứ 1>, <số thứ 2>, ..., <số thứ n>)**

**SUM(<địa chỉ vùng>)**

- ◆ Ý nghĩa:

- Hàm trả về giá trị tổng của các tham số truyền vào.

- ◆ Ví dụ:

- SUM(1,2,3,4,5,6)=21
- SUM(A1:D2)=36

|   | A | B | C | D |
|---|---|---|---|---|
| 1 | 1 | 2 | 3 | 4 |
| 2 | 5 | 6 | 7 | 8 |

## b. Hàm COUNT

- ◆ Cú pháp:

**COUNT (<số thứ 1>, <số thứ 2>, ..., <số thứ n>)**

**COUNT(<địa chỉ vùng>)**

- ◆ Ý nghĩa:

- Hàm đếm các số trong danh sách các tham số truyền vào.

- ◆ Ví dụ:

- COUNT(1,2,3,4,5,6)=6
- COUNT(A1:D2)=8

|   | A | B | C | D |
|---|---|---|---|---|
| 1 | 1 | 2 | 3 | 4 |
| 2 | 5 | 6 | 7 | 8 |

## c. Hàm AVERAGE

- ◆ Cú pháp:

**AVERAGE (<số thứ 1>, <số thứ 2>, ..., <số thứ n>)**

**AVERAGE (<địa chỉ vùng>)**

- ◆ Ý nghĩa:

- Hàm trả về giá trị trung bình qua các số trong danh sách các tham số truyền vào.

- ◆ Ví dụ:

- AVERAGE (1,2,3,4,5,6)=3.5
- AVERAGE (A1:D2)=4.5

|   | A | B | C | D |
|---|---|---|---|---|
| 1 | 1 | 2 | 3 | 4 |
| 2 | 5 | 6 | 7 | 8 |

## d. Hàm MAX

- ◆ Cú pháp:

**MAX (<số thứ 1>, <số thứ 2>, ..., <số thứ n>)**

**MAX (<địa chỉ vùng>)**

- ◆ Ý nghĩa:

- Hàm trả về giá trị lớn nhất qua các số trong danh sách các tham số truyền vào.

- ◆ Ví dụ:

- MAX (1,2,3,4,5,6)=6
- MAX (A1:D2)=8

|   | A | B | C | D |
|---|---|---|---|---|
| 1 | 1 | 2 | 3 | 4 |
| 2 | 5 | 6 | 7 | 8 |

## e. Hàm MIN

- ◆ Cú pháp:

**MIN (<số thứ 1>, <số thứ 2>, ..., <số thứ n>)**

**MIN (<địa chỉ vùng>)**

- ◆ Ý nghĩa:

- Hàm trả về giá trị lớn nhất qua các số trong danh sách các tham số truyền vào.

- ◆ Ví dụ:

- MIN (1,2,3,4,5,6)=1
- MIN (A1:D2)=1

|   | A | B | C | D |
|---|---|---|---|---|
| 1 | 1 | 2 | 3 | 4 |
| 2 | 5 | 6 | 7 | 8 |

# 6. Hàm điều kiện

- ◆ Hàm IF
- ◆ Hàm SUMIF
- ◆ Hàm COUNTIF

## a. Hàm IF

- ◆ Cú pháp:

**IF (<biểu thức logic>, <giá trị đúng>, <giá trị sai>)**

- ◆ Ý nghĩa:

- Hàm trả về một trong hai giá trị tùy theo giá trị của biểu thức logic là TRUE hoặc FALSE.

- ◆ Ví dụ:

|   | A             | B           | C                          | D | E |
|---|---------------|-------------|----------------------------|---|---|
| 1 | <b>Họ tên</b> | <b>Điểm</b> | <b>Kết quả</b>             |   |   |
| 2 | Anh           | 8           | =if(B2>=5,"Lên lớp","Đúp") |   |   |
| 3 | Mai           | 5           |                            |   |   |
| 4 | Dũng          | 4           |                            |   |   |

## b. Hàm SUMIF

- ◆ Cú pháp:

**SUMIF (<vùng kiểm tra>, <điều kiện>, <vùng tính tổng>)**

- ◆ Ý nghĩa:

- Hàm trả về giá trị tổng của những phần tử được chọn trên vùng tính tổng, những phần tử này được chọn ứng với những dòng trên vùng kiểm tra có giá trị thoả mãn điều kiện.

- ◆ Ví dụ:



## Ví dụ: Tính tổng lương của các nhân viên trong công ty

|    | A                           | B                               | C            | D | E |
|----|-----------------------------|---------------------------------|--------------|---|---|
| 1  | <b>BẢNG LƯƠNG NHÂN VIÊN</b> |                                 |              |   |   |
| 2  | <b>Họ và tên</b>            | <b>Chức vụ</b>                  | <b>Lương</b> |   |   |
| 3  | Anh                         | GD                              | 4,000,000    |   |   |
| 4  | Bình                        | NV                              | 2,500,000    |   |   |
| 5  | Mai                         | NV                              | 2,000,000    |   |   |
| 6  | Hùng                        | NV                              | 1,800,000    |   |   |
| 7  | Chung                       | PGD                             | 3,000,000    |   |   |
| 8  |                             |                                 |              |   |   |
| 9  | Tổng lương của các NV:      | <b>=SUMIF(B3:B7,"NV",C3:C7)</b> |              |   |   |
| 10 |                             |                                 |              |   |   |

## c. Hàm COUNTIF

- ◆ Cú pháp:

**COUNTIF (<vùng kiểm tra>, <điều kiện>)**

- ◆ Ý nghĩa:

- Hàm trả về giá trị đếm những phần tử được chọn trên vùng kiểm tra mà thoả mãn điều kiện.

- ◆ Ví dụ:

## Ví dụ: Đếm số nhân viên trong công ty

|    | A                           | B              | C                           | D | E |
|----|-----------------------------|----------------|-----------------------------|---|---|
| 1  | <b>BẢNG LƯƠNG NHÂN VIÊN</b> |                |                             |   |   |
| 2  | <b>Họ và tên</b>            | <b>Chức vụ</b> | <b>Lương</b>                |   |   |
| 3  | Anh                         | GD             | 4,000,000                   |   |   |
| 4  | Bình                        | NV             | 2,500,000                   |   |   |
| 5  | Mai                         | NV             | 2,000,000                   |   |   |
| 6  | Hùng                        | NV             | 1,800,000                   |   |   |
| 7  | Chung                       | PGD            | 3,000,000                   |   |   |
| 8  |                             |                |                             |   |   |
| 9  | <b>Đếm số nhân viên.</b>    |                | <b>=COUNTIF(B3:B7,"NV")</b> |   |   |
| 10 |                             |                |                             |   |   |

# 7. Hàm tìm kiếm

- ◆ Hàm VLOOKUP
- ◆ Hàm HLOOKUP

# a. Hàm VLOOKUP

## ◆ Cú pháp:

**VLOOKUP (<giá trị tìm>, <vùng tìm>, <thứ tự cột lấy giá trị>, <cách tìm>)**

## ◆ Mô tả:

- <giá trị tìm>: Giá trị hàm sẽ tìm trên cột đầu tiên
- <vùng tìm>: vùng tìm kiếm có thể là bảng, vùng có đặt tên, một mảng.
- <thứ tự cột lấy giá trị>: là thứ tự cột sẽ lấy giá trị trả về nếu tìm thấy.
- <Cách tìm>:
  - 0: tìm gần đúng
  - 1: tìm chính xác

# a. Hàm VLOOKUP

## ◆ Ý nghĩa:

- Hàm tìm kiếm giá trị trên cột đầu tiên của vùng tìm và trả về một giá trị của dòng tìm thấy trên cột được chỉ định trong hàm.
- VLOOKUP là hàm tìm kiếm theo chiều dọc (trên cột)

## Ví dụ: Tính phụ Cap chuc vu

|    | A                           | B              | C                                | D | E |
|----|-----------------------------|----------------|----------------------------------|---|---|
| 1  | <b>BẢNG LƯƠNG NHÂN VIÊN</b> |                |                                  |   |   |
| 2  | <b>Họ và tên</b>            | <b>Chức vụ</b> | <b>Phụ cấp</b>                   |   |   |
| 3  | Anh                         | GD             | =VLOOKUP(B3,\$A\$12:\$B\$15,2,0) |   |   |
| 4  | Bình                        | NV             |                                  |   |   |
| 5  | Mai                         | NV             |                                  |   |   |
| 6  | Hùng                        | NV             |                                  |   |   |
| 7  | Minh                        | PGD            |                                  |   |   |
| 8  | Chung                       | PGD            |                                  |   |   |
| 9  |                             |                |                                  |   |   |
| 10 |                             |                |                                  |   |   |
| 11 | <b>BẢNG PHỤ CẤP</b>         |                |                                  |   |   |
| 12 | <b>Chức vụ</b>              | <b>Phụ cấp</b> |                                  |   |   |
| 13 | GD                          | 500,000        |                                  |   |   |
| 14 | PGD                         | 300,000        |                                  |   |   |
| 15 | NV                          | 100,000        |                                  |   |   |

## b. Hàm HLOOKUP

### ◆ Cú pháp:

**HLOOKUP (<giá trị tìm>, <vùng tìm>, <thứ tự dòng lấy giá trị>, <cách tìm>)**

### ◆ Mô tả:

- <giá trị tìm>: Giá trị hàm sẽ tìm trên dòng đầu tiên
- <vùng tìm>: vùng tìm kiếm có thể là bảng, vùng có đặt tên, một mảng.
- <thứ tự dòng lấy giá trị>: là thứ tự dòng sẽ lấy giá trị trả về nếu tìm thấy.
- <Cách tìm>:
  - 0: tìm gần đúng
  - 1: tìm chính xác



# a. Hàm VLOOKUP

## ◆ Ý nghĩa:

- Hàm tìm kiếm giá trị trên dòng đầu tiên của vùng tìm và trả về một giá trị của cột tìm thấy trên dòng được chỉ định trong hàm.
- HLOOKUP là hàm tìm kiếm theo chiều ngang (trên dòng)

## Ví dụ: Tính phụ cấp chức vụ

|    | A                           | B              | C                                | D       | E |
|----|-----------------------------|----------------|----------------------------------|---------|---|
| 1  | <b>BẢNG LƯƠNG NHÂN VIÊN</b> |                |                                  |         |   |
| 2  | <b>Họ và tên</b>            | <b>Chức vụ</b> | <b>Phụ cấp</b>                   |         |   |
| 3  | Anh                         | GD             | =HLOOKUP(B3,\$A\$10:\$D\$11,2,0) |         |   |
| 4  | Bình                        | NV             |                                  |         |   |
| 5  | Mai                         | NV             |                                  |         |   |
| 6  | Hùng                        | NV             |                                  |         |   |
| 7  | Minh                        | PGD            |                                  |         |   |
| 8  | Chung                       | PGD            |                                  |         |   |
| 9  |                             |                |                                  |         |   |
| 10 | <b>Chức vụ</b>              | GD             | PGD                              | NV      |   |
| 11 | <b>Phụ cấp</b>              | 500,000        | 300,000                          | 100,000 |   |
| 12 |                             |                |                                  |         |   |

# **Bài 14. Cơ sở dữ liệu trên bảng tính**

- ◆ Các khái niệm trong cơ sở dữ liệu
- ◆ Các thao tác trên cơ sở dữ liệu

# 1. Các khái niệm trong CSDL

## a. Vùng dữ liệu nguồn – Database range

- ◆ Đây là vùng chứa dữ liệu gồm có ít nhất 2 dòng với dòng đầu tiên là tiêu đề của các cột, các dòng còn lại sẽ chứa dữ liệu, mỗi dòng được gọi là một mẫu tin (Record)

| Họ và tên | Chức vụ | Phụ cấp |
|-----------|---------|---------|
| Anh       | GD      | 500,000 |
| Bình      | NV      | 100,000 |
| Mai       | NV      | 100,000 |
| Hùng      | NV      | 100,000 |
| Minh      | PGD     | 300,000 |
| Chung     | PGD     | 300,000 |

## b. Vùng tiêu chuẩn – Criteria range

- ◆ Vùng này chứa các tiêu chuẩn (hay còn gọi là điều kiện) để giúp người dùng tìm kiếm, xoá hay rút trích những dòng dữ liệu thoả mãn tiêu chuẩn được đặt ra.
- ◆ Vùng này có ít nhất 2 dòng: dòng đầu chứa tiêu đề, các dòng còn lại chứa tiêu chuẩn

| Chức vụ |
|---------|
| GD      |

| Phụ cấp |
|---------|
| 300,000 |
| 100,000 |

## c. Vùng rút trích – Extract range

- ◆ Trong thao tác rút trích, nếu có yêu cầu đưa dữ liệu rút trích được vào vùng mới trên Worksheet thì vùng mới này gọi là vùng rút trích.
- ◆ Vùng có dòng đầu tiên chứa các tiêu đề của dữ liệu muốn rút trích, các dòng còn lại sẽ chứa dữ liệu rút trích được.

|    |                             |                |                |  |
|----|-----------------------------|----------------|----------------|--|
| 1  | <b>BẢNG LƯƠNG NHÂN VIÊN</b> |                |                |  |
| 2  | <b>Họ và tên</b>            | <b>Chức vụ</b> | <b>Phụ cấp</b> |  |
| 3  | Anh                         | GD             | 500,000        |  |
| 4  | Bình                        | NV             | 100,000        |  |
| 5  | Mai                         | NV             | 100,000        |  |
| 6  | Hùng                        | NV             | 100,000        |  |
| 7  | Minh                        | PGD            | 300,000        |  |
| 8  | Chung                       | PGD            | 300,000        |  |
| 9  |                             |                |                |  |
| 10 |                             | <b>Chức vụ</b> |                |  |
| 11 |                             | NV             |                |  |
| 12 |                             |                |                |  |
| 13 |                             |                |                |  |
| 14 | <b>Họ và tên</b>            | <b>Chức vụ</b> | <b>Phụ cấp</b> |  |
| 15 | Bình                        | NV             | 100,000        |  |
| 16 | Mai                         | NV             | 100,000        |  |
| 17 | Hùng                        | NV             | 100,000        |  |
| 18 |                             |                |                |  |

**Database  
range**

**Criteria  
range**

**Extract  
range**

## d. Các dạng vùng tiêu chuẩn

- ◆ **Tiêu chuẩn kiểu số:** Ô tiêu chuẩn có giá trị là số và có thể sử dụng các toán tử so sánh  $<$ ,  $>$ ,  $>=$ ,  $<=$ ,  $=$ ,  $<>$ .

|         |
|---------|
| Phụ cấp |
| 500,000 |

|           |
|-----------|
| Phụ cấp   |
| $>100000$ |



## d. Các dạng vùng tiêu chuẩn

- ◆ **Tiêu chuẩn kiểu chuỗi:** Ô tiêu chuẩn có giá trị là chuỗi và có thể sử dụng các ký tự:
  - \* (thay cho một nhóm ký tự bất kỳ)
  - ? (thay cho một ký tự bất kỳ).


|         |
|---------|
| Chức vụ |
| NV      |

|           |
|-----------|
| Họ và tên |
| *nh       |

## d. Các dạng vùng tiêu chuẩn

- ◆ **Tiêu chuẩn công thức:** Ô tiêu chuẩn dạng là một công thức. Khi dùng tiêu chuẩn công thức phải chú ý hai điểm sau:
  - Tiêu đề của vùng tiêu chuẩn phải khác với tất cả các tiêu đề trên vùng dữ liệu nguồn
  - Trong ô tiêu chuẩn, công thức nhập vào phải lấy địa chỉ của mẫu tin đầu tiên tại cột tương ứng trên vùng dữ liệu làm giá trị so sánh

- ◆ Ví dụ: xây dựng vùng tiêu chuẩn để lọc ra các nhân viên và bảo vệ trong công ty:

|    | A                           | B                | C                                                                                  | D                            |
|----|-----------------------------|------------------|------------------------------------------------------------------------------------|------------------------------|
| 1  | <b>BẢNG LƯƠNG NHÂN VIÊN</b> |                  |                                                                                    |                              |
| 2  | <b>Họ và tên</b>            | <b>Chức vụ</b>   | <b>Phụ cấp</b>                                                                     |                              |
| 3  | Anh                         | GD               | 500,000                                                                            |                              |
| 4  | Bình                        | NV               | 100,000                                                                            |                              |
| 5  | Mai                         | NV               | 100,000                                                                            |                              |
| 6  | Hùng                        | NV               | 100,000                                                                            |                              |
| 7  | Minh                        | PGD              | 300,000                                                                            |                              |
| 8  | Chung                       | PGD              | 300,000                                                                            |                              |
| 9  |                             |                  |                                                                                    |                              |
| 10 |                             | <b>Điều kiện</b> |                                                                                    | <b>Điều kiện</b>             |
| 11 |                             | <b>FALSE</b>     |  | <b>=AND(B3="NV",B3="BV")</b> |
| 12 |                             |                  |                                                                                    |                              |
| 13 |                             |                  |                                                                                    |                              |
| 14 | <b>Họ và tên</b>            | <b>Chức vụ</b>   | <b>Phụ cấp</b>                                                                     |                              |
| 15 | Bình                        | NV               | 100,000                                                                            |                              |
| 16 | Mai                         | NV               | 100,000                                                                            |                              |
| 17 | Hùng                        | NV               | 100,000                                                                            |                              |
| 18 |                             |                  |                                                                                    |                              |

## d. Các dạng vùng tiêu chuẩn

### ◆ Liên kết tiêu chuẩn:

- Ta có thể liên kết các tiêu chuẩn lại với nhau, khi đó:
- Những tiêu chuẩn lọc trên cùng một dòng được hiểu là liên kết AND, khác dòng được hiểu là liên kết bằng toán tử OR.

| Chức vụ |
|---------|
| NV      |
| BV      |

| Chức vụ | Họ và tên |
|---------|-----------|
| NV      | Mai       |

## 2. Các thao tác trên cơ sở dữ liệu

### a. Sắp xếp dữ liệu

- ◆ Để sắp xếp dữ liệu ta thực hiện các bước sau:
  - Chọn vùng dữ liệu cần sắp xếp
  - Từ thực đơn **Data/Sort**
  - Thiết lập các thông số trong hộp thoại sort:

Chọn cột  
sắp xếp

The image shows a 'Sort' dialog box with a blue title bar and standard window controls (help, close). The dialog is divided into three sections for sorting criteria. The first section is labeled 'Sort by' and has a dropdown menu currently showing 'Họ và tên'. To its right are two radio buttons: 'Ascending' (selected) and 'Descending'. The second section is labeled 'Then by' and has an empty dropdown menu. To its right are two radio buttons: 'Ascending' (selected) and 'Descending'. The third section is also labeled 'Then by' and has an empty dropdown menu. To its right are two radio buttons: 'Ascending' (selected) and 'Descending'. At the bottom, there is a section labeled 'My list has' with two radio buttons: 'Header row' (selected) and 'No header row'. At the very bottom are three buttons: 'Options...', 'OK', and 'Cancel'. Three callout boxes with green backgrounds and blue borders point to specific elements: one points to the 'Họ và tên' dropdown, one points to the 'Ascending' radio button, and one points to the 'Descending' radio button.

Sort

Sort by

Ascending

Descending

Then by

Ascending

Descending

Then by

Ascending

Descending

My list has

Header row  No header row

Options... OK Cancel

Sắp tăng

Sắp giảm

Lựa chọn khi  
vùng chọn sắp  
xếp có tiêu đề

## **b. Lọc dữ liệu tự động – Auto filter**

- ◆ Để sử dụng chức năng Auto Filter, ta thực hiện như sau:
  - Chọn vùng dữ liệu cần lọc
  - Từ thực đơn Data/Filter/AutoFilter
  - Chọn nút mũi tên tại cột cần lọc, trên dòng tiêu đề
  - Chọn một giá trị trong danh sách, dữ liệu sẽ lọc theo giá trị đó.
  - Muốn lọc dữ liệu theo các tiêu chuẩn khác chọn Custom

**Ví dụ: Lọc các nhân viên có mức phụ cấp là 100000.**

|   | A                           | B                | C                | D |
|---|-----------------------------|------------------|------------------|---|
| 1 | <b>BẢNG LƯƠNG NHÂN VIÊN</b> |                  |                  |   |
| 2 | <b>Họ và tên</b> ▼          | <b>Chức vụ</b> ▼ | <b>Phụ cấp</b> ▼ |   |
| 3 | Anh                         | GD               | (All)            |   |
| 4 | Bình                        | NV               | (Top 10...)      |   |
| 5 | Mai                         | NV               | (Custom...)      |   |
| 6 | Hùng                        | NV               | 100,000          |   |
| 7 | Minh                        | PGD              | 300,000          |   |
| 8 | Chung                       | PGD              | 300,000          |   |
| 9 |                             |                  |                  |   |



## c. Lọc dữ liệu cao cấp – Advanced Filter

- ◆ Các bước thực hiện:
  - Từ thực đơn Data/Filter/Advanced Filter
  - Đưa con trỏ vào ô List range, sau đó quét chọn vùng dữ liệu nguồn.
  - Đưa con trỏ vào ô Criteria range, sau đó quét chọn vùng tiêu chuẩn
  - Chọn mục Copy to another location để xuất kết quả ra vị trí mới, sau đó chọn vùng rút trích
  - Nhấn OK để thực hiện

## Ví dụ: Lọc ra các nhân viên và bảo vệ trong công ty.

|    | A                           | B              | C              |
|----|-----------------------------|----------------|----------------|
| 1  | <b>BẢNG LƯƠNG NHÂN VIÊN</b> |                |                |
| 2  | <b>Họ và tên</b>            | <b>Chức vụ</b> | <b>Phụ cấp</b> |
| 3  | Anh                         | GD             | 500,000        |
| 4  | Bình                        | NV             | 100,000        |
| 5  | Mai                         | NV             | 100,000        |
| 6  | Hùng                        | NV             | 100,000        |
| 7  | Minh                        | PGD            | 300,000        |
| 8  | Chung                       | PGD            | 300,000        |
| 9  |                             |                |                |
| 10 |                             | <b>Chức vụ</b> |                |
| 11 |                             | <b>NV</b>      |                |
| 12 |                             | <b>BV</b>      |                |
| 13 |                             |                |                |
| 14 |                             |                |                |
| 15 | <b>Họ và tên</b>            | <b>Chức vụ</b> | <b>Phụ cấp</b> |
| 16 | Bình                        | NV             | 100,000        |
| 17 | Mai                         | NV             | 100,000        |
| 18 | Hùng                        | NV             | 100,000        |
| 19 |                             |                |                |

**Advanced Filter** [?] [X]

Action

Filter the list, in-place

Copy to another location

List range:  [icon]

Criteria range:  [icon]

Copy to:  [icon]

Unique records only

OK Cancel

# 3. Các hàm cơ sở dữ liệu

- ◆ Hàm DSUM
- ◆ Hàm DCOUNT
- ◆ Hàm DAVERAGE
- ◆ Hàm DMAX
- ◆ Hàm DMIN

## a. Hàm DSUM

- ◆ Cú pháp:

**DSUM (<vùng dữ liệu><tên cột>, <vùng tiêu chuẩn>)**

- ◆ Ý nghĩa:

- Hàm trả về tổng các ô trên tên cột thỏa điều kiện của vùng tiêu chuẩn.

- ◆ Ví dụ:

## Ví dụ: Tính tổng phụ cấp của các nhân viên trong công ty

|    | A                                      | B              | C                              | D |  |
|----|----------------------------------------|----------------|--------------------------------|---|--|
| 1  | <b>BẢNG LƯƠNG NHÂN VIÊN</b>            |                |                                |   |  |
| 2  | <b>Họ và tên</b>                       | <b>Chức vụ</b> | <b>Phụ cấp</b>                 |   |  |
| 3  | Anh                                    | GD             | 500,000                        |   |  |
| 4  | Bình                                   | NV             | 100,000                        |   |  |
| 5  | Mai                                    | NV             | 100,000                        |   |  |
| 6  | Hùng                                   | NV             | 100,000                        |   |  |
| 7  | Minh                                   | PGD            | 300,000                        |   |  |
| 8  | Chung                                  | PGD            | 300,000                        |   |  |
| 9  |                                        |                |                                |   |  |
| 10 |                                        | <b>Chức vụ</b> |                                |   |  |
| 11 |                                        | NV             |                                |   |  |
| 12 |                                        |                |                                |   |  |
| 13 | <b>BẢNG THỐNG KÊ</b>                   |                |                                |   |  |
| 14 | <b>Tổng phụ cấp của các nhân viên.</b> |                | <b>=DSUM(A2:C8,C2,B10:B11)</b> |   |  |
| 15 |                                        |                |                                |   |  |

## b. Hàm DCOUNT

- ◆ Cú pháp:

**DCOUNT (<vùng dữ liệu><tên cột>, <vùng tiêu chuẩn>)**

- ◆ Ý nghĩa:

- Hàm đếm các ô kiểu số trên tên cột thỏa điều kiện của vùng tiêu chuẩn.

- ◆ Ví dụ:

## Ví dụ: Đếm tổng số các nhân viên trong công ty

|    | A                             | B                                | C              | D |
|----|-------------------------------|----------------------------------|----------------|---|
| 1  | <b>BẢNG LƯƠNG NHÂN VIÊN</b>   |                                  |                |   |
| 2  | <b>Họ và tên</b>              | <b>Chức vụ</b>                   | <b>Phụ cấp</b> |   |
| 3  | Anh                           | GD                               | 500,000        |   |
| 4  | Bình                          | NV                               | 100,000        |   |
| 5  | Mai                           | NV                               | 100,000        |   |
| 6  | Hùng                          | NV                               | 100,000        |   |
| 7  | Minh                          | PGD                              | 300,000        |   |
| 8  | Chung                         | PGD                              | 300,000        |   |
| 9  |                               |                                  |                |   |
| 10 |                               | <b>Chức vụ</b>                   |                |   |
| 11 |                               | NV                               |                |   |
| 12 |                               |                                  |                |   |
| 13 | <b>BẢNG THỐNG KÊ</b>          |                                  |                |   |
| 14 | <b>Tổng số các nhân viên:</b> | <b>=DCOUNT(A2:C8,C2,B10:B11)</b> |                |   |
| 15 |                               |                                  |                |   |

## c. Hàm D AVERAGE

- ◆ Cú pháp:

**DAVERAGE (<vùng dữ liệu><tên cột>, <vùng tiêu chuẩn>)**

- ◆ Ý nghĩa:

- Hàm đếm các ô kiểu số trên tên cột thỏa điều kiện của vùng tiêu chuẩn.

- ◆ Ví dụ:



## Ví dụ: Tính phụ cấp trung bình của Nhân viên và PGĐ

|    | A                                     | B       | C                           | D | E |
|----|---------------------------------------|---------|-----------------------------|---|---|
| 1  | BẢNG LƯƠNG NHÂN VIÊN                  |         |                             |   |   |
| 2  | Họ và tên                             | Chức vụ | Phụ cấp                     |   |   |
| 3  | Anh                                   | GD      | 500,000                     |   |   |
| 4  | Bình                                  | NV      | 100,000                     |   |   |
| 5  | Mai                                   | NV      | 100,000                     |   |   |
| 6  | Hùng                                  | NV      | 100,000                     |   |   |
| 7  | Minh                                  | PGĐ     | 300,000                     |   |   |
| 8  | Chung                                 | PGĐ     | 300,000                     |   |   |
| 9  |                                       |         |                             |   |   |
| 10 |                                       | Chức vụ |                             |   |   |
| 11 |                                       | NV      |                             |   |   |
| 12 |                                       | PGĐ     |                             |   |   |
| 13 |                                       |         |                             |   |   |
| 14 | BẢNG THỐNG KÊ                         |         |                             |   |   |
| 15 | Mức phụ cấp trung bình của NV và PGĐ. |         | =DAVERAGE(A2:C8,C2,B10:B12) |   |   |
| 16 |                                       |         |                             |   |   |

## d. Hàm DMAX

- ◆ Cú pháp:

**DMAX (<vùng dữ liệu><tên cột>, <vùng tiêu chuẩn>)**

- ◆ Ý nghĩa:

- Hàm trả về giá trị lớn nhất của các ô trên tên cột thỏa điều kiện của vùng tiêu chuẩn.

- ◆ Ví dụ:

## Ví dụ: Tính mức lương cấp cao nhất của PGĐ

|    | A                           | B       | C                       | D |  |
|----|-----------------------------|---------|-------------------------|---|--|
| 1  | BẢNG LƯƠNG NHÂN VIÊN        |         |                         |   |  |
| 2  | Họ và tên                   | Chức vụ | Lương                   |   |  |
| 3  | Anh                         | GD      | 5,000,000               |   |  |
| 4  | Bình                        | NV      | 1,500,000               |   |  |
| 5  | Mai                         | NV      | 1,500,000               |   |  |
| 6  | Hùng                        | NV      | 1,500,000               |   |  |
| 7  | Minh                        | PGĐ     | 3,500,000               |   |  |
| 8  | Chung                       | PGĐ     | 3,000,000               |   |  |
| 9  |                             |         |                         |   |  |
| 10 |                             | Chức vụ |                         |   |  |
| 11 |                             | PGĐ     |                         |   |  |
| 12 |                             |         |                         |   |  |
| 13 | BẢNG THỐNG KÊ               |         |                         |   |  |
| 14 | Mức lương cao nhất của PGĐ: |         | =DMAX(A2:C8,C2,B10:B11) |   |  |
| 15 |                             |         |                         |   |  |

## d. Hàm DMIN

- ◆ Cú pháp:

**DMIN (<vùng dữ liệu><tên cột>, <vùng tiêu chuẩn>)**

- ◆ Ý nghĩa:

- Hàm trả về giá trị nhỏ nhất của các ô trên tên cột thỏa điều kiện của vùng tiêu chuẩn.

- ◆ Ví dụ:

## Ví dụ: Tính mức lương cấp thấp nhất của PGĐ

|    | A                           | B       | C                       | D |  |
|----|-----------------------------|---------|-------------------------|---|--|
| 1  | BẢNG LƯƠNG NHÂN VIÊN        |         |                         |   |  |
| 2  | Họ và tên                   | Chức vụ | Lương                   |   |  |
| 3  | Anh                         | GD      | 5,000,000               |   |  |
| 4  | Bình                        | NV      | 1,500,000               |   |  |
| 5  | Mai                         | NV      | 1,500,000               |   |  |
| 6  | Hùng                        | NV      | 1,500,000               |   |  |
| 7  | Minh                        | PGĐ     | 3,500,000               |   |  |
| 8  | Chung                       | PGĐ     | 3,000,000               |   |  |
| 9  |                             |         |                         |   |  |
| 10 |                             | Chức vụ |                         |   |  |
| 11 |                             | PGĐ     |                         |   |  |
| 12 |                             |         |                         |   |  |
| 13 | BẢNG THỐNG KÊ               |         |                         |   |  |
| 14 | Mức lương cao nhất của PGĐ: |         | =DMIN(A2:C8,C2,B10:B11) |   |  |
| 15 |                             |         |                         |   |  |



# Kết thúc

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG  
KHOA CÔNG NGHỆ THÔNG TIN & TOÁN ỨNG DỤNG**



**MÔN HỌC  
TIN HỌC ĐẠI CƯƠNG**

**GIẢNG VIÊN: LÊ THỊ NGỌC THẢO**

# TIN HỌC ĐẠI CƯƠNG

GV: Lê Thị Ngọc Thảo



# Mục tiêu môn học

---

- Cung cấp cho sinh viên các kiến thức cơ bản về:
  - Máy tính điện tử
  - Biểu diễn thông tin trong máy tính điện tử
- Giúp sinh viên nắm vững khái niệm giải thuật và các phương pháp biểu diễn, thiết kế giải thuật.

# Đề cương môn học

---

- Chương 1: Thông tin và xử lý thông tin
- Chương 2: Máy tính điện tử
- Chương 3: Biểu diễn thông tin trong máy tính điện tử
- Chương 4: Chương trình và giải thuật
- Chương 5: Các ngôn ngữ lập trình

# Phương thức tiến hành

---

- Lý thuyết: 30 tiết
- Bài tập: 15 tiết
- Tự học: 90 tiết
- Môn học được phân bổ trong 15 tuần

# Tài liệu tham khảo

---

- Tóm tắt bài giảng (slide)
- Giáo trình Tin Học Đại Cương I,II – Hoàng Kiếm – NXB Giáo Dục, 1997
- Bài Giảng Tin học Đại Cương A1 – Nguyễn Đức Thắng – ĐHQG Tp.HCM

# Phương thức đánh giá

---

- Điểm kiểm tra 10%
- Điểm thi giữa học kỳ 20%
- Kết quả thi cuối học kỳ 70%

# Chương 1: Thông tin & Xử lý thông tin

# Chương 1 - Thông tin & Xử lý thông tin

---

- Thông tin
- Xử lý thông tin
- Tin học

# Thông tin (Information)

---

- Là một khái niệm trừu tượng
- Tin tức
- Mang lại sự hiểu biết, nhận thức
  - Ví dụ: Thông tin về khí tượng
- Được phát sinh, được lưu trữ, được truyền, được tìm kiếm, được sao chép, được xử lý, nhân bản.
- Có thể biến dạng, sai lệch hoặc bị phá hủy.
  - Ví dụ: thông tin di truyền



# Phân biệt Thông tin & Dữ liệu

---

- Dữ liệu (data) là biểu diễn của thông tin và được thể hiện bằng các tín hiệu (signal) vật lý.
- Thông tin chứa đựng ý nghĩa.
- Dữ liệu là các dữ kiện không có cấu trúc và không có ý nghĩa rõ ràng nếu nó không được tổ chức và xử lý.
- Cùng một thông tin, có thể được biểu diễn bằng những dữ liệu khác nhau.
  - Ví dụ: biểu diễn một đơn vị (1 & I)
- Mỗi dữ liệu lại có thể được thể hiện bằng những ký hiệu vật lý khác nhau.
  - Ví dụ: ký hiệu I (Tôi (tiếng Anh) & giá trị một)

# Xử lý thông tin

---

- Con người tiếp nhận thông tin → Xử lý tạo ra thông tin mới, có ích hơn.
- Ví dụ:
  - Xử lý các thông tin trong kinh doanh
  - Xử lý các thông tin trong lĩnh vực điều tra

# Xử lý thông tin trong máy tính

---

- Trong máy tính, các thông tin được biểu diễn bằng hệ đếm nhị phân.
- Tuy chỉ dùng 2 ký số là 0 và 1 mà ta gọi là bit nhưng hệ nhị phân đã giúp máy tính biểu diễn - xử lý được trên hầu hết các loại thông tin mà con người hiện đang sử dụng như văn bản, hình ảnh, âm thanh, video,...

# Tin học

---

- Tin học: là ngành khoa học về thông tin, các cách thức xử lý thông tin và các kỹ thuật xây dựng hệ thống thông tin.
- Tin học nghiên cứu cấu trúc, hoạt động, và tương tác giữa các hệ thống (tự nhiên và nhân tạo) lưu trữ, xử lý và dẫn truyền thông tin.

# Công nghệ thông tin

---

- Information Technology
- Công nghệ thông tin: là ngành nghiên cứu, thiết kế, phát triển, xây dựng, hỗ trợ và quản lý các hệ thống thông tin với sự trợ giúp của máy tính.

# Một số từ khóa tiếng Anh

---

- Computer, Computer science
- Data
- Enginneering, Software engineering
- Information, Information system
- Technique
- Technology, Information technology
- Knowledge

# CHƯƠNG 2: MÁY TÍNH ĐIỆN TỬ

## Chương 2: Máy tính điện tử

---

- Những nguyên lý máy tính cơ bản
- Cấu trúc tổng quát máy tính điện tử
- Sự phát triển của máy tính điện tử



# Những nguyên lý máy tính cơ bản

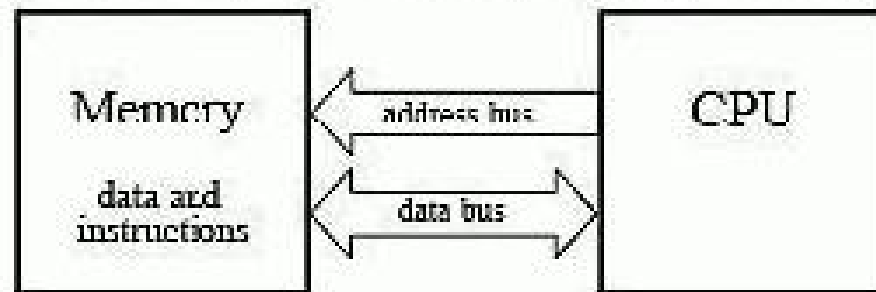
---

- **Kiến trúc Von Neumann**
- **Kiến trúc Harvard**
- **Kiến trúc SHARC**

# Kiến trúc Von Neumann

- Gồm có:
  - Bộ nhớ & Bus để chuyển dữ liệu vào & ra đơn vị xử lý trung tâm (CPU)
  - CPU chỉ có thể đọc lệnh, hay đọc/ghi dữ liệu từ bộ nhớ.
- Cả hai quá trình tương tác với lệnh hoặc với dữ liệu không thể thực hiện cùng lúc. → thực hiện tuần tự

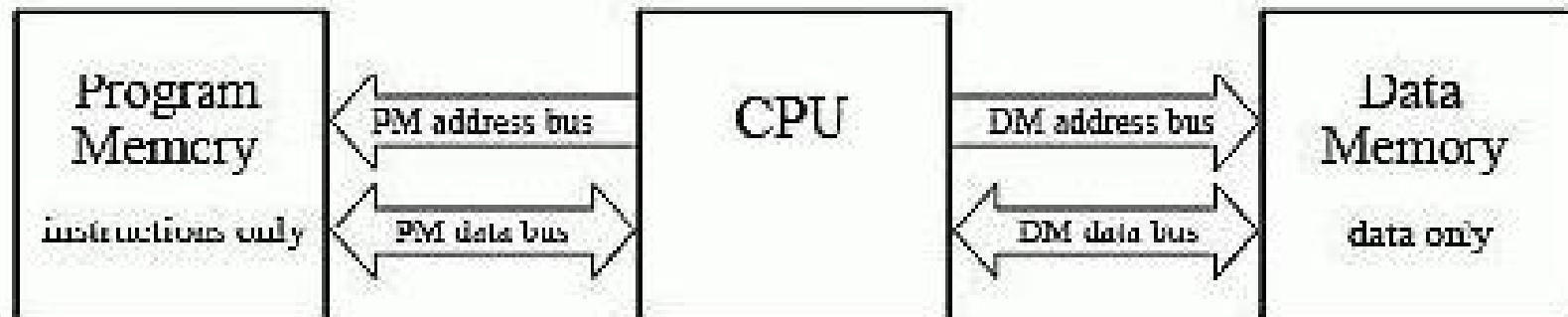
a. Von Neumann Architecture (*single memory*)



# Kiến trúc Harvard

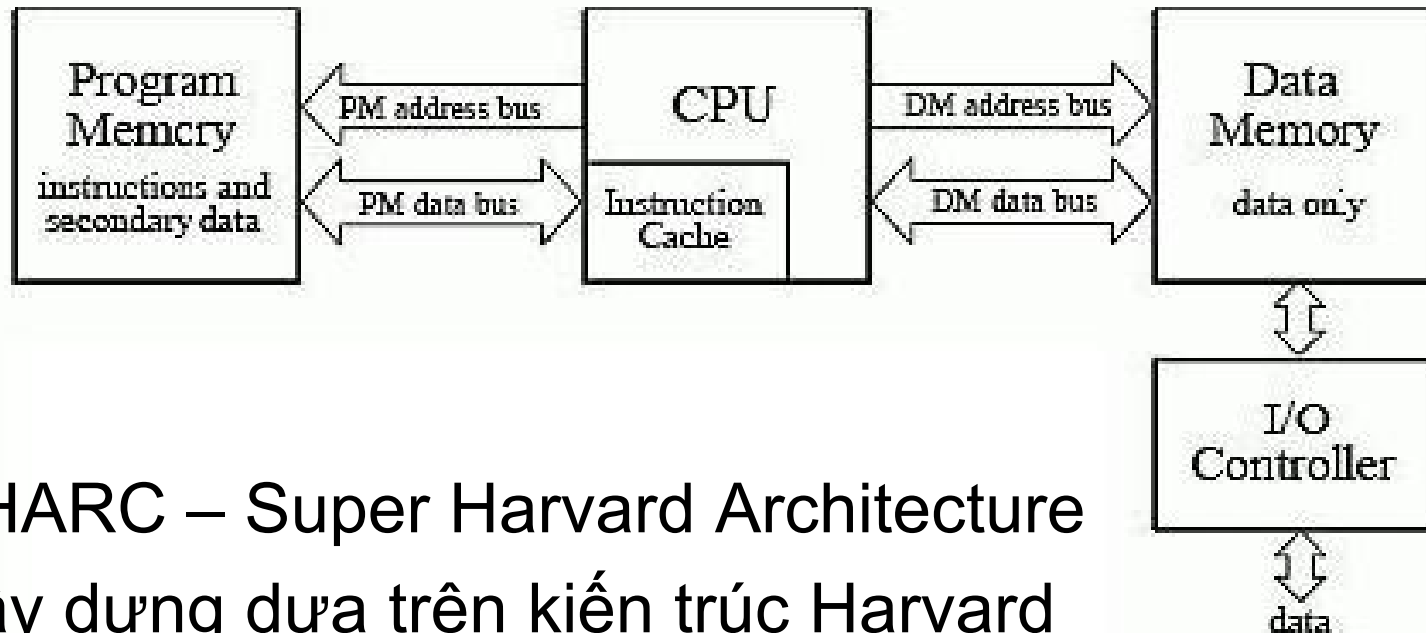
- Được nghiên cứu tại Harvard, dưới sự lãnh đạo của Howard Aiken (1900-1973)
- Bộ nhớ dữ liệu và bộ nhớ chương trình sử dụng các bus riêng rẽ → thực hiện song song → cải thiện tốc độ.

## b. Harvard Architecture ( *dual memory* )



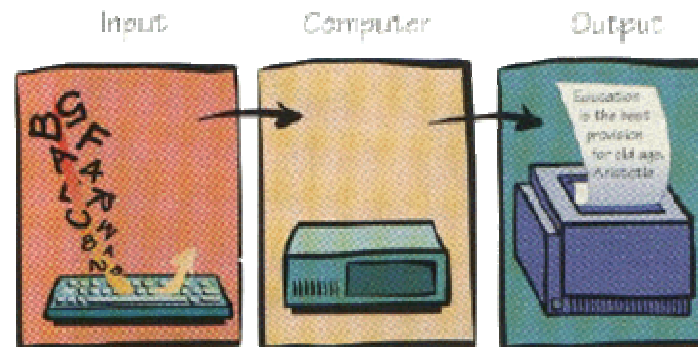
# Kiến trúc SHARC

c. Super Harvard Architecture (*dual memory, instruction cache, I/O controller* )



- SHARC – Super Harvard Architecture
- Xây dựng dựa trên kiến trúc Harvard
- Thêm vào những điểm đặc trưng để cải thiện thông lượng dữ liệu (bộ nhớ đệm chỉ lệnh & điều khiển vào/ra.)

# Quá trình xử lý thông tin trên máy tính

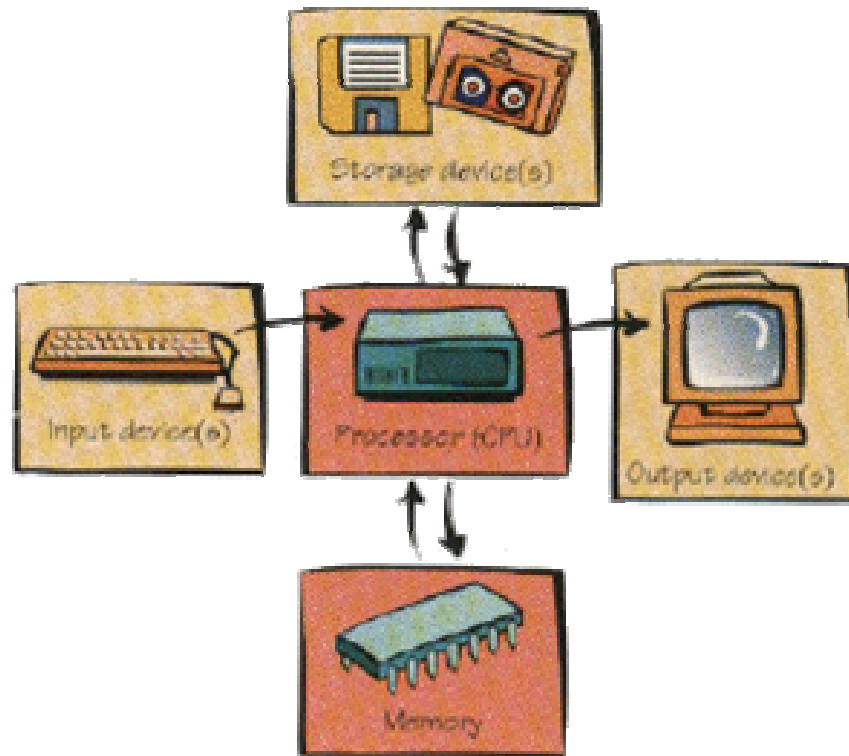


- **Nhận thông tin:** từ thế giới bên ngoài vào máy tính
- **Xử lý thông tin:** biến đổi, phân tích, tổng hợp, tra cứu... những thông tin ban đầu → thông tin mong muốn
- **Xuất thông tin:** đưa kết quả ra thế giới bên ngoài
- **Lưu trữ thông tin:** lưu lại kết quả để sử dụng trong các lần sau

# Các thành phần chính trong máy tính

- Gồm 4 thành phần chính:

- *Thiết bị nhập*
  - Bàn phím, chuột
- *Thiết bị xử lý*
  - Bộ não
- *Thiết bị xuất*
  - Màn hình, loa
- *Thiết bị lưu trữ*
  - Sơ cấp: bộ nhớ
  - Thứ cấp: đĩa cứng, CD



# Đơn vị xử lý trung tâm (1)

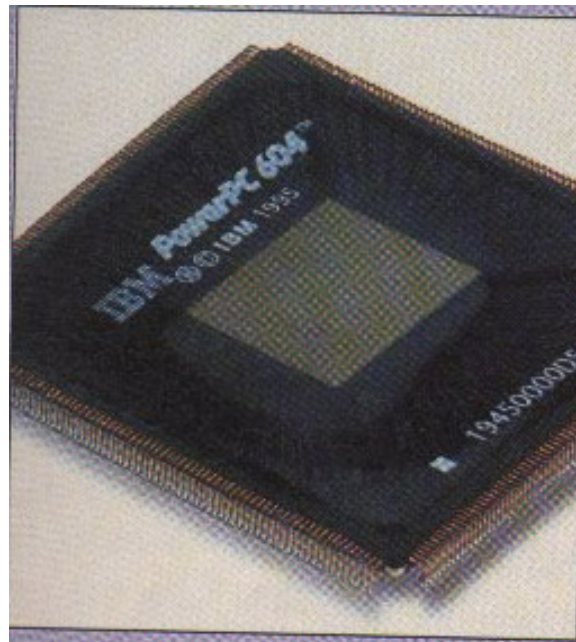
---

- CPU – central processing unit
  - Mạch xử lý dữ liệu
    - theo chương trình thiết lập trước
    - tích hợp phức tạp hàng triệu transistor trên bảng mạch
  - Có thể thi hành hàng triệu lệnh mỗi giây
  - Bao gồm nhiều thành phần phức tạp với các chức năng khác nhau

## Đơn vị xử lý trung tâm (2)

---

- Công việc chính của CPU là thi hành các mã lệnh của chương trình





# Đơn vị xử lý trung tâm (3)

---

- Thành phần căn bản bên trong CPU
  - Arithmetic Logic Unit (ALU)
    - **Gồm một số thanh ghi – register (32, 64 bit)**
    - **Thực hiện các lệnh của đơn vị điều khiển và xử lý tín hiệu:**
      - **Các phép tính số học đơn giản (cộng, trừ, nhân, chia số nguyên)**
      - **Phép tính luận lý đối với dữ liệu (so sánh lớn hơn, nhỏ hơn, ...).**

# Đơn vị xử lý trung tâm (4)

---

- Thành phần căn bản bên trong CPU(tt)
  - Đơn vị nạp lệnh - Prectch unit
  - Đơn vị giải mã - Decode unit
  - Đơn vị nối ghép đường truyền
    - Bus Interface Unit
  - Đơn vị điều khiển - control unit

# Đơn vị xử lý trung tâm (5)

---

- Thành phần căn bản bên trong CPU (tt)
  - Thanh ghi - register
  - Bộ nhớ ẩn - cache memory
    - Primary cache: Level 1
    - Secondary cache: Level 2
- Mạch xung nhịp hệ thống - system clock
  - Đồng bộ các xử lý trong và ngoài CPU
  - Tốc độ xung nhịp: triệu đơn vị mỗi giây Mhz.

# Bộ nhớ máy tính (1)

---

- CPU chỉ có khả năng giải quyết một ít trong phần dữ liệu
- Cần lưu giữ lại sẵn sàng dữ liệu được đọc cho CPU xử lý:
  - Bộ nhớ chính
  - ROM & RAM

## Bộ nhớ máy tính (2)

---

- Read Only Memory - ROM
  - Chỉ đọc
  - Cố định - *nonvolatile memory*: duy trì nội dung nhớ khi không có nguồn điện
  - Chứa chương trình BIOS không thay đổi.
  - Máy trò chơi điện tử (game box)
    - khe cắm ROM

## Bộ nhớ máy tính (3)

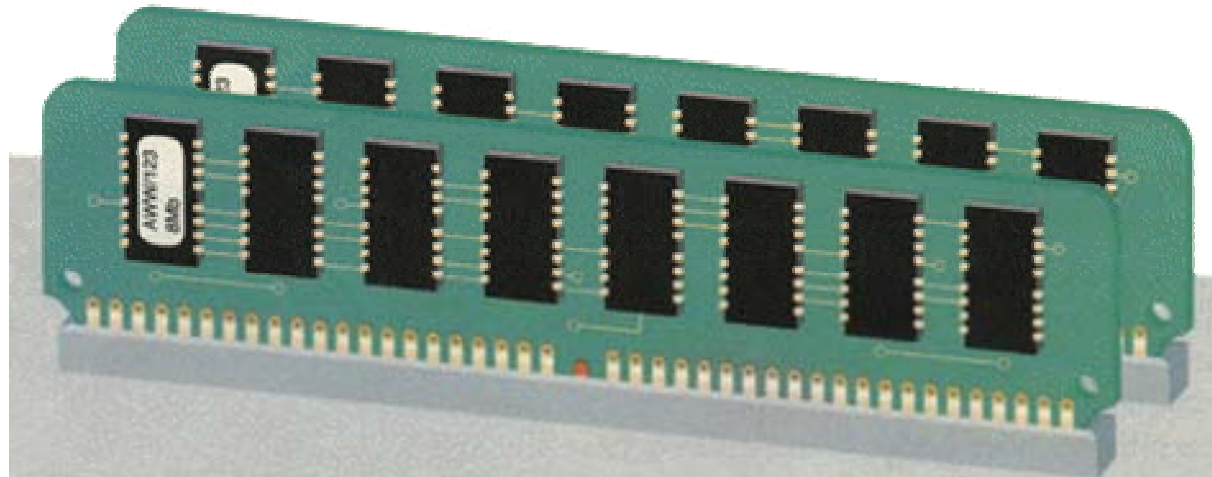
---

- *RAM - Random Access Memory*
  - Bộ nhớ truy cập ngẫu nhiên
  - Gồm nhiều chip RAM
  - Vị trí lưu trữ trong RAM có thể truy cập trực tiếp
    - Thao tác truy tìm và cất trữ có thể thực hiện rất nhanh.
  - không cố định - volatile memory
  - SRAM - RAM tĩnh, DRAM - RAM động
  - 8405010

## Bộ nhớ máy tính (4)

---

- Phải luôn có nguồn nuôi để lưu trữ nội dung thông tin trên RAM
- Mất điện là mất tất cả.



# Bộ nhớ máy tính (5)

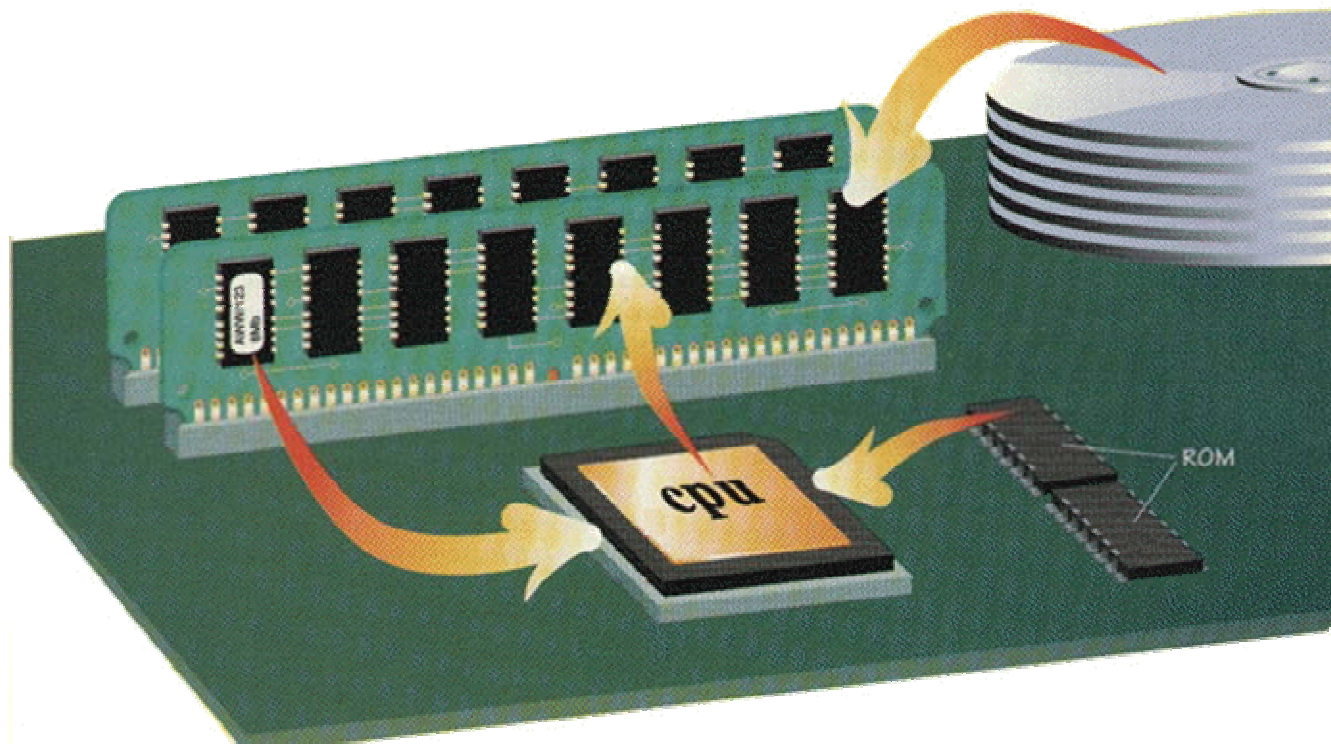
---

- Hoạt động khởi động máy tính:
  - CPU tự động (đã qui định trước) đọc thông tin lưu trong ROM và thi hành, chương trình BIOS - **hệ thống xuất nhập cơ sở**
  - Sau đó đọc thông tin trên đĩa khởi động và nạp các thông tin hệ điều hành trên đĩa vào bộ nhớ RAM.
  - CPU có thể thực hiện các tác vụ từ các thông tin lưu trên RAM



# Hoạt động khởi động máy tính

---



# Thiết bị nhập (1)

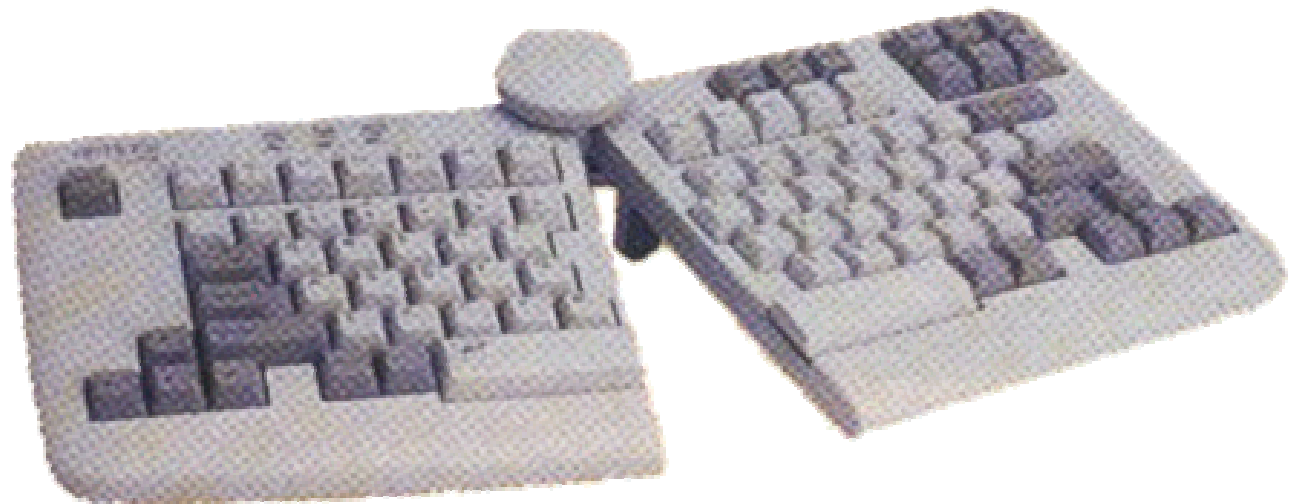
---

- Bàn phím: Key board
- Thiết bị chỉ điểm: Pointing Device
- Thiết bị đọc
- Thiết bị số hóa

# Thiết bị nhập (2)

---

- Bàn phím:
  - Ký tự
  - Điều khiển
  - Cấu trúc QWERT



# Thiết bị nhập (3)

---

- Pointing Device
  - Chuột: Mouse
    - Ball, optical, trackball
  - Phiến nhấn: Touch Pad
  - Cần điều khiển: Joy Stick
  - Màn hình cảm ứng: Touch Screen
  - Bàn vẽ: Graphic Table

# Pointing Device

---



# Thiết bị nhập (4)

---

- Thiết bị đọc:
  - Thiết bị đọc đánh dấu quang học
    - Optical-mark reader
  - Thiết bị đọc mã vạch - Barcode reader
    - dùng ánh sáng để đọc mã sản phẩm, mã kiểm tra
  - Cây đũa thần - wand reader
  - Cây viết máy tính
    - pen-based computer

# Thiết bị nhập (5)

---

- Thiết bị số hóa thế giới thực:
  - Máy quét ảnh – scanner
  - Máy ảnh số – digital camera
  - Máy quay phim số – digital video camera
  - Thiết bị âm thanh số hóa – Audio digitalizer
  - Thiết bị cảm ứng

# Thiết bị số hóa thế giới thực





# Thiết bị xuất (1)

---

- Xuất ra màn hình
- Xuất ra giấy
- Xuất ra âm thanh
- Nói mạng

# Thiết bị xuất (2)

---

- Xuất ra màn hình
  - Màn hình máy tính (Video Monitor)
    - Bóng đèn tia điện tử
      - cathode CRT (cathode ray cathode)
    - Màn hình tinh thể lỏng LCD
      - liquid crystal display
    - Điểm ảnh (pixel) – Độ phân giải (resolution)
      - 72 dpi (dots per inch) , 800x600
    - Màu – (Color depth)

# Thiết bị xuất (3)

---

- Xuất ra màn hình
  - Bộ điều hợp hiển thị (AGP)
    - Video adapter - display adapter
    - Card màn hình - display card, video card
    - Đường dẫn dữ liệu: 4X: 32bit, 8X:64bit
    - Bộ nhớ màn hình : Video RAM

# Thiết bị xuất (4)

---

- Xuất ra giấy
  - Máy in gõ - impact printer
    - In theo dòng hay theo ma trận điểm.
    - Tốc độ in chậm, ồn ào, độ phân giải thấp
    - Ấn các kim qua lớp băng mực theo tín hiệu điều khiển
    - Số đầu kim qui định độ phân giải đạt
    - In trên khổ giấy lớn mà giá máy rẻ
    - Nhân thành nhiều bản bằng giấy than do sự gõ truyền lực
  - Máy in không gõ - nonimpact printer
    - Kỹ thuật hiện đại khác

## Thiết bị xuất (5)

---

- Máy in nhiệt
  - Dùng các xung điện từ làm cho đầu kim ma trận điểm nóng lên và nguội đi rất nhanh → làm đổi màu các điểm trên loại giấy đặc biệt
  - Tốc độ máy in tương đối nhanh và ít tốn điện
  - Dùng giấy in nhiệt - thermal paper
  - Công nghệ máy in truyền mực bằng nhiệt - thermal fusion printer
  - Phổ biến trong máy in xách tay

## Thiết bị xuất (6)

---

- Máy in phun mực – inkjet printer
  - Dùng tinh thể áp điện (máy bơm)
    - Đẩy mực ra khỏi ống và hút thêm mực khác vào
  - Máy in phun bong bóng - bubble jet printer
    - Dùng phần tử nung nóng
  - Hạn chế bởi tốc độ in
  - Ưu điểm dùng điện áp thấp từ 24V đến 50V
  - Dùng với mọi loại giấy, độ nét và độ mịn cao
  - Chi phí in cao

# Thiết bị xuất (7)

---

- Máy in laser :
  - Công nghệ in tĩnh điện
    - Electrostatic - ES
    - Điện tích tĩnh điện làm chảy mực lên giấy
    - Hệ thống tạo hình
      - image formation system (IFS)
  - Máy in di-ốt phát quang
    - Light emitting diode printer
  - Máy in cửa sập tinh thể lỏng
    - Liquid crystal shutter printer.

# Thiết bị xuất (8)

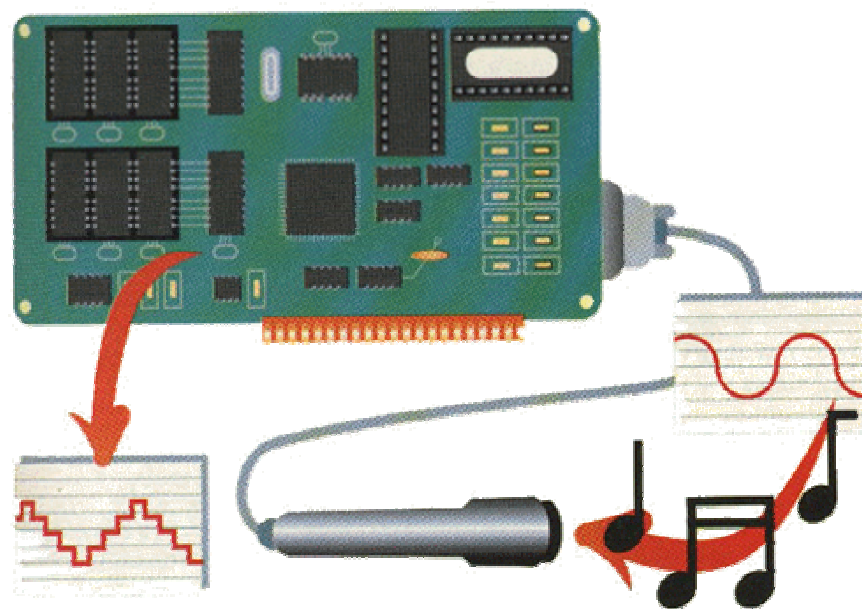
---

- Máy in màu
  - Kiểu RGB - tức là Red-Green-Blue
    - Cách tạo màu của tivi và màn hình máy tính
  - Kiểu HSB - Hue-Saturation-Brightness
    - Sắc màu, lượng màu, độ sáng
  - Kiểu CMYK Cyan, Magenta, Yellow, black
    - In ấn chế bản: C -cyan là xanh, M - magenta là đỏ, Y - yellow là vàng, K - black là đen.
  - Hệ thống hợp màu pantone
    - **Pantone color matching system – CMYK**



# Thiết bị xuất (9)

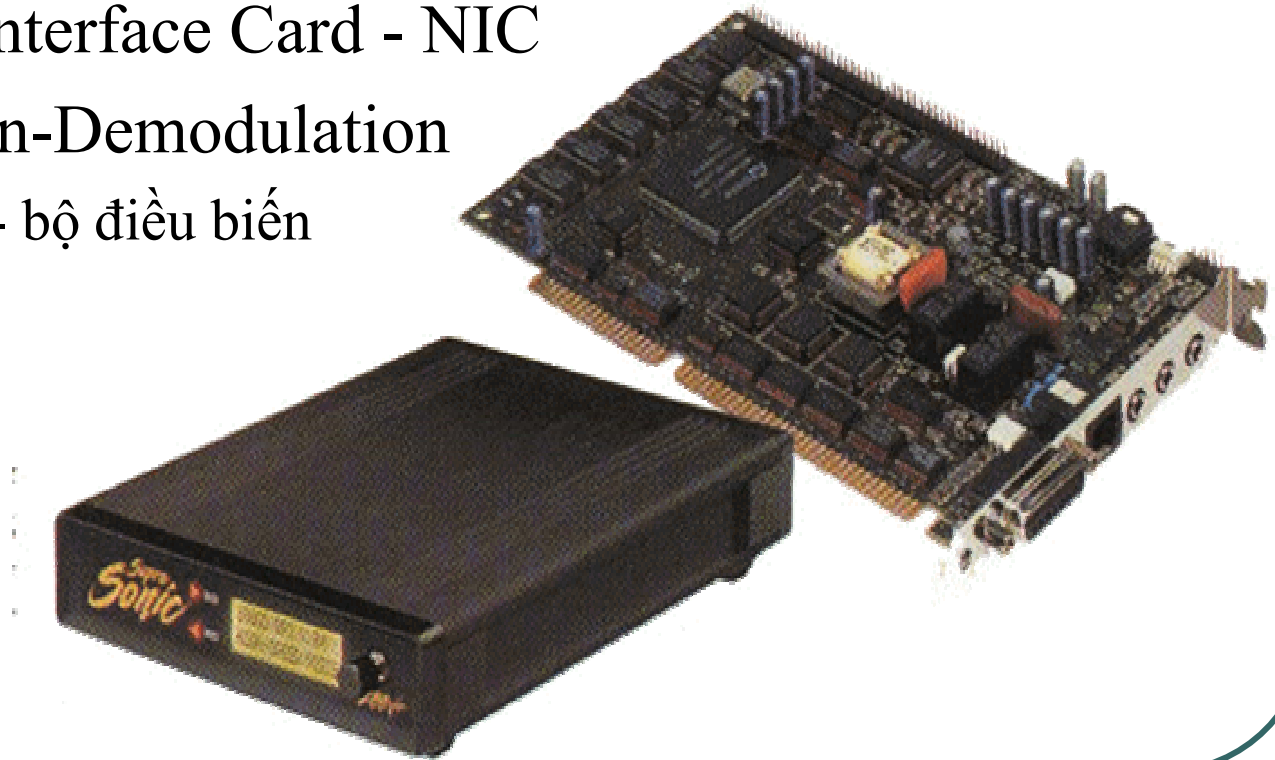
- Xuất ra âm thanh
  - Bo mạch xử lý âm thanh - sound card
  - Midi
    - Bảng mã qui định sẵn - các nhạc cụ điện tử
  - Wave
    - Mọi thứ âm nghe và tổng hợp



# Thiết bị xuất (10)

---

- Kết nối máy tính khác
  - Remote Access
  - Network Interface Card - NIC
  - Modulation-Demodulation
    - Modem - bộ điều biến



# Thiết bị lưu trữ (1)

---

- Đĩa từ tính
- Đĩa từ quang
- Đĩa quang

## Thiết bị lưu trữ (2)

---

- Đĩa từ tính
  - Đĩa mềm - floppy disk
    - Vỏ bảo vệ & đĩa plastic nhỏ có phủ vật liệu từ
      - Thông tin dạng số: hạt từ tính
    - 3.5 inch - đĩa 1.44Mb
    - Hạn chế về dung lượng nhớ, tốc độ chậm
    - Sao lưu dữ liệu, tính cơ động cao, giá rẻ

# Thiết bị lưu trữ (3)

---

- **Đĩa cứng - Hard disk**
  - Vỏ cứng bảo vệ, các bộ phận điều khiển xuất nhập
  - Đĩa từ tính: 5.25 inch và 3.5 inch
  - Dung lượng ổ cứng: 100MB → 160GB
  - Mạch giao tiếp: chuẩn ESDI, IDE, SCSI
    - ESDI (Enhanced Small Device Interface), 1983 24 MB/sec
    - IDE (Intelligent Drive Electronic - Intergrated Drive Electronic) - còn gọi ATA (AT Attachment)
    - SCSI (Small Computer System Interface)
      - 4Mb/giây → 10Mb/giây.

# Thiết bị lưu trữ (4)

---

- Đĩa từ quang
  - Magneto optical drive – MO
  - Ánh sáng laser
  - Dung lượng
    - 5.25 inch là 1.3Gb,
    - 3.5inch là 230 Mb.
  - Bảo đảm dữ liệu 50 năm

# Thiết bị lưu trữ (5)

---

- **Đĩa quang học**
  - Nguyên tắc quang học, tia laser
  - Ba điểm khác biệt chính so với từ tính
    - Dung lượng cao hơn
    - Độ bền cao hơn
    - Tháo lắp dễ dàng

# Thiết bị lưu trữ (6)

---

- **CD-ROM**
  - Compact Disc Read Only Memory
  - Đường kính 12cm, dày 1,2mm
  - Dung lượng ~ 680Mb
- **DVD**
  - Digital Video Disc, Digital Versatile Disc
  - nhiều lớp trên một mặt
  - Dung lượng 3.8 Gb → 17 Gb



## Đường truyền – Cổng thiết bị ngoại vi

---

- Thông tin qua lại giữa các linh kiện thông qua một mạng lưới các *Bus*:
  - 8, 16 hay 32 đường - bus 8bit, bus 16bit bus 32bit.
  - Bus nối với các khe - slot trên bo mạch.
  - Bus nối với các cổng nằm ngoài
- Các thiết bị ngoại vi kết nối qua:
  - Cổng có sẵn của máy
  - Card chuyên biệt

# Sự phát triển của máy tính điện tử (1)

---

- **1946-1948:**

- Sự ra đời của các máy tính điện tử đầu tiên.
- Tiếp đó là việc sản xuất hàng loạt máy tính điện tử thế hệ thứ nhất và thứ hai trong thập kỷ 50
- Chủ yếu được sử dụng trong tính toán khoa học-kỹ thuật.

## Sự phát triển của máy tính điện tử (2)

---

- **Giữa thập kỷ 60:**

- Sự ra đời của các máy tính điện tử thế hệ thứ ba với kỹ thuật mạch tích hợp và các bộ nhớ bán dẫn.
- Máy tính điện tử được bắt đầu ứng dụng trong kinh doanh, quản lý kinh tế.
- Máy lớn và đắt, nên thường chỉ được trang bị cho các trung tâm tính toán. Cuối những năm 60 sang đầu những năm 70 bắt đầu có các mạng nối các trung tâm tính toán với nhau.

## Sự phát triển của máy tính điện tử (3)

---

- **Giữa thập kỷ 70:**

- Ra đời các bộ vi xử lý: các linh kiện thực hiện chức năng của cả bộ xử lý trung tâm được chứa trong chỉ một "chip" bán dẫn có diện tích khoảng 1-2cm<sup>2</sup>.
- Kỹ thuật vi xử lý khởi đầu một cuộc cách mạng trong tin học, tạo cơ sở cho sự ra đời máy vi tính với năng lực ngày càng cao, giá ngày càng rẻ, thâm nhập khắp mọi nơi trên thế giới và được sử dụng trong mọi lĩnh vực.

## Sự phát triển của máy tính điện tử (4)

---

- **Cuối thập kỷ 80 đến nay:**
  - Sự phát triển bùng nổ của các mạng viễn thông truyền dữ liệu quốc gia và quốc tế
  - Xuất hiện những hệ thống “siêu xa lộ thông tin” liên kết hàng trăm triệu người trong từng quốc gia cũng như trong phạm vi khu vực và toàn cầu.
  - Xây dựng “kết cấu hạ tầng về thông tin” làm nền móng cho một “xã hội thông tin”.

# Tốc độ xử lý của máy tính tăng rất nhanh

## **Chiếc máy tính đầu tiên:**

- ENIAC (1947)
- Tốc độ: 6 - 7 nghìn phép tính/giây
- Tiêu hao năng lượng rất nhiều và tỏa nhiệt rất lớn
- Diện tích là 1.800m<sup>2</sup>
- Giá 450 nghìn USD lúc đó (~ 4 - 5 triệu USD bây giờ).

## **Hiện nay:**

- Máy tính thông thường
- Tốc độ: hàng chục tỉ phép tính/giây
- Tiêu hao năng lượng và tỏa nhiệt ít
- Diện tích: ?
- Giá: ?

# Định luật Moore

- Cứ sau mỗi 18 tháng, khả năng xử lý của máy tính tăng gấp đôi.
- Tương ứng với mật độ của bóng bán dẫn (transistor) trên một bảng mạch (chip).

| Bộ xử lý    | Năm sản xuất | Số bóng bán dẫn |
|-------------|--------------|-----------------|
| 4004        | 1971         | 2.250           |
| 8008        | 1972         | 2.500           |
| 8080        | 1974         | 5.000           |
| 8086        | 1978         | 29.000          |
| 286         | 1982         | 120.000         |
| 386         | 1985         | 275.000         |
| 486DX       | 1989         | 1.180.000       |
| Pentium     | 1993         | 3.100.000       |
| Pentium II  | 1997         | 7.500.000       |
| Pentium III | 1997         | 24.000.000      |
| Pentium IV  | 2000         | 42.000.000      |

# **CHƯƠNG 3: BIỂU DIỄN THÔNG TIN TRONG MÁY TÍNH ĐIỆN TỬ**



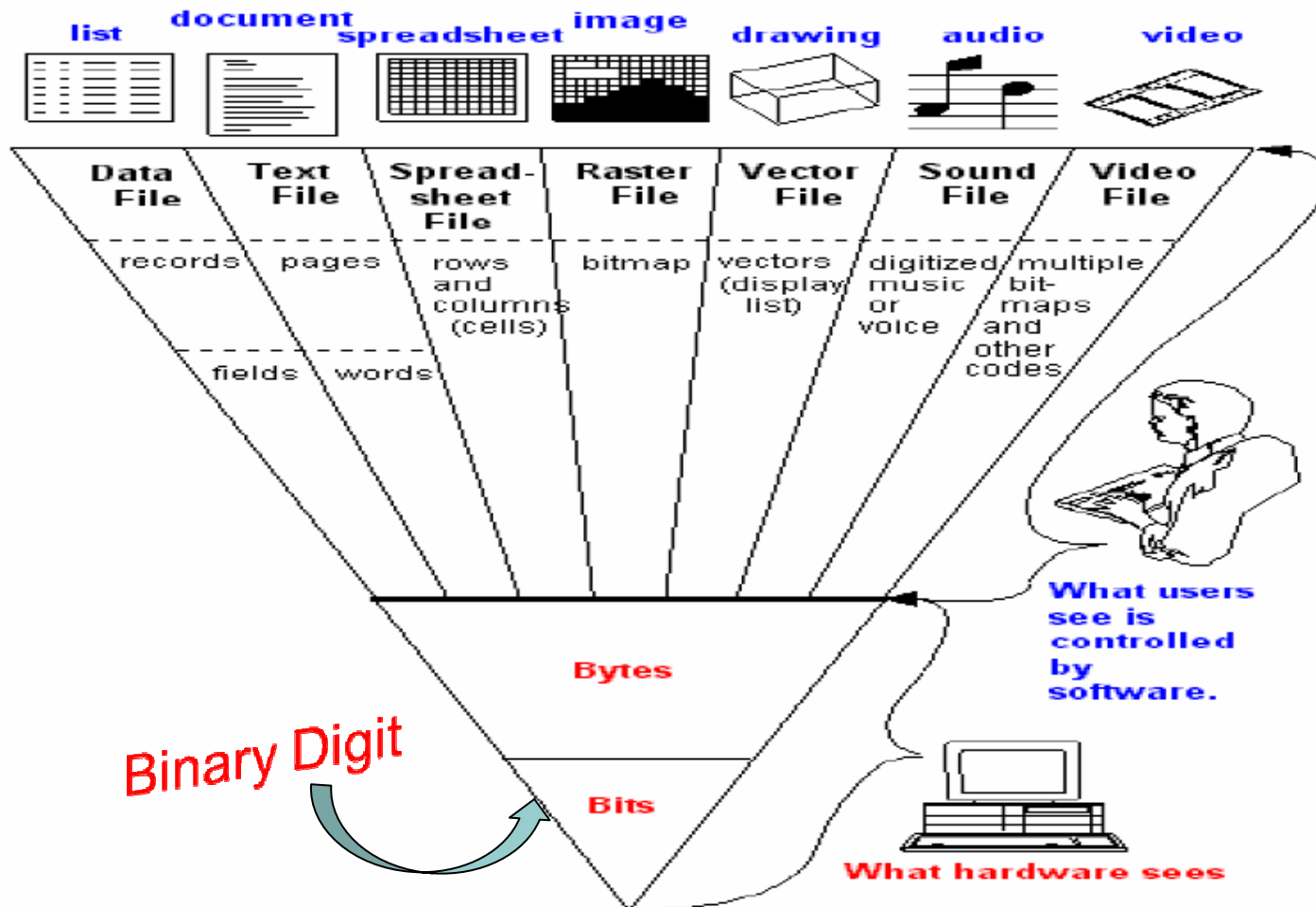
## Chương 3: Biểu diễn thông tin trong máy tính điện tử

---

- Máy tính lưu trữ thông tin như thế nào?
- Các hệ đếm
- Chuyển đổi giữa các hệ đếm

# Máy tính lưu trữ thông tin như thế nào?

From Computer Desktop Encyclopedia  
© 2004 The Computer Language Co., Inc.



# Các hệ đếm

---

- Hệ thập phân
- Hệ nhị phân
- Hệ bát phân
- Hệ thập lục phân

# Hệ thập phân (Số tự nhiên)

---

- Sử dụng hằng ngày
- 10 chữ số: 0→9
- Quy tắc: học từ lớp 1

# Hệ cơ số (1)

---

- Là hệ thống chữ số dùng để biểu diễn số
- Hệ cơ số  $k$ :
  - Dùng  $k$  chữ số để biểu diễn
  - Ký hiệu có giá trị nhỏ nhất là '0'
  - Ký hiệu có giá trị lớn nhất là  $k-1$

## Hệ cơ số (2)

---

- 2 chữ số: 0 và 1 → Hệ cơ số 2: nhị phân
- 8 chữ số: 0,1,2,3,4,5,6,7 → Hệ cơ số 8: bát phân
- 10 chữ số: 0,1,2,3,4,5,6,7,8,9 → Hệ cơ số 10: thập phân
- 16 chữ số: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F → Hệ cơ số 16 hay hệ thập lục phân

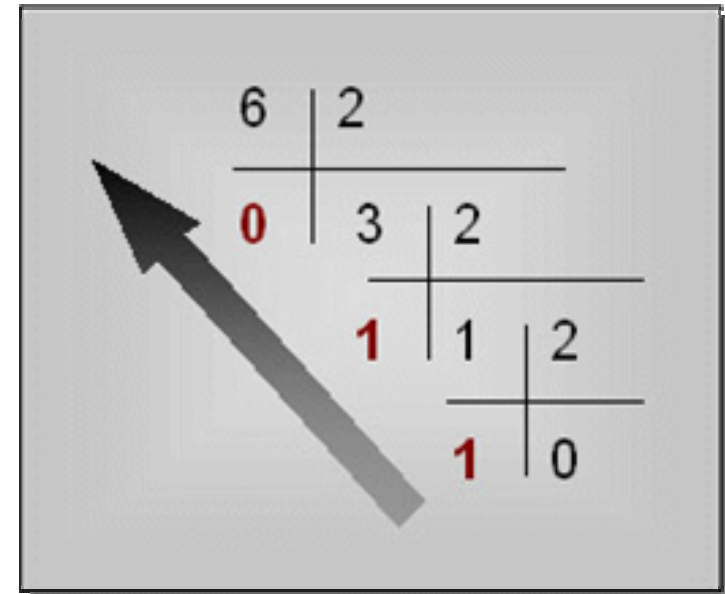
# Chuyển đổi giữa các hệ đếm

---

- Quy tắc 1
- Quy tắc 2
- Quy tắc 3
- Quy tắc 4

# Quy tắc 1 - Chuyển đổi từ hệ thập phân sang hệ cơ số b

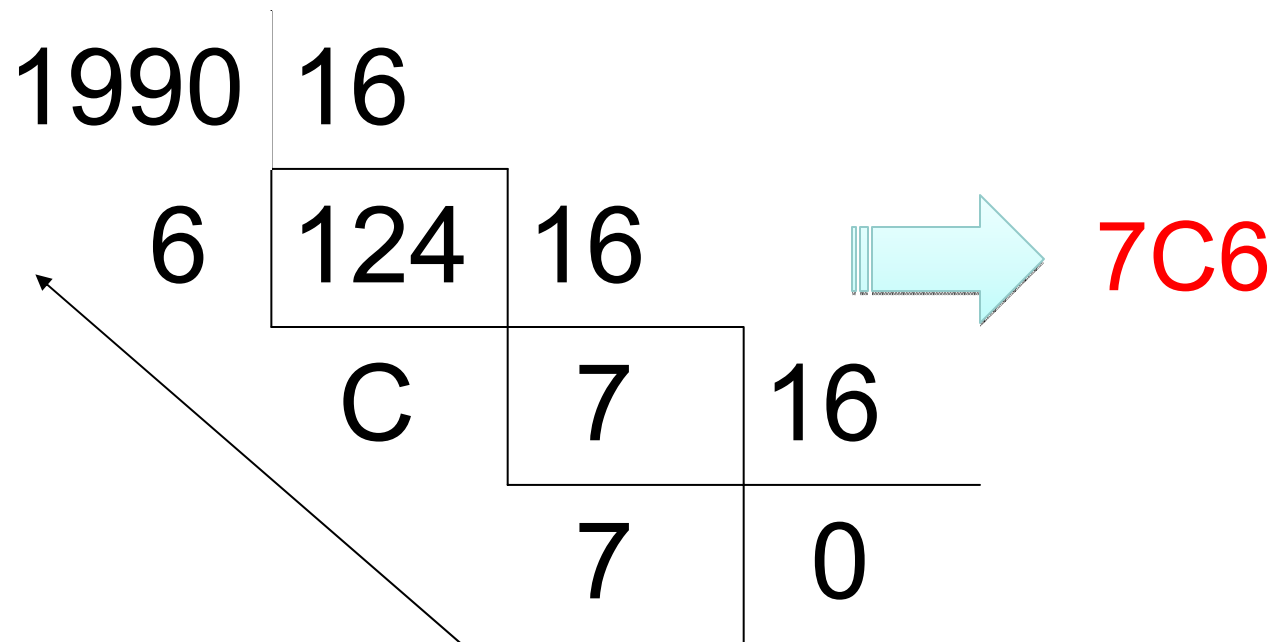
- Lấy số thập phân chia cho cơ số b ( $b \neq 10$ ) cho đến khi phần thương của phép chia bằng 0
- Số đổi được chính là các phần dư của phép chia theo thứ tự ngược lại.



110



Ví dụ:  $1990_{10} = ?_{16}$



## Qui tắc 2 – Chuyển đổi từ hệ cơ số b về hệ thập phân

---

$$\begin{aligned} X &= a_n a_{n-1} \dots a_1 a_0 \\ &= a_n b^n + a_{n-1} b^{n-1} + \dots + a_1 b^1 + a_0 b^0 \quad (*) \end{aligned}$$

Ví dụ:

- $X = 110_2$  thì  $X = 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 6$ .
- $X = 7C6_{16}$  thì  $X = 7 \cdot 16^2 + 12 \cdot 16^1 + 6 \cdot 16^0 = 1990$ .

## Quy tắc 3 - Chuyển từ hệ nhị phân về hệ thập lục phân

---

- Nhóm lần lượt 4 bit từ phải sang trái
- Sau đó thay thế các nhóm 4 bit bằng giá trị tương ứng với hệ thập lục phân
- Ví dụ:
  - $X = 11'1011_2 = 3B_{16}$

| Thập phân | Nhị phân | Thập lục phân |
|-----------|----------|---------------|
| 0         | 0        | 0             |
| 1         | 1        | 1             |
| 2         | 10       | 2             |
| 3         | 11       | 3             |
| 4         | 100      | 4             |
| 5         | 101      | 5             |
| 6         | 110      | 6             |
| 7         | 111      | 7             |
| 8         | 1000     | 8             |
| 9         | 1001     | 9             |
| 10        | 1010     | A             |
| 11        | 1011     | B             |
| 12        | 1100     | C             |
| 13        | 1101     | D             |
| 14        | 1110     | E             |
| 15        | 1111     | F             |

## Quy tắc 4 - Chuyển từ hệ thập lục phân về hệ nhị phân

---

- Ứng với mỗi chữ số sẽ được biểu diễn dưới dạng 4 bit
- Ví dụ:
  - $X = 3B_{16} = 0011'1011_2 = 11'1011_2$

# Bài tập 1

---

1.  $453_{10} = ?_{16}$

2.  $A35D_{16} = ?_8$

3.  $2456_8 = ?_2$

4.  $23E_{16} = ?_{10}$

5.  $291_{10} = ?_{16}$

6.  $EF17_{16} = ?_8$

7.  $2713_8 = ?_2$

8.  $A31_{16} = ?_{10}$

9.  $507_{10} = ?_{16}$

10.  $6F0A_{16} = ?_8$

11.  $3247_8 = ?_2$

12.  $D2A_{16} = ?_{10}$

# Phép cộng

---

- Qui tắc:
  - Thực hiện theo thứ tự từ phải sang trái
  - Cộng hai chữ số cùng cột
  - Nếu tổng của hai chữ số lớn hơn cơ số thì phần lớn hơn được nhớ chuyển sang cột tiếp theo.

- Ví dụ:

$$\begin{array}{r} 100101_2 \\ + 001011_2 \\ \hline 110000_2 \end{array}$$

$$\begin{array}{r} 1465_8 \\ + 407_8 \\ \hline 2074_8 \end{array}$$

# Phép trừ

---

- Qui tắc:
  - Thực hiện theo thứ tự từ phải sang trái
  - Trừ hai chữ số cùng cột
  - Nếu chữ số của số bị trừ nhỏ hơn số trừ thì mượn 1 đơn vị từ cột tiếp theo.
- Ví dụ:

$$\begin{array}{r} 100101_2 \\ - 001011_2 \\ \hline 011010_2 \end{array}$$

$$\begin{array}{r} 1465_8 \\ - 407_8 \\ \hline 1056_8 \end{array}$$



# Phép nhân

- Qui tắc:

- Thực hiện theo thứ tự từ phải sang trái
- Nhân hai chữ số cùng cột
- Nếu tích của hai chữ số lớn hơn cơ số, thì phần lớn hơn được cộng dồn vào kết quả tiếp theo

$$\begin{array}{r} 1011 \\ \times 101 \\ \hline 1011 \\ 0000 \\ 1011 \\ \hline 110111 \end{array}$$

# Phép chia

- Quy tắc: Tương tự phép chia số thập phân
- Ví dụ:

$$\begin{array}{r} \phantom{111}101 \phantom{00} \phantom{00} \\ - \phantom{111}101 \phantom{00} \phantom{00} \\ \hline \phantom{111}0100 \phantom{00} \\ - \phantom{111}000 \phantom{00} \\ \hline \phantom{111}1001 \phantom{00} \\ - \phantom{111}101 \phantom{00} \\ \hline \phantom{111}100 \phantom{00} \end{array}$$

$$\rightarrow 11101/101 = 101 \text{ dư } 100$$

# Phép toán AND

| A | B | A AND B |
|---|---|---------|
| 1 | 1 | 1       |
| 1 | 0 | 0       |
| 0 | 1 | 0       |
| 0 | 0 | 0       |

- Ví dụ:

```
 1 1 0 1 1 0 1 0
AND 1 0 0 1 1 1 1 0

 1 0 0 1 1 0 1 0
```

# Phép toán OR

| A | B | A OR B |
|---|---|--------|
| 1 | 1 | 1      |
| 1 | 0 | 1      |
| 0 | 1 | 1      |
| 0 | 0 | 0      |

- Ví dụ:

$$\begin{array}{r} 10011010 \\ \text{OR} \quad 11001001 \\ \hline 11011011 \end{array}$$

# Phép toán XOR

| A | B | A OR B |
|---|---|--------|
| 1 | 1 | 0      |
| 1 | 0 | 1      |
| 0 | 1 | 1      |
| 0 | 0 | 0      |

- Sử dụng chính của phép toán này là lấy phần bù của một chuỗi bit.

- Ví dụ:

$$\begin{array}{r} \text{XOR} \quad 10011010 \\ \quad 11001001 \\ \hline 01010011 \end{array}$$

$$\begin{array}{r} \text{XOR} \quad 11111111 \\ \quad 10101001 \\ \hline 01010110 \end{array}$$

# Phép dịch chuyển SHIFT (1)

---

- Qui tắc:
  - Được thực hiện tùy theo hướng di chuyển của các dãy bit
  - Cho dãy bit của nó sang hướng trái/phải thì bit đầu tiên của dãy (là bit cao nhất nếu dịch chuyển sang trái, hay bit thấp nhất khi dịch chuyển sang phải) sẽ bị chuyển đi
  - Bit cuối cùng của nó (là bit cao nhất nếu dịch chuyển sang trái, hay bit thấp nhất khi dịch chuyển sang phải) sẽ được đặt là 0

## Phép dịch chuyển SHIFT (2)

---

- Ví dụ:

- SHIFT LEFT    10001110 → 00011100
- SHIFT RIGHT   10001110 → 01000111

# Phép quay ROTATION

---

- Qui tắc:
  - Tương tự phép dịch chuyển SHIFT
  - Bit cuối sẽ được chuyển vào bit đầu tiên
- Ví dụ:
  - ROTATION LEFT    10001110 → 00011101
  - ROTATION RIGHT   10001110 → 01000111



# Biểu diễn số nguyên

---

- Số nguyên không dấu (không âm)
  - Số biểu diễn ở hệ nhị phân
  - Biểu diễn: 8 bit, 16 bit, 32bit hay 64bit
- Số nguyên có dấu:
  - Dựa trên hệ nhị phân không dấu
  - Có 3 phương pháp:
    - Phương pháp dấu lượng
    - Phương pháp biểu diễn số bù 1
    - Phương pháp biểu diễn số bù 2

# Phương pháp dấu lượng (1)

---

- Sign – Magnitude
- Bit cực trái được dùng làm bit dấu
  - 0 là dấu “+”
  - 1 là dấu “-”
- Các bit còn lại biểu diễn độ lớn của số
- Phương pháp để biểu diễn số âm K bit:
  - Lấy phần dương của số đó cộng thêm  $2^{k-1}$
  - Biểu diễn chúng ở hệ nhị phân.

## Phương pháp dấu lượng (2)

- Ví dụ:
  - Mẫu 4 bit
  - Số +5
    - $+5 = 5$
    - 0101
  - Số -5
    - $5 + 8 = 13$
    - 1101

| Mẫu bit | Giá trị | Mẫu bit | Giá trị |
|---------|---------|---------|---------|
| 1111    | -7      | 0111    | 7       |
| 1110    | -6      | 0110    | 6       |
| 1101    | -5      | 0101    | 5       |
| 1100    | -4      | 0100    | 4       |
| 1011    | -3      | 0011    | 3       |
| 1010    | -2      | 0010    | 2       |
| 1001    | -1      | 0001    | 1       |
| 1000    | -0      | 0000    | 0       |

## Phương pháp dấu lượng (3)

---

- Nhược điểm:
  - Số 0 biểu diễn 2 cách
  - Yêu cầu phép tính toán riêng
  - $-5 + 5 = 0$
  - $1101 + 0101 = 10010$  ???
- Cải tiến phương pháp dấu lượng
  - ➔ Ký pháp lệch (biased representation)

# Ký pháp lệch (1)

---

- Qui tắc:
  - Giá trị  $x$  biểu diễn bằng  $x + N$
  - $N$  là giá trị lệch
    - $N = 2^{n-1}$  hoặc  $N = 2^{n-1} - 1$
    - $n$  là số bit biểu diễn
- Ví dụ:
  - Với mẫu 4-bit,  $N = 2^3$
  - Số +5:  $5 + 2^3 = 5 + 8 = 13 \rightarrow 1101$
  - Số -5:  $-5 + 2^3 = -5 + 8 = 3 \rightarrow 0011$
  - $5 + -5 = 0 = 1101 + 0011 = 1\ 0000$

## Ký pháp lịch (2)

---

- Ưu điểm:
  - Chỉ một cách biểu diễn số 0
  - Với mẫu 4 bit:
    - **0** biểu diễn là **1000**
    - **0000** là biểu diễn của **-8**

# Phương pháp biểu diễn số bù 1 (1)

---

- One's complement
- Bit cực trái làm bit dấu
  - 0 cho số dương
  - 1 cho số âm
- Quy tắc biểu diễn số bù 1
  - Biểu diễn dưới dạng nhị phân của trị tuyệt đối  $n$  theo mẫu  $k$ -bit cố định cho trước.
  - Nếu  $n < 0$  thì đổi 1 thành 0 và ngược lại trong dãy số nhị phân (*gọi là đảo ngược bit*)

# Phương pháp biểu diễn số bù 1 (2)

---

- Ví dụ:
  - $n=5 \rightarrow$  Dùng mẫu 4 bit thì biểu diễn là: 0101
  - $n = -5 \rightarrow$  Dùng mẫu 4 bit thì biểu diễn là: 1010
  - $n = 6 \rightarrow$  Dùng mẫu 4 bit thì biểu diễn là: 0110
  - $n = -6 \rightarrow$  Dùng mẫu 4 bit thì biểu diễn là: 1001



# Phương pháp biểu diễn số bù 2 (1)

---

- Two's complement
- Bit cực trái làm bit dấu tương tự bù 1
- Quy tắc biểu diễn số bù 2:
  - Biểu diễn dưới dạng nhị phân của trị tuyệt đối  $n$  theo mẫu  $k$  bit cố định cho trước.
  - Nếu  $n < 0$ :
    - bắt đầu từ **phải qua trái** giữ nguyên các bit cho đến khi gặp bit có giá trị là 1 đầu tiên,
    - sau đó **các bit tiếp theo bên trái bit 1 đầu tiên** đó đổi 1 thành 0 và ngược lại (*đảo bit*)

## Phương pháp biểu diễn số bù 2 (2)

- Thực chất số biểu diễn bù 2 là số biểu diễn bù 1 cộng 1
- Ví dụ mẫu 4 bit  
→

| Mẫu bit | Giá trị | Mẫu bit | Giá trị |
|---------|---------|---------|---------|
| 0111    | 7       | 1111    | -1      |
| 0110    | 6       | 1110    | -2      |
| 0101    | 5       | 1101    | -3      |
| 0100    | 4       | 1100    | -4      |
| 0011    | 3       | 1011    | -5      |
| 0010    | 2       | 1010    | -6      |
| 0001    | 1       | 1001    | -7      |
| 0000    | 0       | 1000    | -8      |

## Phương pháp biểu diễn số bù 2 (3)

| <b>digits</b> | <b>binary</b> | <b>actual value</b> |
|---------------|---------------|---------------------|
| 0             | 0000 0000     | 0                   |
| 1             | 0000 0001     | 1                   |
| ....          |               |                     |
| 126           | 0111 1110     | 126                 |
| 127           | 0111 1111     | 127                 |
| 128           | 1000 0000     | -128                |
| 129           | 1000 0001     | -127                |
| 130           | 1000 0010     | -126                |
| ....          |               |                     |
| 254           | 1111 1110     | -2                  |
| 255           | 1111 1111     | -1                  |

# Phép cộng khi biểu diễn bù 1 (1)

---

- Qui tắc:
  - Thực hiện như phép toán tương ứng trên hệ nhị phân
  - Nếu ở 2 bit cực trái khi thực hiện phép cộng mà phát sinh bit nhớ → cộng nhớ vào kết quả.

# Phép cộng khi biểu diễn bù 1 (2)

---

- Ví dụ biểu diễn bù 1 mẫu 4-bit
  - $-6 \rightarrow 1001$
  - $4 \rightarrow 0100$
  - $1001 + 0100 = 1101 \rightarrow -2$
- Ví dụ biểu diễn bù 1 mẫu 5-bit
  - $-6 \rightarrow 11001$
  - $-4 \rightarrow 11011$
  - $11001 + 11011 = 1\ 10100$  (1 là nhớ)
  - $10100 + 1 = 10101 \rightarrow -10$

# Phép cộng khi biểu diễn bù 2 (1)

---

- Qui tắc:
  - Thực hiện như phép toán tương ứng trên hệ nhị phân
  - Nếu ở 2 bit cực trái khi thực hiện phép cộng mà phát sinh bit nhớ → loại bỏ nhớ

## Phép cộng khi biểu diễn bù 2 (2)

---

- Ví dụ biểu diễn bù 2 mẫu 4-bit
  - $-6 \rightarrow 1010$
  - $4 \rightarrow 0100$
  - $1010 + 0100 = 1110 \rightarrow -2$  ở biểu diễn bù 2
- Ví dụ biểu diễn bù 2 mẫu 5-bit
  - $-6 \rightarrow 11010$
  - $-4 \rightarrow 11100$
  - $11010 + 11100 = 1\ 10110$  (1 là nhớ)  $\rightarrow$  loại bỏ 1 nhớ:  $10110 \rightarrow -10$  ở biểu diễn bù 2

# Lỗi tràn số (1)

---

- Hiện tượng xảy ra khi số cần biểu diễn vượt quá số bit cho trước để biểu diễn nó
- Ví dụ:
  - Mẫu 4 bit cho biểu diễn bù 2 cho -6, -4
  - $-6 \rightarrow 1010$ ,  $-4 \rightarrow 1100$
  - Kết quả phép cộng ở dạng bù 2 là 0110 là biểu diễn của +6, do đó kết quả bị sai.
- Nguyên nhân: số lượng bit để biểu diễn quá ít



## Lỗi tràn số (2)

---

- Khắc phục: tăng số lượng bit nhiều hơn
  - Mẫu 32 bit số dương lớn nhất 2147483647
- Tổng quát:
  - Ở phép biểu diễn bù 1:
    - giá trị dương lớn nhất:  $2^{n-1}$
    - giá trị âm nhỏ nhất:  $-2^{n-1}$
  - Ở phép biểu diễn bù 2 có n-bit
    - giá trị dương lớn nhất:  $2^{n-1} - 1$
    - giá trị âm nhỏ nhất:  $-2^{n-1}$

# Bài tập A – Dùng 8 bit

---

Biểu diễn số bù 1:

1. -120
2. -121
3. -122
4. -123
5. -124
6. -110

Biểu diễn số bù 2:

1. -101
2. -102
3. -103
4. -104
5. -105
6. -106

## Bài tập B – Dùng 16 bit

---

Biểu diễn số bù 1 &

Thực hiện phép cộng:

1.  $(-74) + (-25)$

2.  $(-75) + (-26)$

3.  $(-76) + (-27)$

4.  $(-77) + (-28)$

5.  $(-78) + (-29)$

6.  $(-79) + (-30)$

Biểu diễn số bù 2 &

Thực hiện phép cộng:

1.  $(-91) + 13$

2.  $(-92) + 14$

3.  $(-93) + 15$

4.  $(-94) + 16$

5.  $(-95) + 17$

6.  $(-19) + 18$

## Bài tập C

---

- Tính giá trị dương lớn nhất, giá trị âm nhỏ nhất nếu sử dụng:
  - 8 bit để biểu diễn số bù 1
  - 16 bit để biểu diễn số bù 2

## Bài tập D - Các phép toán luận lý

---

1. 11101010011 AND 111000111
2. 10100111010 OR 100110011
3. 10100111010 XOR 100110011
4. SHIFT LEFT 3 1110101110
5. SHIFT RIGHT 1110101110
6. ROTATION LEFT 2 1110101110

# Biểu diễn số thực

---

- Biểu diễn số thực với các cơ số
- Phương pháp dấu chấm động (Floating Point)

# Biểu diễn số thực với các cơ số (1)

---

- Dùng dấu chấm cơ số (*dấu . hoặc dấu ,*)
  - Radix point
- Biểu diễn số thực trong hệ  $k$ 
  - Phần nguyên: trước dấu chấm cơ số
  - Phần lẻ: sau dấu chấm cơ số
    - Phần thập phân trong hệ 10
- Qui tắc???

## Biểu diễn số thực với các cơ số (2)

---

- **Qui tắc: đổi sang hệ 10**

$$a_n a_{n-1} \dots a_0, b_1 b_2 \dots b_m (k) =$$

$$a_n \times k^n + a_{n-1} \times k^{n-1} + \dots + a_0 \times k^0 +$$

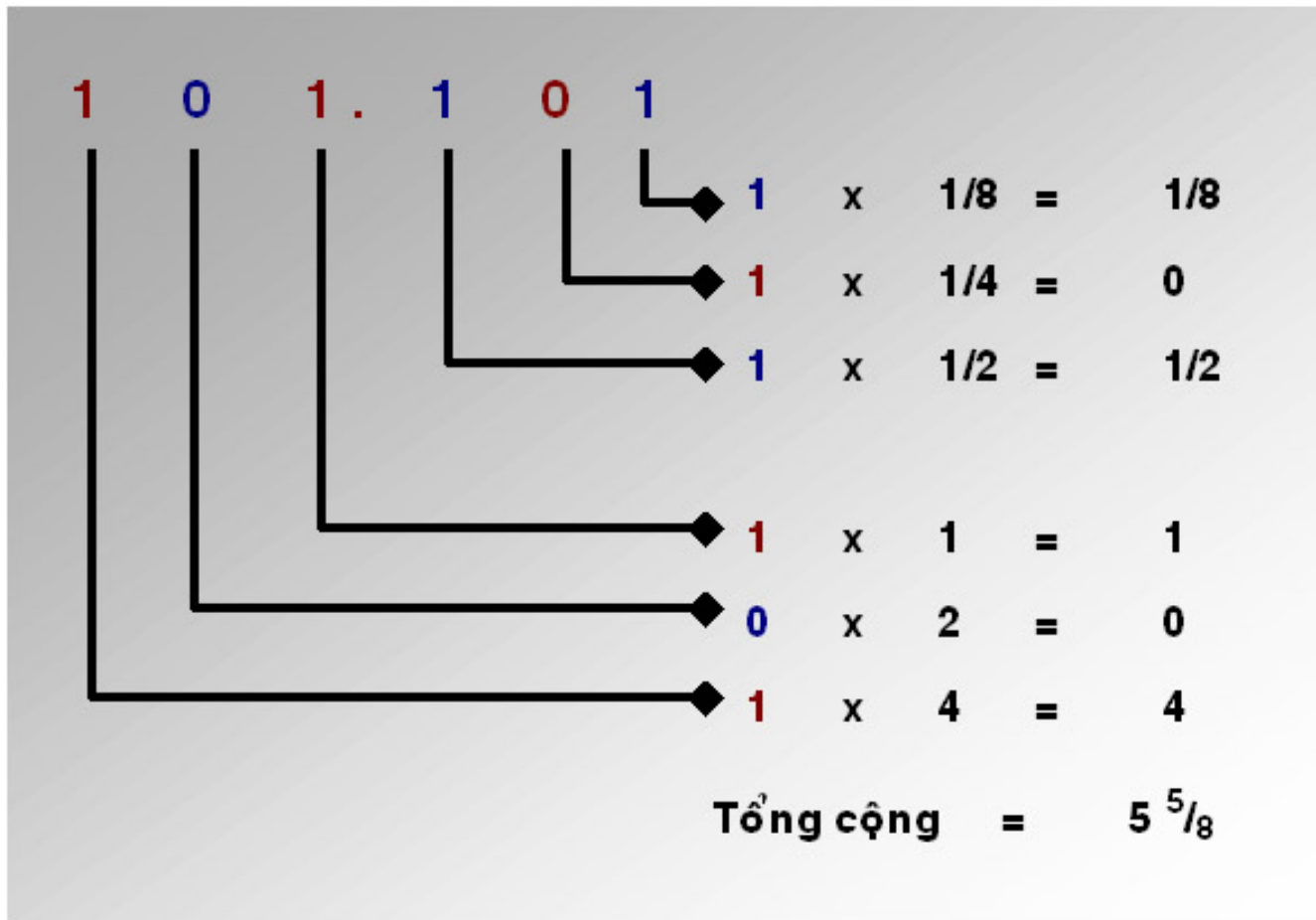
$$b_1 \times k^{-1} + b_2 \times k^{-2} + \dots + b_m \times k^{-m}$$

- **Tính  $101.101_2 = ?$**

**5.625**



# Biểu diễn số thực với các cơ số (3)



## Biểu diễn số thực với các cơ số (4)

---

- **Tính  $1F.28_h = ? = ?_2$**   
 $31.15625 = 11111.00101_2$
- **Tính  $35.21_8 = ?_h = ?_2 = ?$**

## Biểu diễn số thực với các cơ số (5)

---

- **Quy tắc đổi từ số thập phân có phần lẻ sang hệ số k**
  - Đổi phần nguyên như quy tắc đổi số nguyên
    - chia nguyên lần lượt cho k đến khi =0
    - kết quả phần nguyên là số dư ghi ngược lại
  - Đổi phần lẻ theo quy tắc:
    - nhân lần lượt chỉ phần lẻ cho k cho đến khi =0
    - kết quả phần lẻ là phần nguyên lần lượt khi nhân

## Biểu diễn số thực với các cơ số (6)

- Tính  $31.15625 = ?_h = ?_8 = ?_2$
- $31 = 1F_h = 37_8 = 11111_2$
- $0.15625 = 0.28_h = 0.12_8$   
 $= 0.00101_2$
- $31.15625 = 1F.28_h =$   
 $= 37.12_8 = 11111.00101_2$

|         |     |
|---------|-----|
| 0.15625 | x16 |
| 2       | .5  |
| 8       | .0  |

|         |     |
|---------|-----|
| 0.15625 | x8  |
| 1       | .25 |
| 2       | .0  |

## Biểu diễn số thực với các cơ số (7)

---

- **Chú ý** chuyển đổi qua lại giữa các hệ 2 và 8 (16) khi không đủ số lượng bit:
  - Phần nguyên thêm 0 phía trái nhất
  - Phần lẻ thêm 0 phía phải nhất

# Phương pháp dấu chấm động (1)

- Biểu diễn dấu chấm động

- Floating point
- Scientific format
- 3 phần: dấu + mũ + nguyên(trị)

$$+31.15625 = +0.3115625 \times 10^2$$

$+ 3115625 \times 10^{-5}$

Dấu (Sign)      Nguyên (Mantissa)      Mũ (exponent)

The diagram shows the scientific notation  $+ 3115625 \times 10^{-5}$  with three arrows pointing from labels below to its components: the '+' sign is labeled 'Dấu (Sign)', the digits '3115625' are labeled 'Nguyên (Mantissa)', and the '-5' is labeled 'Mũ (exponent)'.

# Phương pháp dấu chấm động (2)

- Khi biểu diễn số thực trên máy tính

- Chuyển về nhị phân
- Chuẩn hóa
- Biểu diễn floating point: **chỉ cần phần lẻ**

$$+31.15625 = +11111.00101_2$$

$$+11111.00101_2 = +1.111100101_2 \times 2^{+4}$$

$$\begin{array}{ccc} \nearrow & \uparrow & \nearrow \\ \text{Dấu} & \text{Nguyên} & \text{Mũ} \\ \text{(Sign)} & \text{(Mantissa)} & \text{(exponent)} \end{array} \quad +1.111100101_2 \times 2^{+100b}$$

## Phương pháp dấu chấm động (3)

- Biểu diễn số thực dấu chấm động 16 bit:
  - 1 bit trái nhất: bit dấu (+ hoặc -)
  - 4 bit tiếp theo: bit mũ (dấu lượng)
  - 11 bit cuối cùng: bit định trị (số nguyên)

$$+1.111100101_2 \times 2^{+100b}$$

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|



## Phương pháp dấu chấm động (4)

- Biểu diễn số thực dấu chấm động 16 bit:
- Biểu diễn  $-35.265625$

$$=-100011.010001_2$$

$$=-1.00011010001 \times 2^{+101b}$$

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

# Bài tập

---

- Biểu diễn số thực dấu chấm động 16 bit:
- **Biểu diễn -29.46875**

# Phương pháp dấu chấm động (5)

- Chuẩn IEEE

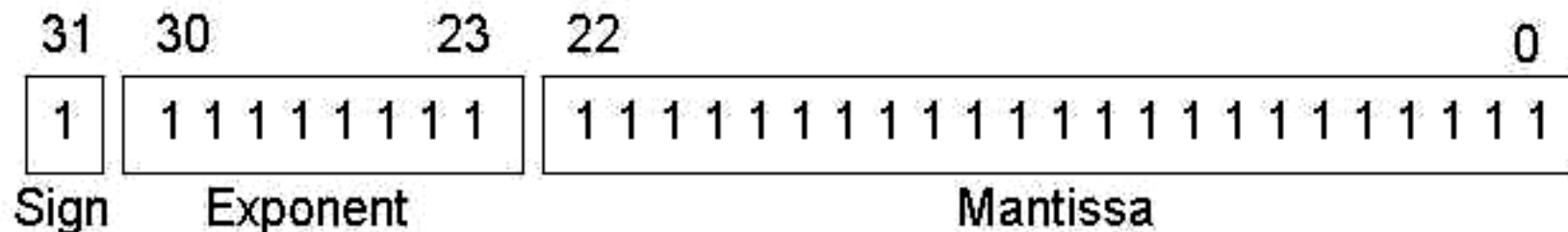
- Short Real-32 bits *single precision*

- 1 bit **sign**, 8 bits **exponent**, 23 bits **mantissa**.

- Long Real: 64 bits *double precision*

- 1 bit **sign**, 11 bits **exponent**, 52 bits **mantissa**.

- Exponent biểu diễn kí pháp lệch  $N=2^7-1=127$



# Phương pháp dấu chấm động (6)

| Giá trị         | Mũ Dấu lượng | Sign, Exponent, Mantissa            |
|-----------------|--------------|-------------------------------------|
| -1.11           | 127          | 1 01111111 110000000000000000000000 |
| +1101.101       | 130          | 0 1000010 101101000000000000000000  |
| -.00101         | 124          | 1 01111100 010000000000000000000000 |
| +100111.0       | 132          | 0 10000100 001110000000000000000000 |
| +.0000001101011 | 120          | 0 01111000 101011000000000000000000 |

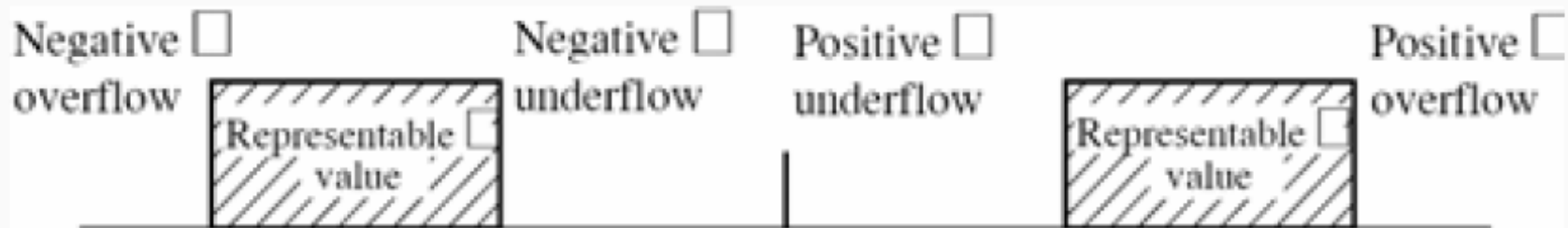
# Tóm tắt biểu diễn dấu chấm động

---

- Qui định định dạng
- Chuyển về nhị phân
- Chuẩn hóa
- Chuyển đổi giá trị phần mũ
- Tính kết quả từng phần
- Kết quả cuối cùng

# Độ chính xác trong tính toán số thực dấu chấm động

- Overflow: khi giá trị  $>$  giá trị tuyệt đối của khả năng biểu diễn lớn nhất.
- Underflow:  $0 <$  giá trị  $<$  giá trị tuyệt đối của khả năng biểu diễn nhỏ nhất.



# Biểu diễn ký tự

---

- Bộ mã ASCII ?
- Bảng mã UNICODE ?

# CHƯƠNG 4: CHƯƠNG TRÌNH & GIẢI THUẬT



# Các khái niệm (1)

---

- Thuật toán
  - Cách hiểu đơn giản
    - Tập các hướng dẫn thực hiện một công việc.
    - Tập hữu hạn các hướng dẫn rõ ràng để người giải toán có thể theo đó mà giải quyết được vấn đề.
    - Một phương pháp thể hiện lời giải của vấn đề.
  - Algorithm: nhà toán học Trung Á
    - **Muhammad Bin Musa Al-Khwarizmi**
    - <http://www2.sjsu.edu/depts/Museum/alkhwa.html>

## Các khái niệm (2)

---

- Thuật toán (tt)
  - Trong khoa học máy tính
    - Một dãy **hữu hạn** các bước **rõ ràng** và **thực thi được**.
    - **Quá trình** hành động theo các bước này phải **dừng** và cho được **kết quả như mong muốn**.
  - Tính xác định
    - Hướng dẫn giải **rõ ràng** và **đúng**
  - Tính hữu hạn
    - Số bước **hữu hạn** và tính chất **dừng**

## Các khái niệm (3)

---

- Tính mập mờ:
  - Xem xét thực hiện công việc tìm một môn học nhiều đvht nhất:
    - Lập danh sách các môn học.
    - Sắp thứ tự các môn học.
    - Chọn ra một môn học nhiều đvht.
  - Kết quả:???
  - Câu hỏi:???

## Các khái niệm (4)

---

- Tính mập mờ (tt):
  - Vi phạm **tính rõ ràng – không mập mờ**
    - **Hiểu đúng – Hiểu 1 nghĩa duy nhất**
  - Sửa lại :
    - 1.Lập danh sách các môn học theo tên, số đvht
    - 2.Sắp thứ tự các môn học giảm dần theo số đvht
    - 3.Chọn ra một môn học có nhiều đvht nhất
  - Câu hỏi???

## Các khái niệm (5)

---

- Tính mập mờ (tt):
  - Phân biệt *mập mờ* và *chọn lựa có quyết định*.
    - Mập mờ là ***thiếu thông tin*** hoặc có nhiều chọn lựa nhưng ***không đủ điều kiện để quyết định***.
    - Chọn lựa có quyết định là ***hoàn toàn xác định duy nhất*** trong điều kiện cụ thể của vấn đề
  - Sửa lại:
    - 3.Chọn ra một môn học có nhiều đvht nhất.
      - 3.1 Nếu chỉ có một môn học nhiều đvht nhất: chọn một
      - 3.2 Nếu có nhiều môn học cùng số đvht: sắp xếp tăng dần theo tên môn học trong thứ tự từ điển rồi chọn môn đầu tiên.

## Các khái niệm (6)

---

- Tính thực thi được:
  - Hiện nhiên
  - Chỉ xét trong điều kiện hiện tại của bài toán
  - Cho ví dụ về điều kiện hiện tại:???
- Tính dừng:
  - Không dừng: Lặp vô tận, bị quẩn
  - Dễ vi phạm nhất

# Các khái niệm (7)

---

- Tính dừng - Ví dụ:
  - Tính tổng các số nguyên dương lẻ trong khoảng từ 1 đến  $n$  ta có thuật toán sau :
    - B1. Hỏi giá trị của  $n$ .
    - B2.  $S = 0$
    - B3.  $i = 1$
    - B4. Nếu  $i = n+1$  thì sang bước B8, ngược lại sang bước B5
    - B5. Cộng thêm  $i$  vào  $S$
    - B6. Cộng thêm 2 vào  $i$
    - B7. Quay lại bước B4.
    - B8. Tổng cần tìm chính là  $S$ .
  - Thuật toán luôn dừng???

## Các khái niệm (8)

---

- Tính dừng - Ví dụ:
  - Chỉ dừng khi n chẵn
  - Khi n lẻ phải thay B4 bằng:
    - B4. Nếu  $i \geq n+1$  thì sang bước B8, ngược lại sang bước B5
- Tính đúng:
  - Chứng minh thuật toán đúng!!!
  - Khó đạt được nhất



## Các khái niệm (9)

---

- Thuật toán thì cứng nhắc !
  - Tính chất chặt chẽ và cứng nhắc.
  - Khả năng giải quyết vấn đề bị giới hạn.
- "làm mềm": *tính xác định và tính đúng*
  - Thuật toán đệ quy.
  - Thuật giải.

# Các khái niệm (10)

---

- Các đặc trưng khác của thuật toán
  - Đầu vào và đầu ra (input/output) .
  - Tính hiệu quả (effectiveness): theo tiêu chuẩn
    - Khối lượng tính toán, không gian và thời gian thi hành.
    - Là yếu tố quyết định để đánh giá, chọn lựa
    - Nhiều phương pháp để đánh giá tính hiệu quả của thuật toán.
  - Tính tổng quát (generalliness) :
    - Áp dụng được cho mọi trường hợp của bài toán
    - Không phải lúc nào cũng đảm bảo được tính tổng quát.
    - Có lúc chỉ cần xây dựng cho một dạng đặc trưng.

# Các khái niệm - Ví dụ (1)

---

- **Thuật toán giải phương trình bậc hai  $ax^2+bx+c=0$  ( $a \neq 0$ )**
  - 1. Yêu cầu cho biết giá trị của 3 hệ số  $a, b, c$
  - 2. Nếu  $a=0$  thì
    - 2.1. Yêu cầu đầu vào không đảm bảo.
    - 2.2. Kết thúc thuật toán.
  - 3. Trường hợp  $a$  khác 0 thì

## Các khái niệm - Ví dụ (2)

---

- 3. Trường hợp  $a \neq 0$  thì
  - 3.1. Tính giá trị  $D = b^2 - 4ac$
  - 3.2. Nếu  $D > 0$  thì
    - 3.2.1. Phương trình có hai nghiệm phân biệt  $x_1, x_2$
    - 3.2.2. Giá trị của hai nghiệm được tính theo công thức sau:
$$x_1 = \frac{-b + \sqrt{D}}{2a}, x_2 = \frac{-b - \sqrt{D}}{2a}$$
    - 3.2.3. Kết thúc thuật toán.
  - 3.3 Nếu  $D=0$  thì

## Các khái niệm - Ví dụ (3)

---

- 3.3. Nếu  $D = 0$  thì
  - 3.3.1. Phương trình có nghiệm kép  $x_0$
  - 3.3.2. Giá trị của nghiệm kép là
$$x = \frac{-b}{2a}$$
  - 3.3.3. Kết thúc thuật toán
- 3.4. Nếu  $D < 0$  thì
  - 3.4.1. Phương trình vô nghiệm.
  - 3.4.2. Kết thúc thuật toán.

## Các khái niệm - Ví dụ (4)

---

- Bài toán : Cho hai số nguyên dương  $a$  và  $b$ . Tìm ước số chung lớn nhất của  $a$  và  $b$ .
- **Thuật toán Euclid**
  - 1. Yêu cầu cho biết giá trị của  $a, b$ .
  - 2.  $a_0 = a$
  - 3.  $b_0 = b$
  - 4.  $i = 0$

## Các khái niệm - Ví dụ (5)

---

- 5. Nếu  $a_i$  khác  $b_i$  thì thực hiện các thao tác sau, ngược lại qua bước 7.
  - 5.1 Tăng  $i$  lên 1.
  - 5.2. Nếu  $a_{i-1} > b_{i-1}$  thì
    - $a_i = a_{i-1} - b_{i-1}$
    - $b_i = b_{i-1}$
  - 5.3. Ngược lại
    - $b_i = b_{i-1} - a_{i-1}$
    - $a_i = a_{i-1}$
- 6. Trở lại bước 5.
- 7. Ước số chung lớn nhất của  $a, b$  là  $a_i$ .

# Các phương pháp biểu diễn

---

- Thuật toán:
  - Một phương pháp thể hiện lời giải bài toán
  - Phải tuân theo một số quy tắc nhất định
- Có 3 phương pháp biểu diễn thuật toán :
  - Dùng **ngôn ngữ tự nhiên**.
  - Dùng **lưu đồ-sơ đồ khối (flowchart)**.
  - Dùng **mã giả (pseudocode)**.



# Ngôn ngữ tự nhiên

---

- Ngôn ngữ thường ngày:
  - Liệt kê các bước của thuật toán, quá trình thực hiện lần lượt (trừ khi có yêu cầu nhảy\_
  - Không thể hiện rõ cấu trúc của thuật toán
  - Dài dòng, có thể gây hiểu lầm hoặc khó hiểu
  - **Không** yêu cầu người viết hay đọc nắm quy tắc.
    - Không có một quy tắc cố định
  - Tính dễ đọc:
    - viết các bước con lùì vào bên phải
    - đánh số bước theo quy tắc phân cấp như 1, 1.1, ...

# Lưu đồ - sơ đồ khối (1)

---

- Công cụ trực quan diễn đạt thuật toán.
  - Biểu diễn bằng mô hình – hình vẽ
- Theo dõi được:
  - sự phân cấp các trường hợp
  - quá trình xử lý của thuật toán
- Được dùng trong những thuật toán
  - rắc rối
  - khó theo dõi được quá trình xử lý

## Lưu đồ - sơ đồ khối (2)

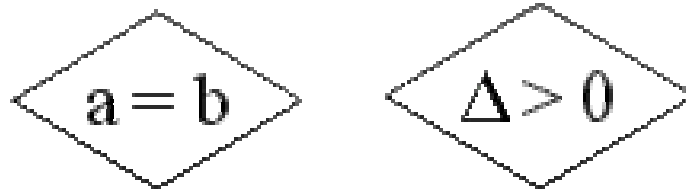
---

- Phân biệt hai loại thao tác:
  - **Chọn lựa** theo một điều kiện nào đó
  - **Xử lý, hành động**

# Lưu đồ - sơ đồ khối (3)

---

- **Chọn lựa** theo một điều kiện nào đó
  - **Decision**
  - Biểu diễn bằng một hình thoi, bên trong chứa biểu thức điều kiện.
  - Ví dụ: thao tác "*nếu  $a = b$  thì thực hiện thao tác B2, ngược lại thực hiện B4*" là thao tác chọn lựa



# Lưu đồ - sơ đồ khối (4)

---

- ***Xử lý, hành động.***
  - **Process**
  - Biểu diễn bằng một hình chữ nhật, bên trong chứa nội dung xử lý.
  - Ví dụ: "*Chọn một môn học và in ra.*" là một thao tác thuộc loại hành động.

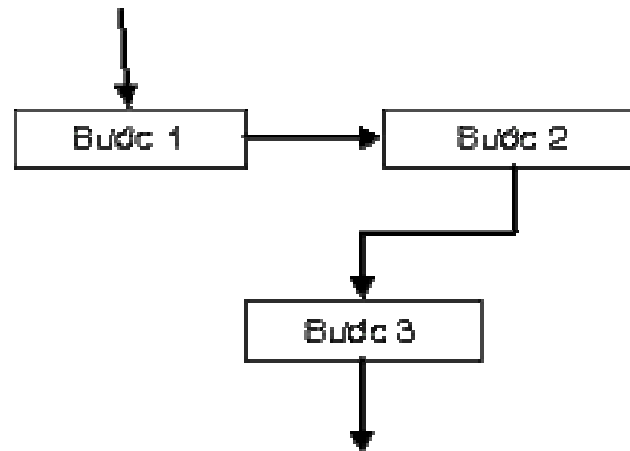
**Tăng lên 1**

**Chọn một hộp bất kỳ**

# Lưu đồ - sơ đồ khối (5)

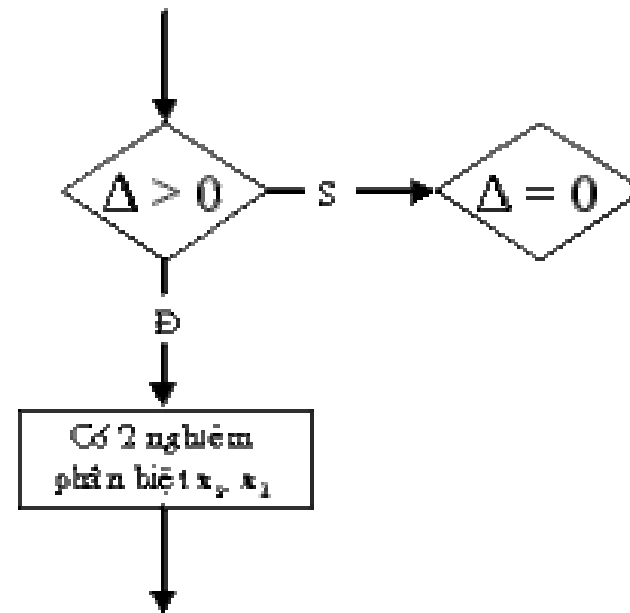
---

- Quá trình thực hiện các thao tác:
  - Đường đi – **route**
  - Biểu diễn bằng cung có hướng
    - nối giữa 2 thao tác: thực hiện lần lượt



# Lưu đồ - sơ đồ khối (6)

- Thao tác chọn lựa: có thể có hai hướng đi
  - một hướng ứng với điều kiện thỏa
  - một hướng ứng với điều kiện không thỏa.
  - 2 cung có nhãn
    - Đ/Đúng, Y/Yes
    - S/Sai, N/No



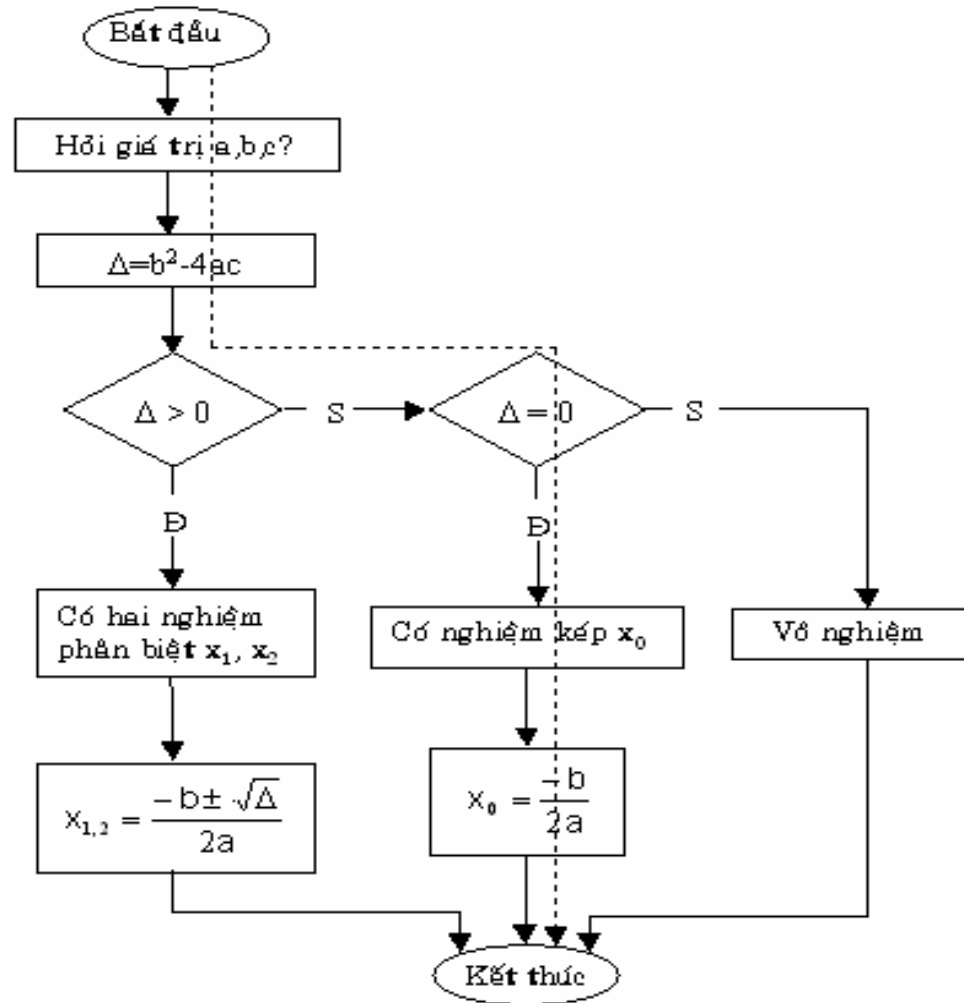
# Lưu đồ - sơ đồ khối (7)

---

- Điểm cuối (terminator)
  - Biểu diễn bằng hình ovan
  - Điểm khởi đầu
    - chỉ có cung đi ra
    - bên trong ovan ghi chữ: bắt đầu/start/begin
  - Điểm kết thúc
    - Chỉ có cung đi vào
    - bên trong ovan ghi chữ: kết thúc/end



Phương trình thuật toán giải pt bậc 2. Đường chấm ứng với trường hợp nghiệm kép, ví dụ:  $a=1, b=2, c=1$





# Mã giả (pseudo code) (1)

---

- **Vay mượn** các cú pháp của một ngôn ngữ lập trình
  - dùng một phần ngôn ngữ tự nhiên
  - bị phụ thuộc vào ngôn ngữ lập trình.
- Mọi ngôn ngữ lập trình đều có những thao tác cơ bản
  - xử lý, rẽ nhánh và lặp
  - tận dụng được các khái niệm trong ngôn ngữ lập trình,
- Dễ dàng nắm bắt nội dung thuật toán

## Mã giả (2)

---

*Một đoạn mã giả của thuật toán giải pt bậc hai*

**if** Delta > 0 **then begin**

$x1 = (-b - \sqrt{\text{delta}}) / (2 * a)$

$x2 = (-b + \sqrt{\text{delta}}) / (2 * a)$

xuất kết quả : phương trình có hai nghiệm là x1 và x2

**end**

**else**

**if** delta = 0 **then**

xuất kết quả : phương trình có nghiệm kép là  
 $-b / (2 * a)$

**else** {trường hợp delta < 0 }

xuất kết quả : phương trình vô nghiệm

# CHƯƠNG 5: CÁC NGÔN NGỮ LẬP TRÌNH

## Chương 5: Các ngôn ngữ lập trình

---

- Khái niệm ngôn ngữ lập trình
- Sự tiến hóa của ngôn ngữ lập trình
- Các ngôn ngữ lập trình tiêu biểu
- Biên dịch và thông dịch
- Các phương pháp lập trình

# Khái niệm ngôn ngữ lập trình (1)

---

- Con người “nói chuyện” với nhau:
  - Ngôn ngữ tự nhiên
- Con người “nói chuyện” với máy tính:
  - Ngôn ngữ lập trình

# Khái niệm ngôn ngữ lập trình (2)

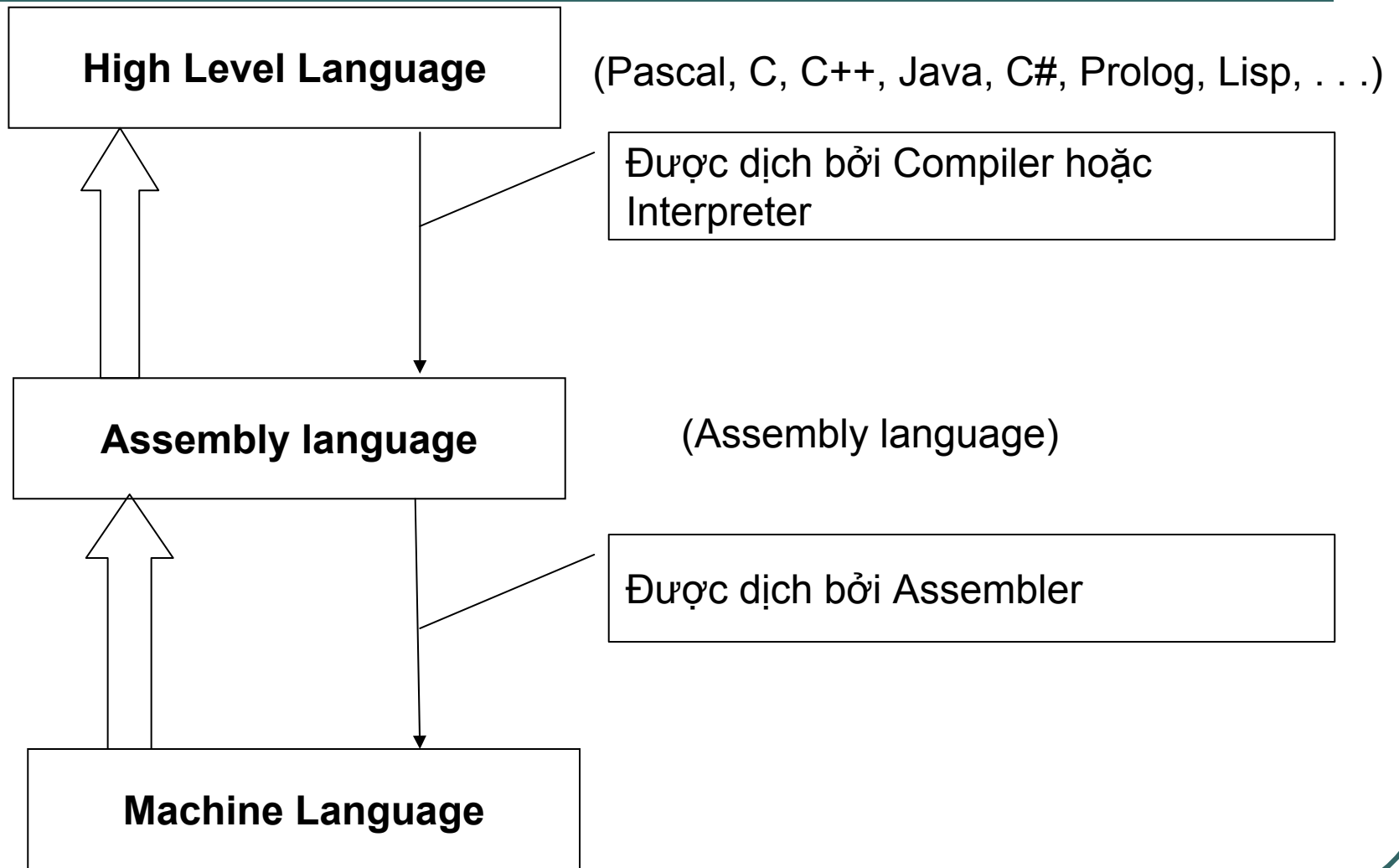
---

- **Ngôn ngữ lập trình (Programming language)** là một hệ thống dùng để viết các chương trình cho máy tính bao gồm:
  - Một tập hữu hạn các **kí hiệu**
  - Một tập hữu hạn các **qui ước** về
    - **Cú pháp** (Syntax): một chuỗi các kí hiệu hợp lệ
    - **Ngữ nghĩa** (Semantic): ý nghĩa của các kí hiệu
- **Chương trình (Program)** là một tập **lệnh** (instruction) điều khiển việc xử lý **dữ liệu** (data) của máy tính

**Program = Data Structures + Algorithms**

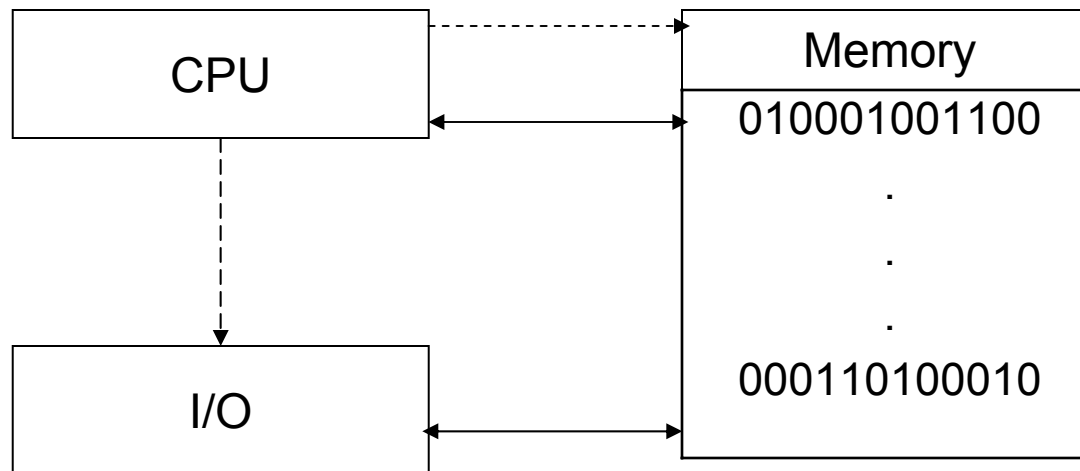


# Sự tiến hóa của ngôn ngữ lập trình



# Machine Language (1)

- Là ngôn ngữ duy nhất được máy tính hiểu trực tiếp
- Sử dụng hệ thống số nhị phân để biểu diễn lệnh (Instruction)
- Instruction = mẫu bit (bit pattern) được hiểu bởi phần cứng máy tính
  - Instruction: Opcode | Operands
  - Chương trình là một dãy các chỉ thị



# Machine Language (2)

---

## 👉 Ưu điểm

- ✓ Chương trình viết bằng ngôn ngữ máy thực thi nhanh

## 👉 Nhược điểm

- ✓ Lập trình khó
- ✓ Phụ thuộc cấu hình phần cứng của máy
- ✓ Dễ dẫn đến sai sót
- ✓ Khó bảo trì và sửa lỗi

# Assembly Language (1)

---

Dùng Alphanumeric code để biểu diễn lệnh và địa chỉ ô nhớ thay vì bit pattern

- Ví dụ 1:

**MOV r<sub>0</sub>,B            #move B into register r<sub>0</sub>**

**ADD r<sub>0</sub>,C            #add**

**MOV A, r<sub>0</sub> #store**

- Ví dụ 2: Tính  $(A+B)^2 = A^2 + 2AB + B^2$

**LOAD    A**

**MULT    A**

**STORE   H1        # H1 chứa A<sup>2</sup>**

**LOAD    A**

**MULT    B**

**MULT    2**

**STORE   H2        # H2 chứa 2AB**

**LOAD    B**

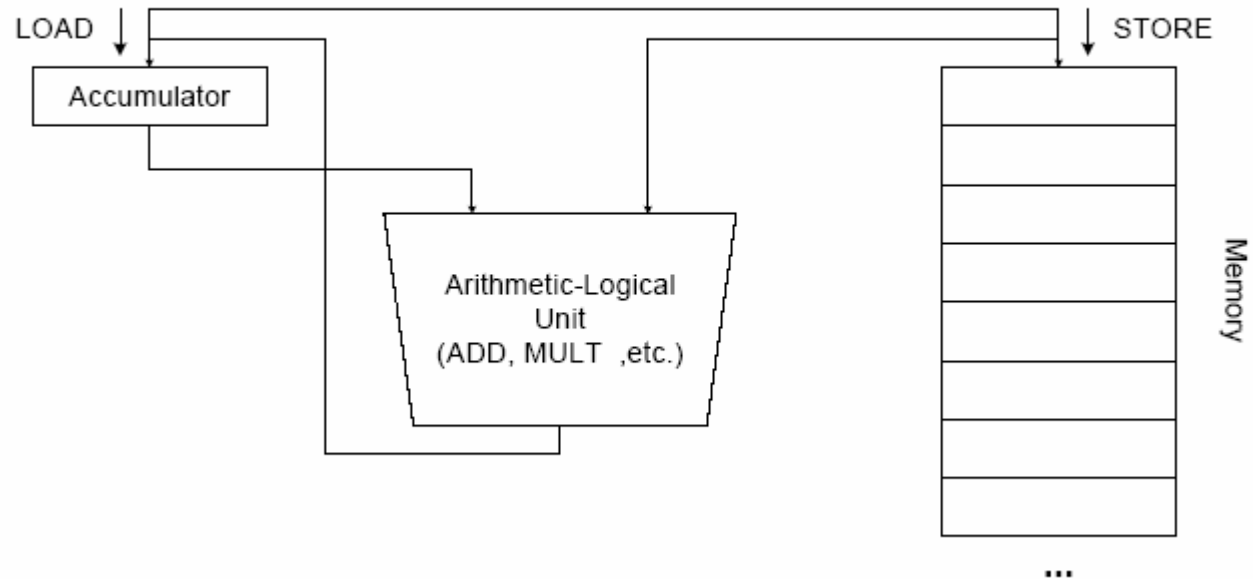
**MULT    B**

**ADD      H1**

**ADD      H2        # H2 chứa kết quả**

# Assembly Language (2)

---



# Assembly Language (3)

---

- Chương trình được viết bằng ngôn ngữ Assembly trước khi được thực thi bởi CPU nó phải được dịch sang mã máy (machine code).



- Ưu điểm
  - Việc lập trình nhanh hơn, và chương trình đọc dễ hiểu hơn ngôn ngữ mã máy
  - Người lập trình dễ nhớ lệnh hơn dùng mã máy
- Nhược điểm
  - Chương trình phụ thuộc bộ xử lý
  - Chương trình chạy chậm hơn khi được viết bằng mã máy

# Ngôn ngữ lập trình cấp cao

- Thân thiện với người sử dụng, bởi vì nó dùng các khái niệm, thuật ngữ gần với ngôn ngữ tự nhiên.

- Ví dụ 1:

```
if A > B then print "A larger than B"
else print "A less than B"
```

- Ví dụ 2:

```
int fact(int n) {
 if (n==0)
 return 1
 else
 return n*fact(n-1)
}
```

$$n! = \begin{cases} 1 & n=0 \\ n*(n-1)! & n>0 \end{cases}$$

- Độc lập với các hệ máy
- Chương trình dễ đọc, dễ hiểu, dễ bảo trì
- Khả chuyển trên nhiều hệ máy khác nhau
- Chương trình phải được dịch sang mã máy trước khi được thực thi

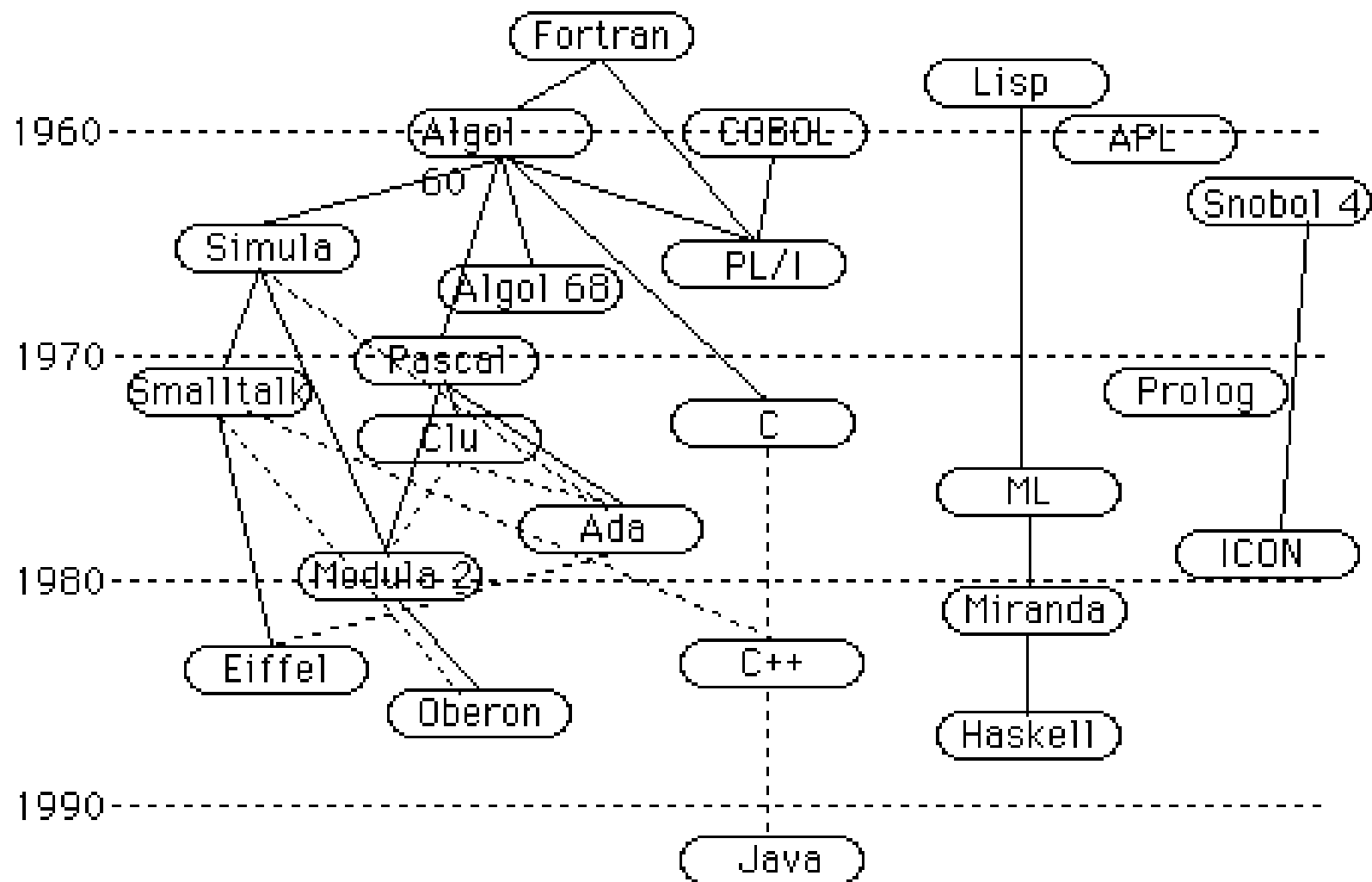
# Một số ngôn ngữ lập trình cấp cao

---

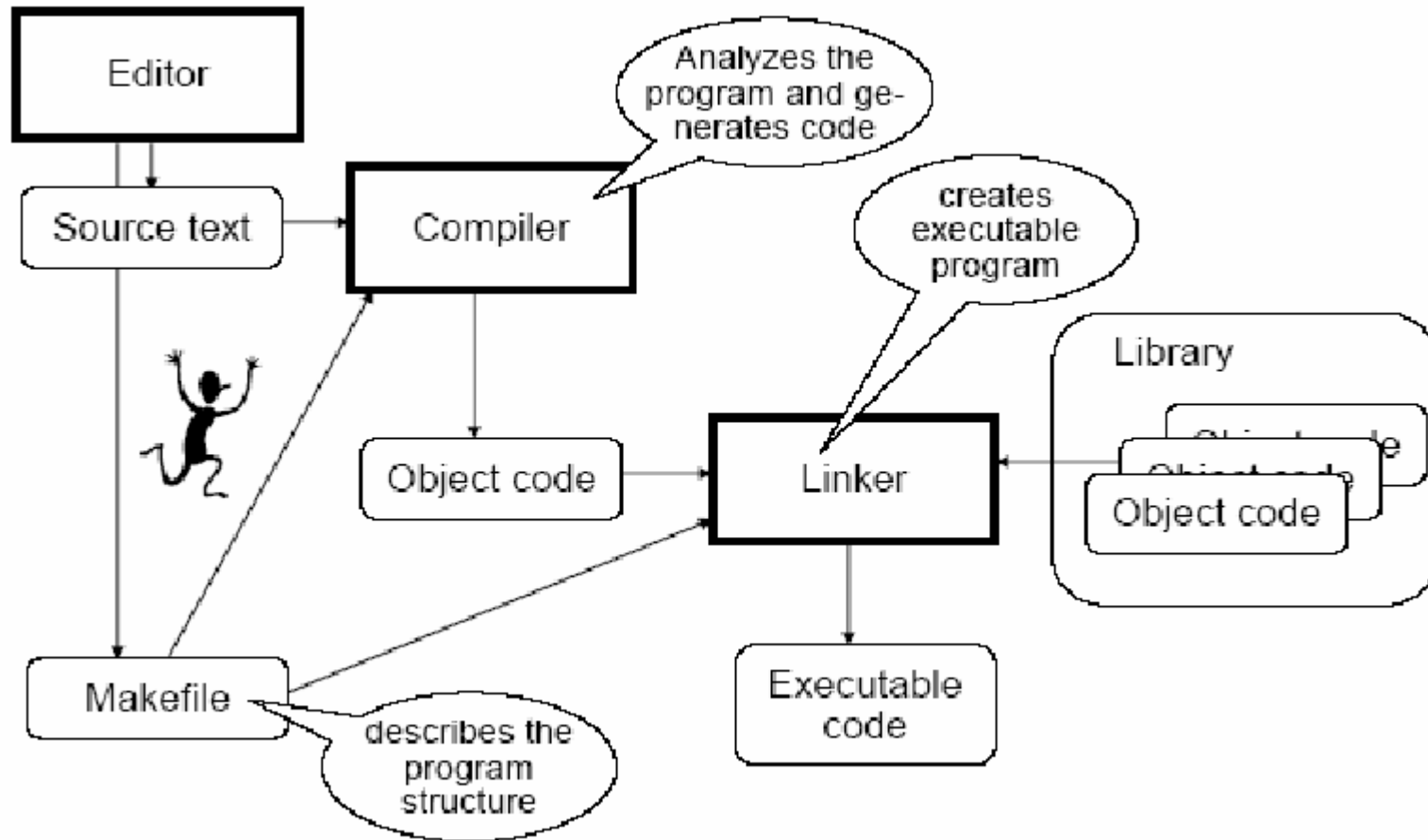
- BASIC, Beginner's All-Purpose Symbolic Instruction Code
  - John Kermeny và Thomas Kurtz (1964) tại trường đại học Dartmouth.
- COBOL, COmmon Business Oriented Language,
  - Bộ quốc phòng Hoa Kỳ (1960) hình thức tựa tiếng Anh
- C
  - Dennis Ritchie, phòng thí nghiệm Bell vào năm 1972
- FORTRAN, FORmula TRANslator
  - IBM, John Backus (1957) dành cho khoa học, kỹ sư & toán học
- PASCAL
  - Niklaus Wirth, Zurich, Thụy Sĩ, 1968, giảng dạy lập trình.
  - Blaise Pascal (1623 - 1662) người đầu tiên tạo ra máy tính



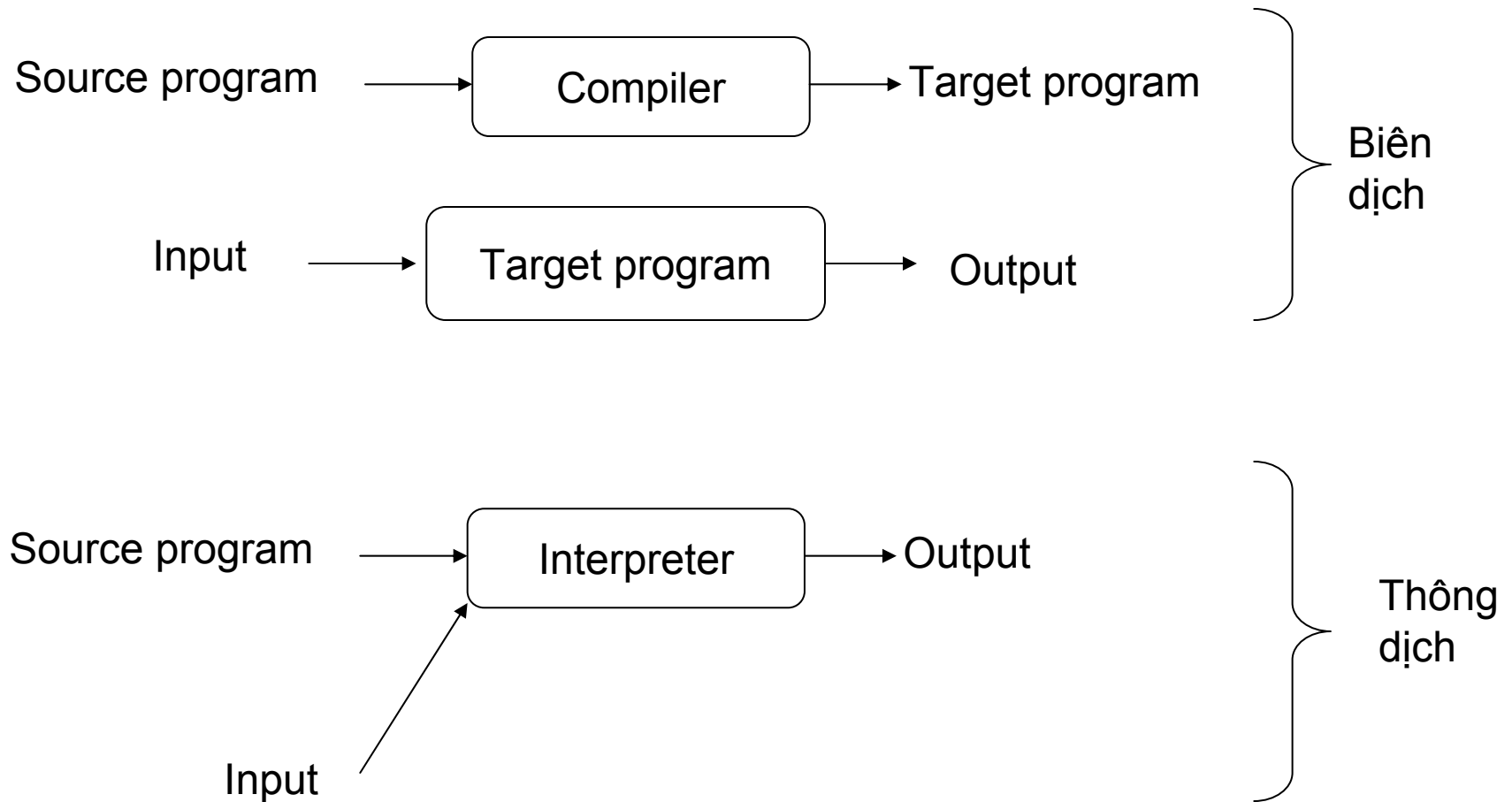
# Lịch sử phát triển các NNLT



# Quá Trình Dịch Một Chương Trình



# Thông dịch & Biên dịch (1)



## Thông dịch & Biên dịch (2)

| <b>Compiler</b>                                                                                                                                                                      | <b>Interpreter</b>                                                                         |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------|
| Dịch toàn bộ chương trình nguồn thành mã đối tượng.                                                                                                                                  | Dịch và thực thi từng dòng lệnh một của chương trình nguồn.                                |
| Mã đối tượng được lưu lại thành các tập tin .Obj để sử dụng cho các lần thực thi chương trình sau đó, mà không cần phải dịch lại chương trình → làm cho chương trình chạy nhanh hơn. | Mỗi lần thực thi chương trình đều phải dịch lại mã nguồn → làm cho chương trình chạy chậm. |
| Trong quá trình dịch, nếu phát hiện lỗi, compiler sẽ hiển thị lỗi của toàn bộ chương trình nguồn.                                                                                    | Hiển thị lỗi từng dòng lệnh của chương trình nguồn                                         |

# Các phương pháp lập trình

---

- Lập trình cấu trúc (Structural Programming)
  - Tính đơn thể,
  - Cấu trúc điều khiển
  - Vào/Ra đơn.
  - Dễ: viết, đọc, hiểu, kiểm lỗi và hiệu chỉnh.
- Lập trình hướng đối tượng (Object Oriented Programming – OOP)
  - Phương pháp kết hợp cả dữ liệu và các câu lệnh của chương trình.
  - Tạo ra các đối tượng (object)

TRƯỜNG.....

KHOA.....

.....o0o.....

# Tin học đại cương

## CHƯƠNG 1

# THÔNG TIN VÀ XỬ LÝ THÔNG TIN

### I. THÔNG TIN.

1. Khái niệm về thông tin, phân loại thông tin.
2. Đơn vị đo thông tin.
3. Mã hóa thông tin rời rạc.

### II. XỬ LÝ THÔNG TIN.

1. Sơ đồ tổng quát của một quá trình xử lý thông tin.
2. Xử lý thông tin bằng máy tính điện tử.

### III. TIN HỌC.

1. Các lĩnh vực nghiên cứu của tin học.
2. Ứng dụng của tin học.
3. Máy tính điện tử và lịch sử phát triển.

## I. THÔNG TIN

### 1. Khái niệm về thông tin, phân loại thông tin

*Dữ liệu (data) là các sự kiện không có cấu trúc, không có ý nghĩa rõ ràng, cho đến khi chúng được tổ chức theo một tiến trình tính toán nào đó.*

*Thông tin (Information) là một khái niệm trừu tượng được thể hiện qua các thông báo, các biểu hiện ... đem lại một nhận thức chủ quan cho một đối tượng nhận tin. Thông tin là dữ liệu đã được xử lý xong, mang ý nghĩa rõ ràng. Thông tin cũng có thể bị diễn đạt sai lệch, xuyên tạc do tác động cố ý hay vô ý của con người hay sinh vật khác.*

Một hệ thống thông tin (information system) là một tiến trình ghi nhận dữ liệu, xử lý nó và cung cấp tạo nên dữ liệu mới có ý nghĩa thông tin, liên quan một phần đến một tổ chức, để trợ giúp các hoạt động liên quan đến tổ chức.



### 2. Đơn vị đo thông tin

Đơn vị dùng để đo thông tin gọi là bit. Một bit tương ứng với một chỉ thị hoặc một thông báo nào đó về 1 sự kiện có trong 2 trạng thái có số đo khả năng xuất hiện đồng thời là Tắt(Off) / Mở(On) hay Đúng(True) / Sai(False).

Ví dụ 1. Một mạch đèn có 2 trạng thái là:

- Tắt (Off) khi mạch điện qua công tắc là hở
- Mở (On) khi mạch điện qua công tắc là đóng

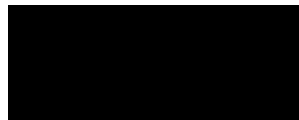
Số học nhị phân (sẽ giới thiệu ở chương 3) sử dụng hai số hạng 0 và 1. Vì khả năng sử dụng hai số 0 và 1 như nhau nên một chỉ thị chỉ gồm 1 chữ số nhị phân có thể xem như là chứa đơn vị thông tin nhỏ nhất.

Bit là chữ viết tắt của BInary digiT. Trong tin học, người ta thường sử dụng các đơn vị đo thông tin lớn hơn sau:

**Bảng 1. Bảng đơn vị đo thông tin**

| Tên gọi   | Ký hiệu | Giá trị                 |
|-----------|---------|-------------------------|
| Byte      | B       | 8 bit                   |
| KiloByte  | KB      | $2^{10}$ B = 1024 Bytes |
| MegaByte  | MB      | $2^{20}$ B              |
| GigaByte  | GB      | $2^{30}$ B              |
| TetraByte | TB      | $2^{40}$ B              |

Năm 1948, nhà bác học Shannon đã đưa ra công thức sau để tính lượng thông tin, bằng cách đo khả năng xuất hiện các sự kiện trong một thông báo, ký hiệu là H và gọi là Entropi :



trong đó :

n là số sự kiện lớn nhất có khả năng xuất hiện

Ví dụ 2. Gieo ngẫu nhiên 1 đồng xu, có thể xảy ra 1 trong 2 sự kiện trong mỗi lần gieo là khả năng xuất hiện mặt sấp hoặc mặt ngửa. Vì đồng xu xem như đồng chất, nên xác suất xuất hiện của mỗi sự kiện là  $p = 1/2$  . Vậy lượng tin sẽ là :



Ví dụ 3. Tương tự như trên nếu ta gieo 1 con súc sắc, thì  $n = 6$  và  $p = 1/6$

Lượng tin H sẽ là : 0



Ví dụ 4. Một trạm khí tượng X bằng việc phân tích nhiều số liệu khác nhau đã đưa ra dự đoán khả năng có mưa rơi một vùng nào đó vào ngày mai là 75%.

Suy ra khả năng không có mưa sẽ là  $100\% - 75\% = 25\%$ . Như vậy thông báo trên có lượng tin theo Shannon là :





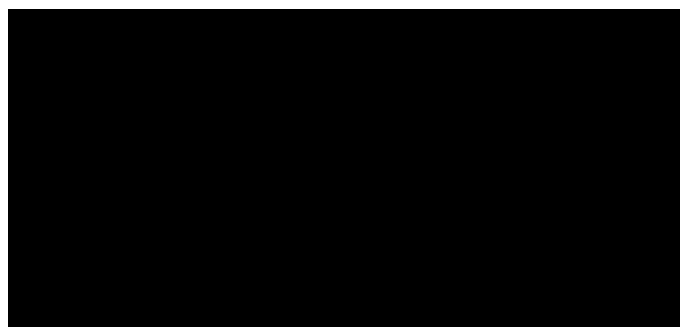
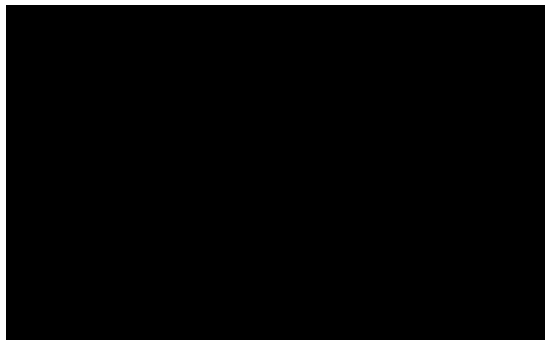
So sánh 3 ví dụ trên, ta thấy xác suất xuất hiện một tin càng thấp thì lượng tin càng *cao* vì *mức độ bất ngờ của nó càng lớn và ngược lại*. Tuy nhiên, không phải lúc nào ta cũng xác định được xác suất xuất hiện sự kiện nên việc áp dụng công thức Shannon bị hạn chế.

### 3. Mã hóa thông tin rời rạc :

Tất cả các thông tin ở dạng văn bản (text), chữ (character), số (number), ký hiệu (symbol), đồ họa (graphic), hình ảnh (image) hoặc âm thanh (sound) ... đều được gọi là các tín hiệu (signals). Tín hiệu có thể là liên tục hay rời rạc. Máy tính tương tự (Analog Computer) là máy tính chuyên dụng xử lý một số các tín hiệu liên tục như tín hiệu điện, âm thanh... Trong khi đó, hầu hết các dữ liệu mà chúng ta có được thường ở dạng các tín hiệu rời rạc để diễn tả các tín hiệu liên tục qua các số đo hữu hạn. Khi đưa các tín hiệu này vào máy tính, chúng được mã hóa theo các tín hiệu số (digital signal) nhằm giúp máy tính có thể hiểu được thông tin đưa vào. Đây là cơ sở thực tiễn của nguyên lý mã hoá thông tin rời rạc. Nguyên lý này tập trung các điểm chủ yếu sau :

- Tín hiệu liên tục có thể xem như một chuỗi xấp xỉ các tín hiệu rời rạc với chu kỳ lấy mẫu nhỏ ở mức độ chấp nhận được (Hình 1.).

- Tín hiệu rời rạc có thể được đặc trưng qua các bộ ký hiệu hữu hạn (chữ cái, chữ số, dấu, ...) gọi là phép mã hóa (encode) (Hình 2.). Mọi phép mã hóa đều có thể xây dựng trên bộ ký hiệu các chữ số, đặc biệt chỉ cần bộ ký hiệu gồm 2 chữ số là 0 và 1. Ngược với phép mã hoá gọi là phép giải mã (decode).



Tín hiệu rời rạc là tín hiệu có trục thời gian bị rời rạc hoá với chu kỳ lấy mẫu là  $T_s = 1/F_s$  , với  $F_s$  là tần số lấy mẫu. Tiếng nói con người thường có tần số  $F_s = 10$  kHz. Một ví dụ về thông tin rời rạc là hình trên phim khi được chiếu lên màn ảnh là các ảnh rời rạc xuất hiện với tốc độ 25 ảnh/giây. Mắt người không phân biệt sự rời rạc này nên có cảm tưởng hình ảnh là liên tục.

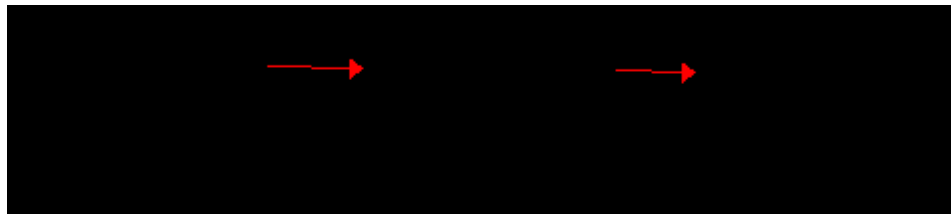
Mã hoá thông tin rời rạc là một khái niệm rất căn bản trong kỹ thuật máy tính.

## II. XỬ LÝ THÔNG TIN

### 1. Sơ đồ tổng quát của một quá trình xử lý thông tin

Mọi quá trình xử lý thông tin bằng máy tính hay bằng con người đều được thực hiện theo một qui trình sau :

Dữ liệu (data) được nhập ở đầu vào (input). Máy tính hay con người sẽ thực hiện quá trình xử lý nào đó để nhận được thông tin ở đầu ra (output). Quá trình nhập dữ liệu, xử lý và xuất thông tin đều có thể được lưu trữ (Hình 3.).



**Hình 1.3 Mô hình tổng quát quá trình xử lý thông tin**

## 2. Xử lý thông tin bằng máy tính điện tử

Thông tin là kết quả bao gồm nhiều quá trình xử lý các dữ liệu và thông tin có thể trở thành dữ liệu mới để theo một quá trình xử lý khác tạo ra thông tin mới hơn theo ý đồ của con người.

Con người có nhiều cách để có dữ liệu và thông tin. Người ta có thể lưu trữ thông tin qua tranh vẽ, giấy, sách báo, hình ảnh trong phim, băng từ, ... Trong thời đại hiện nay, khi lượng thông tin đến với chúng ta càng lúc càng nhiều thì con người có thể dùng một công cụ hỗ trợ cho việc lưu trữ, chọn lọc và xử lý lại thông tin gọi là *máy tính điện tử (computer)*. *Máy tính điện tử giúp con người tiết kiệm rất nhiều thời gian, công sức và tăng độ chính xác cao trong việc tự động hoá một phần hay toàn phần của quá trình xử lý dữ liệu hay thông tin.*

## III. TIN HỌC

### 1. Các lĩnh vực nghiên cứu của tin học

*Tin học (Informatics) được định nghĩa là ngành khoa học nghiên cứu các phương pháp, công nghệ và kỹ thuật xử lý thông tin tự động. Công cụ chủ yếu của tin học là máy tính điện tử và các thiết bị truyền tin khác. Việc nghiên cứu chính của tin học nhằm vào 2 kỹ thuật phát triển song song :*

- Kỹ thuật phần cứng (hardware engineering): nghiên cứu chế tạo các thiết bị, linh kiện điện tử, công nghệ vật liệu mới... hỗ trợ cho máy tính và mạng máy tính đầy mạnh khả năng xử lý toán học và truyền thông tin.

- Kỹ thuật phần mềm (software engineering): nghiên cứu phát triển các hệ điều hành, ngôn ngữ lập trình cho các bài toán khoa học kỹ thuật, mô phỏng, điều khiển tự động, tổ chức dữ liệu và quản lý hệ thống thông tin.

### 2. Ứng dụng của tin học

Tin học hiện đang được ứng dụng rộng rãi trong tất cả các ngành nghề khác nhau của xã hội từ khoa học kỹ thuật, y học, kinh tế, công nghệ sản xuất đến khoa học xã hội, nghệ thuật,... như:

- Tự động hóa văn phòng - Quản trị kinh doanh

- Thống kê - An ninh, quốc phòng

- Công nghệ thiết kế - Giáo dục

- Y học - Công nghệ in

- Nông nghiệp - Nghệ thuật, giải trí, v.v....

### 3. Máy tính điện tử và lịch sử phát triển

Do nhu cầu cần tăng độ chính xác và giảm thời gian tính toán, con người đã quan tâm chế tạo các công cụ tính toán từ xưa: bàn tính tay của người Trung Quốc, máy cơ học của nhà toán học Pháp Blaise Pascal (1623 - 1662), máy tính cơ học có thể cộng trừ nhân chia của nhà toán học Đức Gottfried Wilhelm von Leibniz (1646 - 1716), máy sai phân để tính các đa thức toán học, máy phân giải điều khiển bằng phiếu đục lỗ của Charles Babbage (1792 - 1871) ...

Tuy nhiên, máy tính điện tử thực sự hình thành bắt đầu vào thập niên 1950 và đến nay đã trải qua 5 thế hệ được phân loại theo sự tiến bộ về công nghệ điện tử và vi điện tử cũng như các cải tiến về nguyên lý, tính năng và loại hình của nó.

\* Thế hệ 1 (1950 - 1958): máy tính sử dụng các bóng đèn điện tử chân không, mạch riêng rẽ, vào số liệu bằng phiếu đục lỗ, điều khiển bằng tay. Máy có kích thước rất lớn, tiêu thụ năng lượng nhiều, tốc độ tính chậm khoảng 300 - 3.000 phép tính mỗi giây. Loại máy tính điển hình thế hệ 1 như EDVAC (Mỹ) hay BESM (Liên xô cũ), ...

\* Thế hệ 2 (1958 - 1964): máy tính dùng bộ xử lý bằng đèn bán dẫn, mạch in. Máy đã có chương trình dịch như Cobol, Fortran và hệ điều hành đơn giản. Kích thước máy còn lớn, tốc độ tính khoảng 10.000 đến 100.000 phép/s. Điển hình như loại IBM-1070 (Mỹ) hay MINSK (Liên xô cũ), ...

\* Thế hệ 3 (1965 - 1974): máy tính được gắn các bộ xử lý bằng vi mạch điện tử cỡ nhỏ có thể có được tốc độ tính khoảng 100.000 đến 1 triệu phép/s. Máy đã có các hệ điều hành đa chương trình, nhiều người dùng đồng thời hoặc theo kiểu chia thời gian. Kết quả từ máy tính có thể in ra trực tiếp ở máy in. Điển hình như loại IBM 360 (Mỹ) hay EC (Liên Xô cũ), ...

\* Thế hệ 4 (1974 đến nay): máy tính bắt đầu có các vi mạch đa xử lý có tốc độ tính hàng chục triệu đến hàng tỷ phép/giây. Giai đoạn này hình thành 2 loại máy tính chính : máy tính cá nhân để bàn (Personal Computer - PC) hoặc xách tay (Laptop hoặc Notebook computer) và các loại máy tính chuyên nghiệp thực hiện đa chương trình, đa vi xử lý ... hình thành các hệ thống mạng máy tính (Computer Networks), và các ứng dụng phong phú đa phương tiện.

\* Thế hệ 5 (1990 - nay): bắt đầu có các nghiên cứu tạo ra các máy tính mô phỏng các hoạt động của não bộ và hành vi con người, có trí khôn nhân tạo với khả năng tự suy diễn phát triển các tình huống nhận được và những hệ quản lý kiến thức cơ sở để giải quyết các bài toán đa dạng.

## CHƯƠNG 2

### NHỮNG NGUYÊN LÝ CƠ BẢN VÀ CẤU TRÚC TỔNG QUÁT

#### CỦA HỆ XỬ LÝ THÔNG TIN TỰ ĐỘNG

- I. [NHỮNG NGUYÊN LÝ THIẾT KẾ CƠ BẢN.](#)
  1. [Nguyên lý Turing.](#)
  2. [Nguyên lý Von-Neumann.](#)
- II. [CẤU TRÚC TỔNG QUÁT CỦA HỆ XỬ LÝ THÔNG TIN TỰ ĐỘNG.](#)
  1. [Phần cứng.](#)
  2. [Phần mềm.](#)

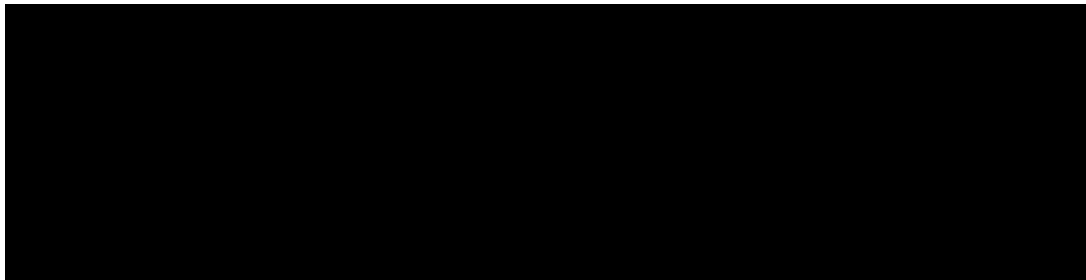
[BÀI ĐỌC THÊM KHÁI NIỆM VỀ MẠNG MÁY TÍNH.](#)

## I. NHỮNG NGUYÊN LÝ THIẾT KẾ CƠ BẢN

### 1. Nguyên lý Turing

Alan Mathison Turing (1912 - 1954) là một nhà toán học người Anh đã đưa ra một thiết bị tính đơn giản gọi là máy Turing. Về lý thuyết, mọi quá trình tính toán có thể được thì đều có thể mô phỏng lại trên máy Turing. Máy Turing gồm có (xem hình vẽ 2.1):

- Một bộ điều khiển trạng thái hữu hạn (finite control), trong đó có các trạng thái đặc biệt như trạng thái khởi đầu và trạng thái kết thúc.
- Một băng ghi (tape) chứa tín hiệu trong các ô.
- Một đầu đọc (head) và ghi có thể di chuyển theo 2 chiều trái hoặc phải một đơn vị.



Hình 2.1 Sơ đồ máy Turing

Đầu đọc/ghi mang chức năng thông tin nối giữa Bộ điều khiển hữu hạn và băng ghi. Đầu băng cách đọc dấu hiệu từ băng và cũng dùng nó để thay đổi dấu hiệu trên băng. Bộ kiểm soát vận hành theo từng bước riêng biệt; mỗi bước nó thực hiện 2 chức năng tùy thuộc vào trạng thái hiện tại của nó và tín hiệu hiện tại của băng:

1. Đặt bộ điều khiển ở trạng thái ban đầu  $q_1$ , băng trắng và đầu đọc/ghi chỉ vào ô khởi đầu.
2. Nếu:
  - (a) trạng thái hiện tại  $q$  trùng với trạng thái kết thúc  $q_0$  thì máy sẽ dừng.
  - (b) ngược lại, trạng thái  $q$  sẽ chuyển qua  $q'$ , tín hiệu trên băng  $s$  thành  $s'$  và đầu đọc dịch chuyển sang phải hoặc trái một đơn vị. Máy hoàn thành xong một bước tính toán và sẵn sàng cho bước tiếp theo.

### 2. Nguyên lý Von Neumann

Năm 1946, nhà toán học Mỹ John Von Neumann (1903 - 1957) đã đề ra một nguyên lý máy tính hoạt động theo một chương trình được lưu trữ và truy nhập theo địa chỉ. Nguyên lý này được trình bày ở một bài báo nổi tiếng nhan đề: Thảo luận sơ bộ về thiết kế logic của máy tính điện tử. Nội dung nguyên lý Von Neumann gồm :

- Máy tính có thể hoạt động theo một chương trình đã được lưu trữ.

Theo Von Neumann, chúng ta có thể tập hợp các lệnh cho máy thi hành theo một chương trình được thiết kế và coi đó như một tập dữ liệu. Dữ liệu này được cài vào trong máy và được truyền bằng xung điện. Đây là một cuộc cách mạng mới cho máy tính nhằm tăng tốc độ tính toán vào thời đó vì trước kia máy chỉ có thể nhận được các lệnh từ băng giấy hoặc bìa đục lỗ và nạp vào bằng tay. Nếu gặp bài toán lặp lại nhiều lần thì cũng tiếp tục bằng cách nạp lại một cách thủ công như vậy gây hạn chế trong tính toán sử dụng.

- Bộ nhớ được địa chỉ hóa

Mỗi dữ liệu đều có một địa chỉ của vùng nhớ chứa số liệu đó. Như vậy để truy nhập dữ liệu ta chỉ cần xác định địa chỉ của nó trên bộ nhớ.

- Bộ đếm của chương trình

Nếu mỗi câu lệnh phải dùng một vùng nhớ để chứa địa chỉ của câu lệnh tiếp theo thì không gian bộ nhớ sẽ bị thu hẹp. Để khắc phục hạn chế này, máy được gắn một thanh ghi để chỉ ra vị trí của lệnh tiếp theo cần được thực hiện và nội dung của nó tự động được tăng lên mỗi lần lệnh được truy cập. Muốn đổi thứ tự lệnh ta chỉ cần thay đổi nội dung thanh ghi bằng một địa chỉ của lệnh cần được thực hiện tiếp.

## II. CẤU TRÚC TỔNG QUÁT CỦA HỆ XỬ LÝ THÔNG TIN TỰ ĐỘNG

Mỗi loại máy tính có thể có các hình dạng hoặc cấu trúc khác nhau, tùy theo mục đích sử dụng nhưng, một cách tổng quát, máy tính điện tử là một hệ xử lý thông tin tự động gồm 3 phần chính:

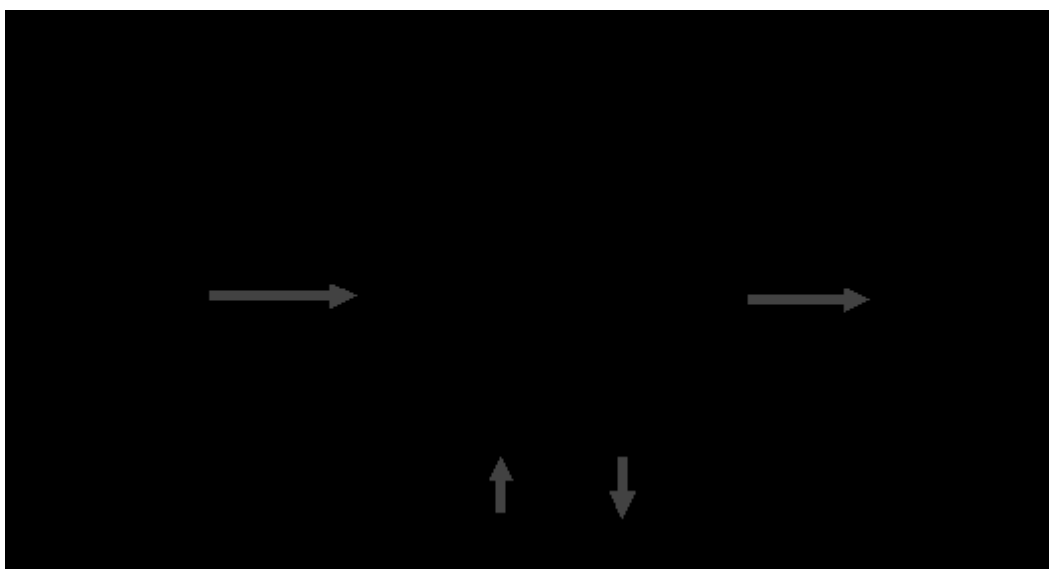
- \* Đơn vị xử lý trung ương (CPU - Central Processing Unit)
- \* Khối bộ nhớ (Memory): để chứa chương trình và dữ liệu
- \* Khối vào ra (Input/Output): bao gồm màn hình, máy in, bàn phím,...

### 1. Phần cứng (Hardware)

Phần cứng có thể được hiểu đơn giản là tất cả các phần trong một hệ máy tính mà chúng ta có thể thấy hoặc sờ được. Phần cứng gồm các thiết bị máy có thể thực hiện các chức năng sau:

- \* Nhập dữ kiện vào máy (input)
- \* Xử lý dữ kiện (processing)
- \* Xuất dữ kiện/ thông tin (output)

#### a. Sơ đồ cấu trúc phần cứng



Hình 2.2 Sơ đồ cấu trúc phần cứng

## b. Bộ nhớ

Bộ nhớ là thiết bị lưu trữ thông tin trong quá trình máy tính xử lý. Bộ nhớ bao gồm bộ nhớ trong và bộ nhớ ngoài.

*Bộ nhớ trong gồm ROM và RAM :*

- ROM (Read Only Memory) là Bộ nhớ chỉ đọc thông tin dùng để lưu trữ các chương trình hệ thống, chương trình điều khiển việc nhập xuất cơ sở (ROM-BIOS : ROM-Basic Input/Output System). Thông tin được giữ trên ROM thường xuyên ngay cả khi mất điện. Bộ nhớ này được các công ty sản xuất máy tính cài đặt sẵn trên máy thường có kích cỡ 16 KB(loại IBM PC XT), hoặc 32 KB, 64 KB (loại IBM PC AT). Người sử dụng máy tính không thể tự thay đổi nội dung thông tin trong ROM.

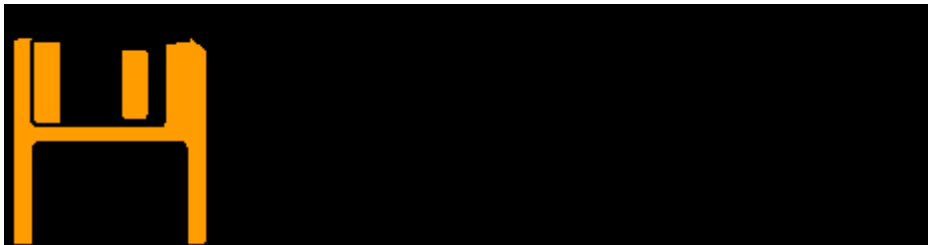
- RAM (Random Access Memory) là Bộ nhớ truy xuất ngẫu nhiên, được dùng để lưu trữ dữ kiện và chương trình trong quá trình thao tác và tính toán. RAM có đặc điểm là nội dung thông tin chứa trong nó sẽ mất đi khi mất điện hoặc tắt máy.

Dung lượng bộ nhớ cho các máy tính hiện nay (loại PC AT 486 trở lên) thông thường vào khoảng 4 MB, 8 MB, 16 MB, 32 MB hoặc 64 MB và có thể hơn nữa.

*Bộ nhớ ngoài: như đĩa từ, băng từ ... Để lưu trữ thông tin và có thể chuyển các tin này qua máy tính khác, người ta sử dụng các đĩa, băng từ như là các bộ nhớ ngoài. Các bộ nhớ này có dung lượng chứa lớn, không bị mất đi khi không có nguồn điện. Trên các máy vi tính phổ biến hiện nay có các loại đĩa từ sau:*

- Đĩa cứng (hard disk) : có nhiều loại dung lượng từ vài trăm đến vài ngàn MB, đến nay đã có đĩa cứng hơn 10 GB.

- Đĩa mềm (floppy disk) : phổ biến có 2 loại đĩa có đường kính 5.25 inches (dung lượng 360 KB hoặc 1.2 MB) và loại 3.5 inches (dung lượng 720 KB hoặc 1.44 MB)



**Hình 2.3 Đĩa máy tính loại 3.5 in và 5.25 in.**

- Đĩa quang (Compact disk): loại 4.72 inches có dung lượng vào khoảng 600 MB. Đĩa quang thường chỉ được đọc và không ghi được (CD-ROM) là thiết bị phổ biến với các phần mềm phong phú mang nhiều thông tin, hình ảnh, âm thanh không thể thiếu được trong các phương tiện đa truyền thông (multimedia).

Tốc độ quay của đĩa mềm 5.25 in. khoảng 300 vòng/phút, đĩa 3.5 in. khoảng 600 vòng/phút. Tốc độ quay của đĩa cứng rất cao thường đạt trên 3600 vòng/phút. Vì vậy, thông tin chứa trên đĩa cứng sẽ được truy cập nhanh hơn trên đĩa mềm rất nhiều.

Hiện nay trên thị trường còn có loại đĩa nén, có kích thước như loại đĩa 1.44 MB, nhưng có dung lượng đến 100 MB và dễ dàng mang đi các nơi.

## c. Bộ xử lý trung ương (CPU)

Bộ xử lý trung ương chỉ huy các hoạt động của máy tính theo lệnh và thực hiện các phép tính. CPU có 3 bộ phận chính: khối điều khiển, khối tính toán số học và logic, và một số thanh ghi.

· *Khối điều khiển (CU: Control Unit) là trung tâm điều hành máy tính. Nó có nhiệm vụ giải mã các lệnh, tạo ra các tín hiệu điều khiển công việc của các bộ phận khác của máy tính theo yêu cầu của người sử dụng hoặc theo chương trình đã cài đặt.*

· *Khối tính toán số học và logic (ALU: Arithmetic-Logic Unit) bao gồm các thiết bị thực hiện các phép tính số học (cộng, trừ, nhân, chia, ...), các phép tính logic (AND, OR, NOT, XOR) và các phép tính quan hệ (so sánh lớn hơn, nhỏ hơn, bằng nhau, ...)*

· Các thanh ghi (registers) được gắn chặt vào CPU bằng các mạch điện tử làm nhiệm vụ bộ nhớ trung gian. Các thanh ghi mang các chức năng chuyên dụng giúp tăng tốc độ trao đổi thông tin trong máy tính.

Ngoài ra, CPU còn được gắn với một đồng hồ (clock) hay còn gọi là bộ tạo xung nhịp. Tần số đồng hồ càng cao thì tốc độ xử lý thông tin càng nhanh. Thường thì đồng hồ được gắn tương xứng với cấu hình máy và có các tần số dao động (cho các máy PC 386 DX trở lên) là 33 MHz, 66 MHz, 100 MHz, 120 MHz, 133 MHz, ... hoặc cao hơn.

#### d. Các thiết bị xuất / nhập

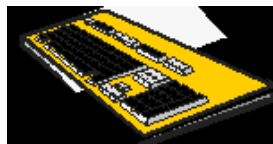


**Hình 2.4 Các bộ phận của một máy tính**

Các thiết bị nhập thông tin chính:


- Bàn phím (Keyboard): là thiết bị nhập dữ liệu và câu lệnh, bàn phím máy vi tính phổ biến hiện nay là một bảng chứa 104 phím có các tác dụng khác nhau.

Có thể chia làm 3 nhóm phím chính:




**Hình 2.5 Bàn phím**

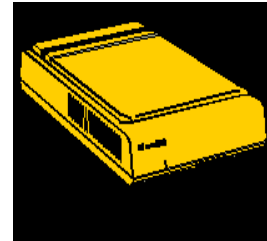
+ Nhóm phím đánh máy: gồm các phím chữ, phím số và phím các ký tự đặc biệt (~, !, @, #, \$, %, ^, &, ?, ...).

+ Nhóm phím chức năng (function key): gồm các phím từ F1 đến F12 và các phím khác như  (phím di chuyển từng điểm), phím PgUp (lên trang màn hình), PgDn (xuống trang màn hình), Insert (chèn), Delete (xóa), Home (về đầu), End (về cuối), ...

+ Nhóm phím đệm số (numeric keypad) như NumLock (cho các ký tự số), CapsLock (tạo các chữ in), ScrollLock (chế độ cuộn màn hình) thể hiện ở các đèn chỉ thị.

Ngoài 3 phím có đèn chỉ thị trên ta còn các nút điều khiển sau:

- Phím Shift: kèm với các phím chữ sẽ tạo ra chữ in hoa hoặc thường, đổi phím số thành các ký hiệu tương ứng trên nó.
- Phím  BackSpace: lùi điểm nháy đồng thời xóa ký tự đứng trước nó.
- Phím Enter: nút thi hành lệnh hoặc xuống hàng.
- Phím Space: thanh dài nhất, tạo ký tự rỗng.
- Phím PrintScreen: nút in nội dung màn hình ra giấy.
- Phím Pause: dừng thi hành chương trình.
- Phím Ctrl (Control) và Alt (Alternate): là phím dùng để phối hợp các phím khác tùy chương trình sử dụng.
- Phím Esc (Escape): phím thoát, được dùng khi có chỉ định rõ.
- Phím Tab: phím nhảy cách, thường 8 khoảng (khoảng nhảy có thể khác đi tùy chương trình hay người sử dụng định).
- **Con chuột (Mouse): là thiết bị cần thiết phổ biến hiện nay, nhất là các máy tính chạy trong môi trường Windows.**

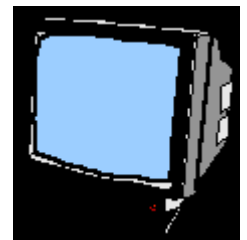


Con chuột có kích thước vừa nắm tay di chuyển trên một tấm phẳng (mouse pad) theo hướng nào thì đầu nháy hoặc mũi tên trên màn hình sẽ di chuyển theo hướng đó tương ứng với vị trí của của viên bi hoặc tia sáng (optical mouse) nằm dưới bụng của nó. Một số máy tính có con chuột được gắn trên bàn phím.



**Hình 2.6 Con chuột**

- **Máy quét (scanner): là thiết bị dùng để nhập văn bản hay hình vẽ, hình chụp vào máy tính.** Thông tin nguyên thủy trên giấy sẽ được quét thành các tín hiệu số tạo thành các tập tin ảnh (image file). Scanner đi kèm với phần mềm để nhận diện các tập tin ảnh hoặc văn bản.
- **Digitizer: dùng để nhập dữ liệu đồ họa theo tọa độ X-Y vào máy tính, thường được dùng trong vẽ bản đồ.**
- **Bút quang (Light pen): dùng nhập điểm bằng cách chấm lên màn hình.**
- **Touch screen: màn hình đặc biệt có thể dùng ngón tay để chạm lên các điểm.**



Các thiết bị xuất thông tin chính:

- **Màn hình (Screen hay Monitor): là thiết bị xuất chuẩn, dùng để thể hiện thông tin cho người sử dụng xem.**

Thông tin được thể hiện ra màn hình bằng phương pháp ánh xạ bộ nhớ (memory mapping), với cách này màn hình chỉ việc đọc liên tục bộ nhớ và hiển thị (display) bất kỳ thông tin nào hiện có trong vùng nhớ ra màn hình. Vì vậy để xuất thông tin ra màn hình ta chỉ cần xuất ra vùng nhớ tương ứng.



### Hình 2.8 Màn hình

- Trong chế độ văn bản, màn hình thể hiện 80 cột ký tự (đánh số từ 0 - 79) và 25 dòng (đánh số từ 0 - 24).

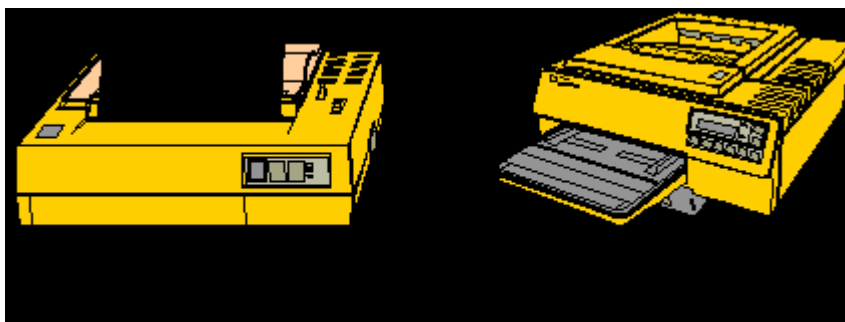
- Trong chế độ đồ họa, màn hình được chia thành các phần tử ảnh (pixel: picture element). Độ phân giải màn hình được xác định bằng tích số kích thước chiều ngang và chiều cao tính theo phần tử ảnh. Tích số này càng lớn thì màn hình càng mịn, rõ nét.

Màn hình phổ biến hiện nay trên thị trường là màn hình màu SVGA.

| Loại màn hình màu               | Độ phân giải (pixel) |
|---------------------------------|----------------------|
| CCA : Color Graphics Adapter    | 320 x 200            |
| EGA : Enhanced Graphics Adapter | 640 x 350            |
| VGA : Video Graphics Array      | 640 x 480            |
| SVGA : Super VGA                | 1020 x 768           |

- **Máy in (printer): là thiết bị xuất để đưa thông tin ra giấy. Máy in phổ biến** hiện nay là loại máy in ma trận điểm (dot matrix) loại 9 kim và 24 kim, máy in phun mực, máy in laser trắng đen hoặc màu.

Giấy in thường dùng là loại giấy in 80 cột (in được 80 ký tự, in nén được 132 ký tự) và loại giấy in khổ rộng in được 132 cột (in 132 ký tự, in nén 256 ký tự). Cả 2 loại giấy in đều có khả năng in 66 dòng/trang.



- Máy vẽ (plotter): loại máy đặc biệt dùng các bút màu để vẽ đồ họa, chữ ...

- Đĩa từ, băng từ (diskette, tape) ... : dùng để chứa thông tin xuất

## 2. Phần mềm (Software)

Muốn giải một bài toán trên máy tính điện tử người ta cần một chương trình hay phần mềm hướng dẫn máy tính thực hiện các thao tác cần thiết. Trước khi giới thiệu phần mềm ta cần hiểu các bước để giải một bài toán trên máy tính.

### a. Các giai đoạn giải một bài toán trên máy tính điện tử

Để giải quyết một bài toán trên máy tính điện tử, cần qua các giai đoạn:

- Tìm hiểu mục tiêu chính của bài toán: số liệu nhập và kết quả xuất.
- Xây dựng một chuỗi thao tác tính toán theo tuần tự, gọi là thuật giải.
- Lập chương trình diễn tả chi tiết các bước tính theo thuật giải
- Nhập chương trình vào máy tính, thông dịch và chạy thử để sửa chữa lỗi

- Thực hiện giải bài toán với số liệu thu thập được và ghi nhận kết quả
- Thử nghiệm với nhiều trường hợp khác nhau của bài toán
- Phân tích kết quả và hoàn chỉnh chương trình

Trong các bước trên, việc thiết kế thuật toán là giai đoạn quan trọng nhất.

## **b. Thuật toán (algorithm)**

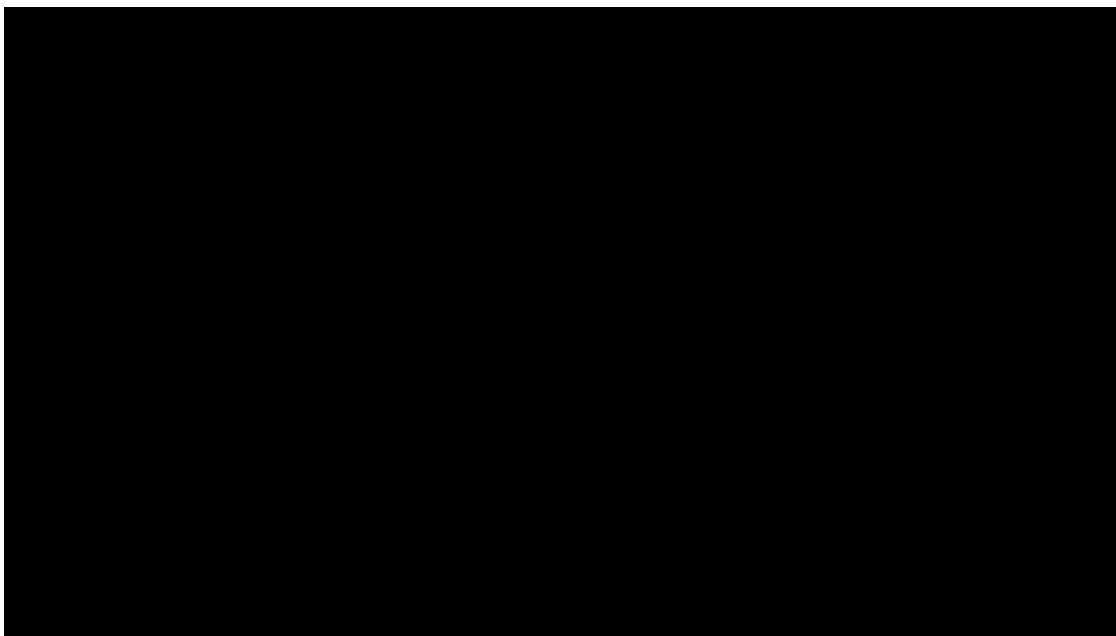
**Định nghĩa:** Thuật toán là một phương pháp trình bày các bước giải quyết một hay nhiều bài toán theo một tiến trình xác định.

Thuật toán có các đặc tính sau:

- *Tính xác định:* Các thao tác của thuật toán là rõ ràng và chắc chắn thực hiện được để dẫn đến kết quả nào đó.
- *Tính hữu hạn và dừng:* thuật toán phải có một số bước giải nhất định và cuối cùng phải có kết thúc ở điểm dừng.
- *Tính kết quả:* Với dữ liệu hợp lý, thuật toán phải cho kết quả thỏa yêu cầu.
- *Tính phổ dụng:* Thuật toán phải giải được nhiều bài toán có cùng cấu trúc với các dữ liệu khác nhau và đều dẫn đến một kết quả mong muốn.
- *Tính hiệu quả:* Thuật giải phải đơn giản, dễ hiểu trong các bước giải, tối thiểu hoá bộ nhớ và thời gian thực hiện.
- *Tính hình thức:* Các bước trong thuật toán là máy móc, nghĩa là nó phải thực hiện đúng như quy định mà không cần biết đến mục tiêu cuối cùng.

Thuật toán có thể diễn giải một cách trực quan bằng lưu đồ (flowchart). Lưu đồ được sử dụng thông dụng trong việc trình bày các bước cần thiết để giải quyết vấn đề qua các hình khối khác nhau và dòng dữ liệu giữa các bước được chỉ định đi theo các đường mũi tên.

Một số qui ước ký hiệu lưu đồ:



Ví dụ 2.1: Thuật toán giải phương trình bậc nhất :  $ax + b = 0$ , ta đi qua các bước:

- Bước 1: Nhập vào 2 hệ số  $a$  và  $b$ .

- Bước 2: Xét điều kiện  $a = 0$  ?

Nếu đúng là  $a = 0$ , thì đi đến bước 3. Nếu không, nghĩa là  $a \neq 0$ , thì đi đến bước 4.

- Bước 3: Xét điều kiện  $b = 0$  ?

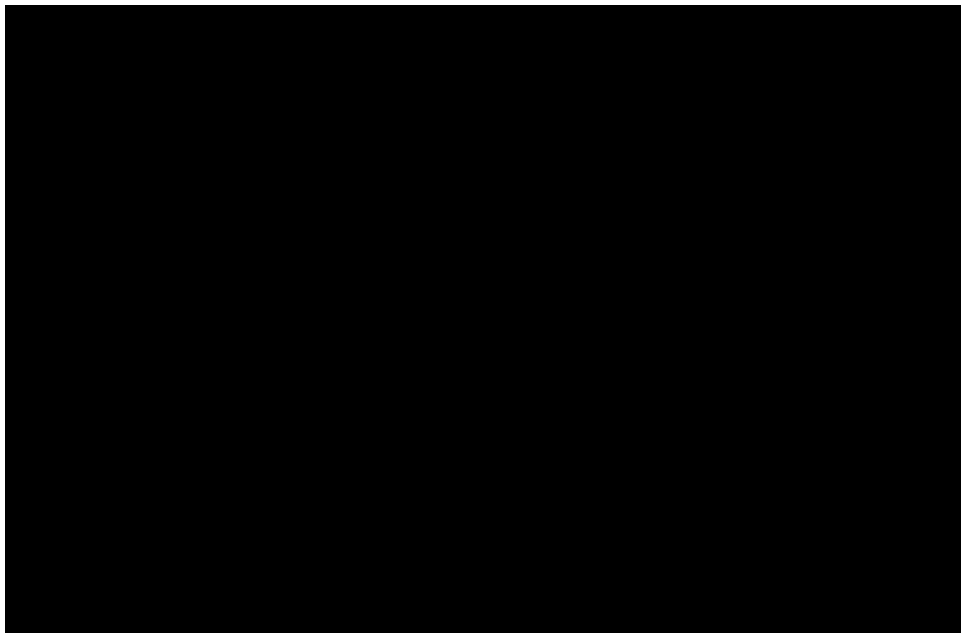
Nếu  $b = 0$ , thì báo phương trình có vô số nghiệm. Đi đến bước 5.

Nếu  $b \neq 0$ , thông báo phương trình vô nghiệm. Đi đến bước 5.

- Bước 4: Thông báo phương trình có một nghiệm duy nhất là  $x = -b/a$ .

- Bước 5: Ngưng dứt thuật toán

Ví dụ 2.2 Với bài toán như trong ví dụ 2.1, ta có thể trình bày với lưu đồ sau:



### c. Khái niệm và phân loại phần mềm

Phần mềm là một bộ chương trình các chỉ thị điện tử ra lệnh cho máy tính thực hiện một điều nào đó theo yêu cầu của người sử dụng. Chúng ta không thể thấy hoặc sờ được phần mềm, mặt dầu ta có thể hiển thị được chương trình trên màn hình hoặc máy in. Phần mềm có thể được ví như hồn của máy tính mà phần cứng của nó được xem như phần xác. Có 2 loại phần mềm cơ bản:

+ Phần mềm hệ điều hành (Operating System Software): là một bộ các câu lệnh để chỉ dẫn phần cứng máy tính và các phần mềm ứng dụng làm việc với nhau. Phần mềm hệ thống phổ biến hiện nay ở Việt nam là MS-DOS và Windows. Đối với mạng máy tính ta cũng có các phần mềm hệ điều hành mạng (Network Operating System) như Novell Netware, Unix, Windows NT, ...

+ Phần mềm ứng dụng (Application Software): rất phong phú và đa dạng, bao gồm những chương trình được viết ra cho một hay nhiều mục đích ứng dụng cụ thể như soạn thảo văn bản, tính toán, phân tích số liệu, tổ chức hệ thống, bảo mật thông tin, vẽ đồ họa, chơi games, ...

### BÀI ĐỌC THÊM

## KHÁI NIỆM VỀ MẠNG MÁY TÍNH

### (COMPUTER NETWORKS)

--- oOo ---

Một hệ thống gồm nhiều máy tính được kết nối nhau qua cáp truyền tin và có thể làm việc đồng thời với nhau gọi là mạng máy tính (Computer Networks). Ưu điểm của việc nối mạng là:

- Chia sẻ tài nguyên máy tính: bộ nhớ, phần mềm, dữ liệu, máy in, ...
- Trao đổi thông tin giữa các trạm làm việc (workstations) nhanh chóng, tiết kiệm thời gian và bảo đảm an toàn dữ liệu.

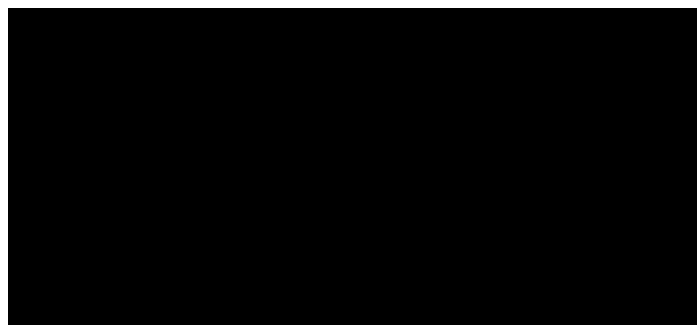
Hệ điều hành mạng (Networks Operating System - NOS) của mạng máy tính thường là các hệ phổ biến như UNIX, NOVELL NETWARE và WINDOWS NT.

Dựa vào khoảng cách nối kết, người ta phân biệt 2 loại:

+ Mạng cục bộ (Local Area Networks - LAN) là mạng máy tính nối với nhau trong một không gian hẹp như một phòng, tòa nhà, trường học, xí nghiệp,...

+ Mạng diện rộng (Wide Area Networks - WAN) để chỉ một loạt các máy hoặc các LAN được nối với nhau trên một phạm vi lớn hơn như liên tỉnh, liên vùng, quốc gia hay đa quốc gia.

Sơ đồ một mạng cục bộ đơn giản:



Các máy tính làm việc trên mạng cục bộ bao gồm:

- **Máy chủ (File Server) :** là một máy tính có bộ nhớ lớn, tốc độ truy xuất cao và có khả năng chứa nhiều tài nguyên. Máy chủ hệ điều hành mạng và các chương trình chung cho toàn mạng như các phần mềm xử lý dữ liệu, văn bản, phần mềm chuyên dụng, các phần mềm an toàn và bảo mật,... mỗi hệ thống mạng phải có ít nhất 1 máy chủ.

- **Trạm làm việc (WorkStation):** là các máy tính con được nối với nhau và vào máy chủ. Người làm việc trên các Trạm có thể sử dụng tài nguyên trên máy chủ và chia sẻ dữ liệu cho các đồng nghiệp.

- **Các thiết bị khác như máy in, máy vẽ,...** các máy này có thể nối trực tiếp vào máy chủ hoặc trạm làm việc tùy theo nhu cầu làm việc của hệ thống.

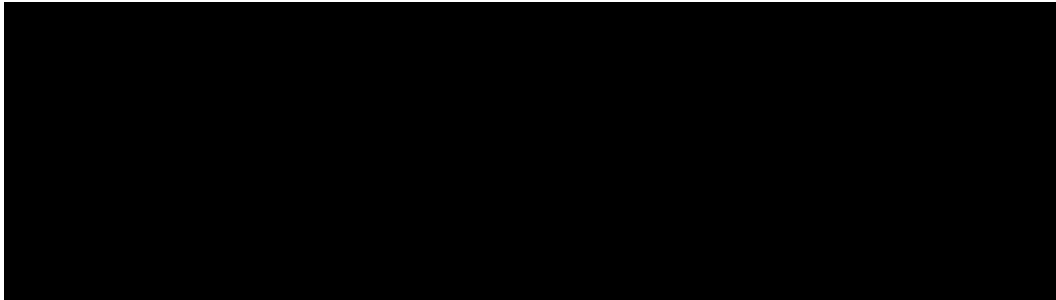
Có nhiều phương pháp kết nối máy tính với nhau theo cấu hình mạng (Topology). Mỗi loại đều có ưu khuyết điểm riêng. Thông dụng có các kiểu nối:

- **Kiểu Tuyến (Bus):** Tất cả máy tính nối với nhau theo một đường dây chung.



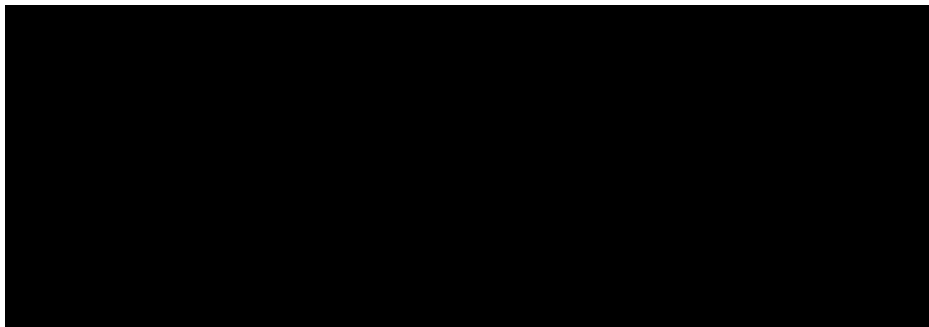
### Sơ đồ một LAN nối theo kiểu tuyến

- Kiểu Sao (Star): Tất cả các máy nối thành hình sao quanh máy chủ.



### Sơ đồ một LAN nối theo kiểu sao

- Kiểu Vòng (Ring) : Tất cả máy được nối vào một đường dây vòng chung.



### Sơ đồ một LAN nối theo kiểu vòng

Ngoài ra, ta có thể kết nối mạng theo kiểu phối hợp giữa tuyến và sao, hoặc giữa vòng và tuyến. Khi đó, file server sẽ được nối với thiết bị gọi là hộp công nhiều nhánh multiport hub cho phép kết nối cấp dẫn với trạm làm việc theo các cấu hình thiết kế khác nhau.

Người làm việc trên mạng bao gồm:

**Người quản trị mạng (Network supervisor):** là người giữ vai trò quản lý toàn bộ hệ thống mạng, chịu trách nhiệm cài đặt, sửa chữa, bảo trì mạng và toàn quyền phân phối tài nguyên truy xuất trên mạng cho người sử dụng.

**Người sử dụng mạng (Network Users):** là người làm việc trên từng Trạm làm việc riêng rẽ, có quyền sử dụng các tài nguyên trên mạng với một số quyền hạn nhất định mà người giám thị mạng cho phép.

**Nhóm trên mạng (Group):** là tập hợp một số người sử dụng có cùng chung một số yêu cầu sử dụng, chia sẻ và do đó được giám thị mạng cung cấp quyền như nhau.

## CHƯƠNG 3

### BIỂU DIỄN THÔNG TIN TRONG MÁY TÍNH ĐIỆN TỬ

- I. HỆ ĐẾM VÀ LOGIC MỆNH ĐỀ.
  - 1. Biểu diễn số trong các hệ đếm.
  - 2. Số học nhị phân.
  - 3. Mệnh đề logic.
- II. BIỂU DIỄN DỮ LIỆU.
  - 1. Biểu diễn số nguyên.
  - 2. Biểu diễn số thực.
  - 3. Biểu diễn ký tự.

#### PHU LUC.

#### BÀI ĐỌC THÊM CHUYÊN ĐỒI HỆ THỐNG SỐ DỰA TRÊN HỆ 8 VÀ HỆ 16.

### I. HỆ ĐẾM VÀ LOGIC MỆNH ĐỀ

#### 1. Biểu diễn số trong các hệ đếm

Hệ đếm là tập hợp các ký hiệu và qui tắc sử dụng tập ký hiệu đó để biểu diễn và xác định các giá trị các số. Mỗi hệ đếm có một số ký số (digits) hữu hạn. Tổng số ký số của mỗi hệ đếm được gọi là cơ số (base hay radix), ký hiệu là b.

Hệ đếm phổ biến hiện nay là hệ đếm La mã và hệ đếm thập phân.

##### a. Hệ đếm La mã

Hệ đếm La mã được xem như là hệ đếm có hệ thống đầu tiên của con người. Hệ đếm La mã sử dụng các ký hiệu ứng với các giá trị như sau:

$$I = 1 \quad V = 5 \quad X = 10 \quad L = 50 \quad C = 100 \quad D = 500 \quad M = 1000$$

Ký số La mã có một số qui tắc sau:

- Số lần n liên tiếp kế nhau của mỗi ký hiệu thể hiện giá trị ký hiệu tăng lên n lần. Số lần n chỉ là 1 hoặc 2 hoặc 3. Riêng ký hiệu M được phép xuất hiện 4 lần liên tiếp.

Ví dụ 3.1:  $III = 3 \times 1 = 3; \quad XX = 2 \times 10 = 20; \quad MMMM = 4000, \dots$

- Hai ký hiệu đứng cạnh nhau, nếu ký hiệu nhỏ hơn đứng trước thì giá trị của chúng sẽ là hiệu số của giá trị ký hiệu lớn trừ giá trị ký hiệu nhỏ hơn.

Ví dụ 3.2:  $IV = 5 - 1 = 4; \quad IX = 10 - 1 = 9; \quad CD = 500 - 100 = 400; \quad CM = 1000 - 100 = 900$

- Hai ký hiệu đứng cạnh nhau, nếu ký hiệu nhỏ đứng sau thì giá trị của chúng sẽ là tổng số của 2 giá trị ký hiệu.

Ví dụ 3.3:  $XI = 10 + 1 = 11; \quad DCC = 500 + 100 + 100 = 700$

Giá trị 3986 được thể hiện là:  $MMMCMCLXXXVI$

- Để biểu thị những số lớn hơn 4999 (MMMMCMXCIX), chữ số La mã giải quyết bằng cách dùng những vạch ngang đặt trên đầu ký tự. Một vạch ngang tương đương với việc nhân giá trị của ký tự

đó lên 1000 lần. Ví dụ  $M = 1000 \times 1000 = 10^6$ . Như vậy, trên nguyên tắc chữ số La mã có thể biểu thị các giá trị rất lớn. Tuy nhiên trong thực tế người ta thường sử dụng 1 - 2 vạch ngang là nhiều.

Hệ đếm La mã hiện ít được sử dụng trong tính toán hiện nay.

**b. Hệ đếm thập phân (decimal system)**

Hệ đếm thập phân hay hệ đếm cơ số 10 là một trong các phát minh của người Ả rập cổ, bao gồm 10 ký số theo ký hiệu sau:

$$0, 1, 2, 3, 4, 5, 6, 7, 8, 9$$

Quy tắc tính giá trị của hệ đếm này là mỗi đơn vị ở một hàng bất kỳ có giá trị bằng 10 đơn vị của hàng kế cận bên phải. Ở đây  $b = 10$ . Bất kỳ số nguyên dương trong hệ thập phân có thể thể hiện như là một tổng các chuỗi các ký số thập phân nhân cho 10 lũy thừa, trong đó số mũ lũy thừa được tăng thêm 1 đơn vị kể từ số mũ lũy thừa phía bên phải nó. Số mũ lũy thừa của hàng đơn vị trong hệ thập phân là 0.

Ví dụ 3.4: Số 5246 có thể được thể hiện như sau:

$$\begin{aligned} 5246 &= 5 \times 10^3 + 2 \times 10^2 + 4 \times 10^1 + 6 \times 10^0 \\ &= 5 \times 1000 + 2 \times 100 + 4 \times 10 + 6 \times 1 \end{aligned}$$

Thể hiện như trên gọi là ký hiệu mở rộng của số nguyên.

Vì  $5246 = 5000 + 200 + 40 + 6$


Như vậy, trong số 5246 : ký số 6 trong số nguyên đại diện cho giá trị 6 đơn vị (1s), ký số 4 đại diện cho giá trị 4 chục (10s), ký số 2 đại diện cho giá trị 2 trăm (100s) và ký số 5 đại diện cho giá trị 5 ngàn (1000s). Nghĩa là, số lũy thừa của 10 tăng dần 1 đơn vị từ trái sang phải tương ứng với vị trí ký hiệu số,


$$10^0 = 1 \quad 10^1 = 10 \quad 10^2 = 100 \quad 10^3 = 1000 \quad 10^4 = 10000 \dots$$

Mỗi ký số ở thứ tự khác nhau trong số sẽ có giá trị khác nhau, ta gọi là giá trị vị trí (place value).

Phần phân số trong hệ thập phân sau dấu chấm phân cách (theo qui ước của Mỹ) thể hiện trong ký hiệu mở rộng bởi 10 lũy thừa âm tính từ phải sang trái kể từ dấu chấm phân cách:



Tổng quát, hệ đếm cơ số  $b$  (  ,  $b$  là số nguyên dương) mang tính chất sau :

- Có  $b$  ký số để thể hiện giá trị số. Ký số nhỏ nhất là 0 và lớn nhất là  $b-1$ .
- Giá trị vị trí thứ  $n$  trong một số của hệ đếm bằng cơ số  $b$  lũy thừa  $n$  : 

Số  $N(b)$  trong hệ đếm cơ số ( $b$ ) thể hiện : 

trong đó, số N(b) có n+1 ký số chẵn ở phần nguyên và m ký số lẻ, sẽ có giá trị là :



Trong ngành toán - tin học hiện nay phổ biến 4 hệ đếm như sau :

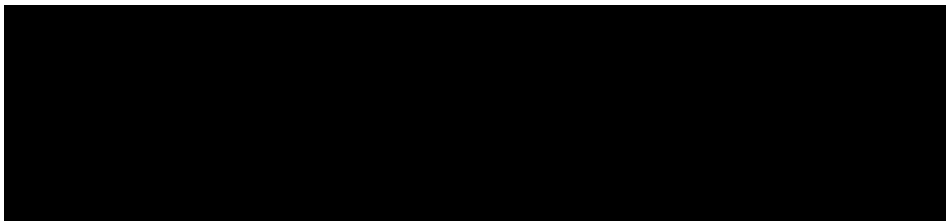
| Hệ đếm           | Cơ số | Ký số và trị tuyệt đối                         |
|------------------|-------|------------------------------------------------|
| Hệ nhị phân      | 2     | 0, 1                                           |
| Hệ bát phân      | 8     | 0, 1, 2, 3, 4, 5, 6, 7                         |
| Hệ thập phân     | 10    | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9                   |
| Hệ thập lục phân | 16    | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F |

**c. Hệ đếm nhị phân (binary number system)**

Với b = 2, chúng ta có hệ đếm nhị phân. Đây là hệ đếm đơn giản nhất với 2 chữ số là 0 và 1. Mỗi chữ số nhị phân gọi là BIT (viết tắt từ chữ BInary digiT). Hệ nhị phân tương ứng với 2 trạng thái của các linh kiện điện tử trong máy tính chỉ có: đóng (có điện) ký hiệu là 1 và tắt (không điện) ký hiệu là 0. Vì hệ nhị phân chỉ có 2 trị số là 0 và 1, nên khi muốn diễn tả một số lớn hơn, hoặc các ký tự phức tạp hơn thì cần kết hợp nhiều bit với nhau.

Ta có thể chuyển đổi hệ nhị phân theo hệ thập phân quen thuộc.

Ví dụ 3.6: Số  sẽ tương đương với giá trị thập phân là :



như vậy:

$$11101.11_{(2)} = 1x16 + 1x8 + 1x4 + 0x2 + 1x1 + 1x0.5 + 1x0.25 = 29.75_{(10)}$$

tương tự số 10101 (hệ 2) sang hệ thập phân sẽ là:

$$10101_{(2)} = 1x2^4 + 0x2^3 + 1x2^2 + 0x2^1 + 1x2^0 = 8 + 0 + 4 + 0 + 1 = 13_{(10)}$$

**d. Hệ đếm bát phân (octal number system)**

Nếu dùng 1 tập hợp 3 bit thì có thể biểu diễn 8 trị khác nhau : 000, 001, 010, 011, 100, 101, 110, 111. Các trị này tương đương với 8 trị trong hệ thập phân là 0, 1, 2, 3, 4, 5, 6, 7. Tập hợp các chữ số này gọi là hệ bát phân, là hệ đếm với b = 8 = 2<sup>3</sup>. Trong hệ bát phân, trị vị trí là lũy thừa của 8.



**e. Hệ đếm thập lục phân (hexa-decimal number system)**

Hệ đếm thập lục phân là hệ cơ số [redacted], tương đương với tập hợp 4 chữ số nhị phân (4 bit). Khi thể hiện ở dạng hexa-decimal, ta có 16 ký tự gồm 10 chữ số từ 0 đến 9, và 6 chữ in A, B, C, D, E, F để biểu diễn các giá trị số tương ứng là 10, 11, 12, 13, 14, 15. Với hệ thập lục phân, trị vị trí là lũy thừa của 16.

Ví dụ 3.8: [redacted]

Ghi chú: Một số chương trình qui định viết số hexa phải có chữ H ở cuối chữ số.

Ví dụ 3.9: Số 15 viết là FH.

Bảng qui đổi tương đương 16 chữ số đầu tiên của 4 hệ đếm

| Hệ 10 | Hệ 2 | Hệ 8 | Hệ 16 |
|-------|------|------|-------|
| 0     | 0000 | 00   | 0     |
| 1     | 0001 | 01   | 1     |
| 2     | 0010 | 02   | 2     |
| 3     | 0011 | 03   | 3     |
| 4     | 0100 | 04   | 4     |
| 5     | 0101 | 05   | 5     |
| 6     | 0110 | 06   | 6     |
| 7     | 0111 | 07   | 7     |
| 8     | 1000 | 10   | 8     |
| 9     | 1001 | 11   | 9     |
| 10    | 1010 | 12   | A     |
| 11    | 1011 | 13   | B     |
| 12    | 1100 | 14   | C     |
| 13    | 1101 | 15   | D     |
| 14    | 1110 | 16   | E     |
| 15    | 1111 | 17   | F     |

**f. Đổi một số nguyên từ hệ thập phân sang hệ b**

Tổng quát: Lấy số nguyên thập phân  $N(10)$  lần lượt chia cho  $b$  cho đến khi thương số bằng 0. Kết quả số chuyển đổi  $N(b)$  là các dư số trong phép chia viết ra theo thứ tự ngược lại.

Ví dụ 3.10: 

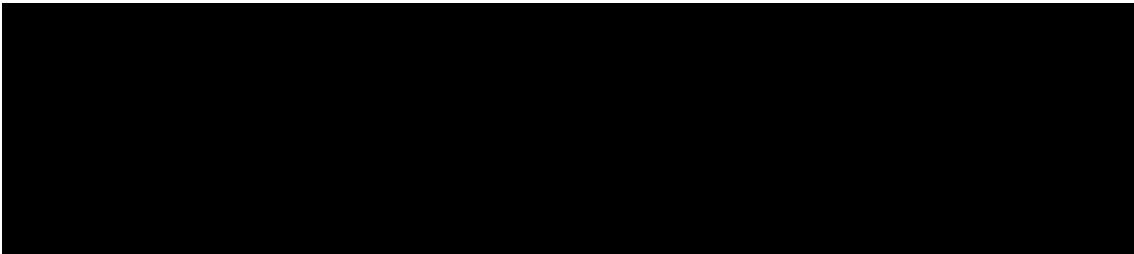
Dùng các phép chia 2 liên tiếp, ta có một loạt các số dư như sau:



Kết quả: 

**g. Đổi phần thập phân từ hệ thập phân sang hệ cơ số b**

Tổng quát: Lấy số nguyên thập phân N(10) lần lượt nhân cho b cho đến khi phần thập phân của tích số bằng 0. Kết quả số chuyển đổi N(b) là các số phần nguyên trong phép nhân viết ra theo thứ tự tính toán.



Kết quả: 

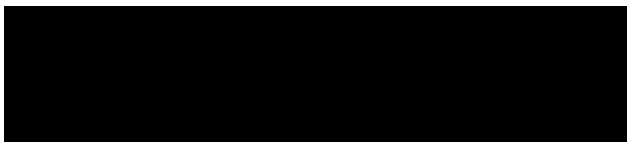
**2. Số học nhị phân**

Trong số học nhị phân chúng ta cũng có 4 phép toán cơ bản như trong số học thập phân là cộng, trừ, nhân và chia. Quy tắc của 2 phép tính cơ bản cộng và nhân:

| X | Y | X + Y | X * Y |
|---|---|-------|-------|
| 0 | 0 | 0     | 0     |
| 0 | 1 | 1     | 0     |
| 1 | 0 | 1     | 0     |
| 1 | 1 | 10    | 1     |

Ghi chú: Với phép cộng trong hệ nhị phân, 1 + 1 = 10, số 10 (đọc là một - không) chính là số 2 tương đương trong hệ thập phân. Viết 10 có thể hiểu là viết 0 nhớ 1. Một cách tổng quát, khi cộng 2 hay nhiều chữ số nếu giá trị tổng lớn hơn cơ số b thì ta viết phần lẻ và nhớ phần lớn hơn sang bên trái cạnh nó.

Ví dụ 3.12: Cộng 2 số 0101 + 1100 = ?

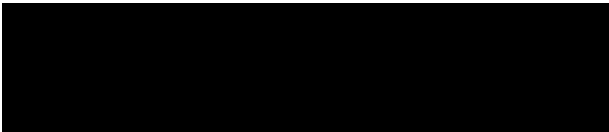


Ví dụ 3.13: Nhân 2 số 0110 x 1011 = ?

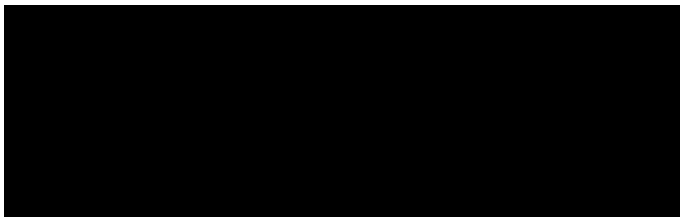


Phép trừ và phép chia là các phép toán đặc biệt của phép cộng và phép nhân.

Ví dụ 3.14: Trừ hai số



Ví dụ 3.15: Chia hai số



**Qui tắc 1:** Khi nhân một số nhị phân với  $2^n$  ta thêm n số 0 vào bên phải số nhị phân đó.

Ví dụ 3.16:  $1011 \times 2^3 = 1011000$

**Qui tắc 2:** Khi chia một số nguyên nhị phân cho  $2^n$  ta đặt dấu chấm ngăn ở vị trí n chữ số bên trái kể từ số cuối của số nguyên đó.

Ví dụ 3.17:  $10011111 : 2^3 = 100111.110$

### 3. Mệnh đề logic

Mệnh đề logic là mệnh đề chỉ nhận một trong 2 giá trị : Đúng (TRUE) hoặc Sai (FALSE), tương đương với TRUE = 1 và FALSE = 0.

*Qui tắc:* TRUE = NOT FALSE và FALSE = NOT TRUE

Phép toán logic áp dụng cho 2 giá trị TRUE và FALSE ứng với tổ hợp AND (và) và OR (hoặc) như sau:

| x     | y     | x AND y | x OR y |
|-------|-------|---------|--------|
| TRUE  | TRUE  | TRUE    | TRUE   |
| TRUE  | FALSE | FALSE   | TRUE   |
| FALSE | TRUE  | FALSE   | TRUE   |
| FALSE | FALSE | FALSE   | FALSE  |



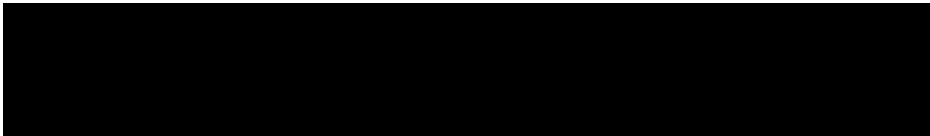
## II. BIỂU DIỄN DỮ LIỆU

Dữ liệu số trong máy tính gồm có số nguyên và số thực.

### 1. Biểu diễn số nguyên

Số nguyên gồm số nguyên không dấu và số nguyên có dấu.

- \* Số nguyên không dấu là số không có bit dấu như 1 byte = 8 bit, có thể biểu diễn [redacted] số nguyên dương, cho giá trị từ 0 (0000 0000) đến 255 (1111 1111).
- \* Số nguyên có dấu thể hiện trong máy tính ở dạng nhị phân là số dùng 1 bit làm bit dấu, người ta qui ước dùng bit ở hàng đầu tiên bên trái làm bit dấu (S): 0 là số dương và 1 cho số âm. Đơn vị chiều dài để chứa thay đổi từ 2 đến 4 bytes.



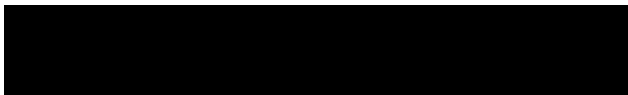
Ta thấy, với chiều dài 16 bit : bit đầu là bit dấu và 15 bit sau là bit số

Trị dương lớn nhất của dãy 2 bytes sẽ là: [redacted]

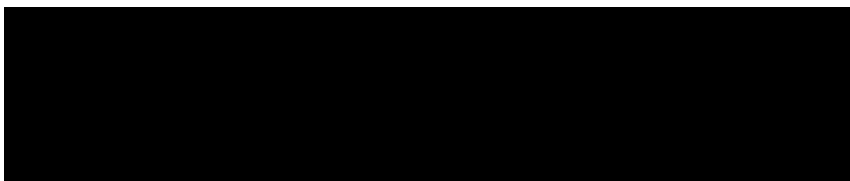
Trị âm lớn nhất trong dãy 2 bytes là [redacted]

Để thể hiện số âm trong hệ nhị phân ta có 2 khái niệm:

- Số bù 1: Khi đảo ngược tất cả các bit của dãy số nhị phân: 0 thành 1 và 1 thành 0, dãy số đảo đó gọi là số bù 1 của số nhị phân đó.



- Số bù 2: Số bù 2 của số N là số đảo dấu của nó (-N). Trong hệ nhị phân, số bù 2 được xác định bằng cách lấy số bù 1 của N rồi cộng thêm 1.



### 2. Biểu diễn số thực

Đối với các số thực (real number) là số có thể có cả phần lẻ hoặc phần thập phân. Trong máy tính, người ta biểu diễn số thực với số dấu chấm tĩnh (fixed point number) và số dấu chấm động (floating point number).

**a. Số dấu chấm tĩnh: thực chất là số nguyên (integers) là những số không có chấm thập phân**

**b. Số dấu chấm động: là số có chữ số phần lẻ không cố định. Mỗi số như vậy có thể trừ và xử lý trong máy tính ở dạng số mũ.**





Ghi chú: Dấu chấm thể hiện trong máy tính để phân biệt phần lẻ, dấu phẩy tượng trưng cho phần ngàn, được viết theo qui ước của Mỹ.

Tổng quát, số dấu chấm động được biểu diễn theo 3 phần :

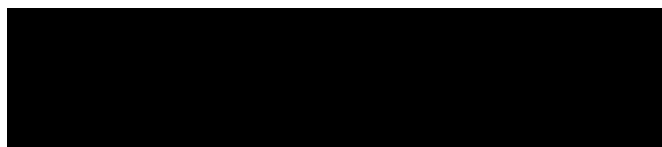
- phần dấu S (sign) : 0 cho + và 1 cho -
- phần định trị m (mantissa)
- phần mũ e (exponent), có thể là số nguyên dương (+) hoặc âm (-)


với một số X bất kỳ, có thể viết :

$$X = \pm m \cdot b^e = \pm m E e$$

Trong đó, b là cơ số qui ước, trị số mũ e có thể thay đổi tùy theo số vị trí cần dời dấu chấm để có lại trị số ban đầu. Khi dời dấu chấm sang n vị trí về phía trái (+n) hay phía phải (-n) thì số mũ e thay đổi lên đơn vị tương ứng

Để biểu diễn số có dấu chấm động, người ta dùng dãy 32 bit với hệ thống cơ số 16. Trong đó, 1 bit cho phần dấu, 7 bit cho phần mũ để biểu diễn phần đặc trị C (characteristic) và 24 bit cho phần định trị m.



Phần mũ có 7 bit = đặc trị C, tương ứng phần mũ e từ -64 đến +63

$$C = \text{số mũ biểu diễn} + 64$$

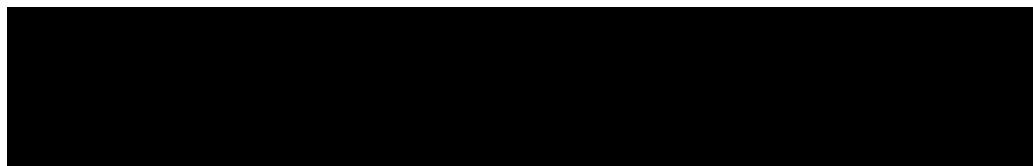
|           |      |      |      |     |     |     |    |    |     |     |     |
|-----------|------|------|------|-----|-----|-----|----|----|-----|-----|-----|
| Phần mũ e | - 64 | - 63 | - 62 | ... | - 2 | - 1 | 0  | 1  | ... | 62  | 63  |
| Đặc trị C | 0    | 1    | 2    | ... | 62  | 63  | 64 | 65 | ... | 126 | 127 |



Trong máy tính, số A sẽ được trữ theo vị trí nhớ 32 bit như sau :

Dấu A đặc trị C (7bit)

định trị m (24 bit)



### 3. Biểu diễn ký tự

Để có thể biểu diễn các ký tự như chữ cái in và thường, các chữ số, các ký hiệu... trên máy tính và các phương tiện trao đổi thông tin khác, người ta phải lập ra các bộ mã (code system) qui ước khác nhau dựa vào việc chọn tập hợp bao nhiêu bit để diễn tả 1 ký tự tương ứng, ví dụ các hệ mã phổ biến :

- Hệ thập phân mã nhị phân BCD (Binary Coded Decima) dùng 6 bit.

- Hệ thập phân mã nhị phân mở rộng EBCDIC (Extended Binary Coded Decimal Interchange Code) dùng 8 bit tương đương 1 byte để biểu diễn 1 ký tự.

- Hệ chuyển đổi thông tin theo mã chuẩn của Mỹ ASCII (American Standard Code for Information Interchange) là hệ mã thông dụng nhất hiện nay trong kỹ thuật tin học. Hệ mã ASCII dùng nhóm 7 bit hoặc 8 bit để biểu diễn tối đa 128 hoặc 256 ký tự khác nhau và mã hóa theo ký tự liên tục theo cơ số 16.

Hệ mã ASCII 7 bit, mã hoá 128 ký tự liên tục như sau:

- 0 : NUL (ký tự rỗng)
- 1 - 31 : 31 ký tự điều khiển
- 32 - 47 : các dấu trống SP (space) ! # \$ % & ( ) \* + , - . /
- 48 - 57 : ký số từ 0 đến 9
- 58 - 64 : các dấu : ; < = > ? @
- 65 - 90 : các chữ in hoa từ A đến Z
- 91 - 96 : các dấu [ \ ] \_ `
- 97 - 122 : các chữ thường từ a đến z
- 123 - 127 : các dấu { | } ~ DEL (xóa)

Hệ mã ASCII 8 bit (ASCII mở rộng) có thêm 128 ký tự khác ngoài các ký tự nêu trên gồm các chữ cái có dấu, các hình vẽ, các đường kẻ khung đơn và khung đôi và một số ký hiệu đặc biệt (xem phụ lục).

## PHỤ LỤC 3.1

BẢNG MÃ ASCII với 128 ký tự đầu tiên

| Hex      | 0        | 1         | 2        | 3       | 4       | 5       | 6        | 7          |
|----------|----------|-----------|----------|---------|---------|---------|----------|------------|
| <b>0</b> | NUL<br>0 | DLE<br>16 | SP<br>32 | 0<br>48 | @<br>64 | P<br>80 | `<br>96  | p<br>112   |
| <b>1</b> | SOH<br>1 | DC1<br>17 | !<br>33  | 1<br>49 | A<br>65 | Q<br>81 | a<br>97  | q<br>113   |
| <b>2</b> | STX<br>2 | DC2<br>18 | “<br>34  | 2<br>50 | B<br>66 | R<br>82 | b<br>98  | r<br>114   |
| <b>3</b> | ♥<br>3   | DC3<br>19 | #<br>35  | 3<br>51 | C<br>67 | S<br>83 | c<br>99  | s<br>115   |
| <b>4</b> | ♦<br>4   | DC4<br>20 | \$<br>36 | 4<br>52 | D<br>68 | T<br>84 | d<br>100 | t<br>116   |
| <b>5</b> | ♣<br>5   | NAK<br>21 | %<br>37  | 5<br>53 | E<br>69 | U<br>85 | e<br>101 | u<br>117   |
| <b>6</b> | ♠<br>6   | SYN<br>22 | &<br>38  | 6<br>54 | F<br>70 | V<br>86 | f<br>102 | v<br>118   |
| <b>7</b> | BEL<br>7 | ETB<br>23 | ‘<br>39  | 7<br>55 | G<br>71 | W<br>87 | g<br>103 | w<br>119   |
| <b>8</b> | BS<br>8  | CAN<br>24 | (<br>40  | 8<br>56 | H<br>72 | X<br>88 | h<br>104 | x<br>120   |
| <b>9</b> | HT<br>9  | EM<br>25  | )<br>41  | 9<br>57 | I<br>73 | Y<br>89 | I<br>105 | y<br>121   |
| <b>A</b> | LF<br>10 | SUB<br>26 | *<br>42  | :<br>58 | J<br>74 | Z<br>90 | j<br>106 | z<br>122   |
| <b>B</b> | VT<br>11 | ESC<br>27 | +<br>43  | ;<br>59 | K<br>75 | [<br>91 | k<br>107 | {<br>123   |
| <b>C</b> | FF<br>12 | FS<br>28  | ,<br>44  | <<br>60 | L<br>76 | \<br>92 | l<br>108 | <br>124    |
| <b>D</b> | CR<br>13 | GS<br>29  | -<br>45  | =<br>61 | M<br>77 | ]<br>93 | m<br>109 | }<br>125   |
| <b>E</b> | SO<br>14 | RS<br>30  | .<br>46  | ><br>62 | N<br>78 | ^<br>94 | n<br>110 | ~<br>126   |
| <b>F</b> | SI<br>15 | US<br>31  | /<br>47  | ?<br>63 | O<br>79 | _<br>95 | o<br>111 | DEL<br>127 |

**PHỤ LỤC 3.2**

**BẢNG MÃ ASCII với ký tự số 128 - số 255**

| Hex      | 8   | 9   | A   | B   | C   | D   | E   | F            |
|----------|-----|-----|-----|-----|-----|-----|-----|--------------|
| <b>0</b> | Ç   | É   | á   | ⌘   | ℒ   | ℓ   | α   | ≡            |
|          | 128 | 144 | 160 | 176 | 192 | 208 | 224 | 240          |
| <b>1</b> | ü   | æ   | í   | ⌘   | ℒ   | ⌘   | β   | ±            |
|          | 129 | 145 | 161 | 177 | 193 | 209 | 225 | 241          |
| <b>2</b> | é   | Æ   | ó   | ⌘   | ⌘   | ⌘   | Γ   | ≥            |
|          | 130 | 146 | 162 | 178 | 194 | 210 | 226 | 242          |
| <b>3</b> | â   | ô   | ú   |     | ⌘   | ℒ   | π   | ≤            |
|          | 131 | 147 | 163 | 179 | 195 | 211 | 227 | 243          |
| <b>4</b> | ä   | ö   | ñ   | ⌘   | -   | ℒ   | Σ   |              |
|          | 132 | 148 | 164 | 180 | 196 | 212 | 228 | 244          |
| <b>5</b> | à   | ò   | Ñ   | ⌘   | ⌘   | ⌘   | σ   | ∫            |
|          | 133 | 149 | 165 | 181 | 197 | 213 | 229 | 245          |
| <b>6</b> | å   | û   | ª   | ⌘   | ⌘   | ⌘   | μ   | ÷            |
|          | 134 | 150 | 166 | 182 | 198 | 214 | 230 | 246          |
| <b>7</b> | ç   | ù   | º   | ⌘   | ⌘   | ⌘   | τ   | ≈            |
|          | 135 | 151 | 167 | 183 | 199 | 215 | 231 | 247          |
| <b>8</b> | ê   | ÿ   | ¿   | ⌘   | ℒ   | ⌘   | Φ   | °            |
|          | 136 | 152 | 168 | 184 | 200 | 216 | 232 | 248          |
| <b>9</b> | ë   | ÿ   | ⌘   | ⌘   | ⌘   | ⌘   | Θ   | •            |
|          | 137 | 153 | 169 | 185 | 201 | 217 | 233 | 249          |
| <b>A</b> | è   | ÿ   | ⌘   | ⌘   | ℒ   | ⌘   | Ω   | •            |
|          | 138 | 154 | 170 | 186 | 202 | 218 | 234 | 250          |
| <b>B</b> | ï   | ç   | ½   | ⌘   | ⌘   | ■   | δ   | √            |
|          | 139 | 155 | 171 | 187 | 203 | 219 | 235 | 251          |
| <b>C</b> | î   | £   | ¼   | ⌘   | ⌘   | ■   | ∞   | <sup>n</sup> |
|          | 140 | 156 | 172 | 188 | 204 | 220 | 236 | 252          |
| <b>D</b> | ì   | ¥   | ¡   | ⌘   | =   | ■   | φ   | <sup>2</sup> |
|          | 141 | 157 | 173 | 189 | 205 | 221 | 237 | 253          |
| <b>E</b> | Ë   | £   | «   | ⌘   | ⌘   | ■   | ε   | ■            |
|          | 142 | 158 | 174 | 190 | 206 | 222 | 238 | 254          |
| <b>F</b> | Ë   | f   | »   | ⌘   | ⌘   | ■   | ∩   | 255          |
|          | 143 | 159 | 175 | 191 | 207 | 223 | 239 |              |

**BÀI ĐỌC THÊM**



## CHUYỂN ĐỔI HỆ THỐNG SỐ DỰA TRÊN HỆ 8 VÀ HỆ 16

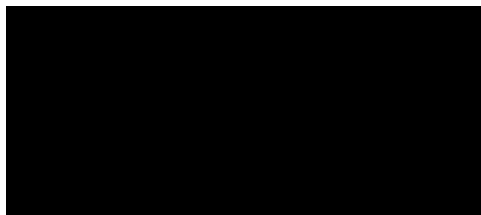
--- oOo ---

Trong phần bài giảng, chúng ta đã làm quen với cách chuyển đổi giữa hệ 2 và hệ 10. Tuy nhiên, ở những trị số lớn và dài thì làm cách trên trở nên rất phức tạp và dễ nhầm lẫn, ví dụ :



Trong ví dụ thứ nhất ta phải liên tiếp làm nhiều phép nhân và ở ví dụ thứ hai, ta lại thực hiện nhiều phép chia liên tiếp.

Người ta đưa ra hệ thống số trung gian là hệ 8 và hệ 16 để giải quyết:

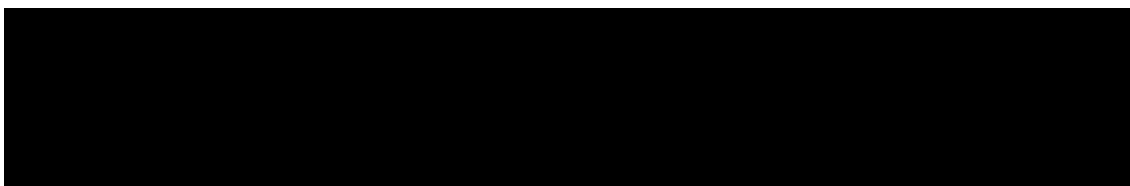


### Thông qua hệ 8 và hệ 16 để chuyển đổi hệ 2 sang hệ 10

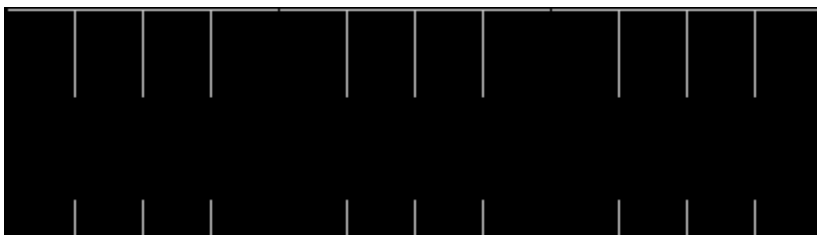
Chia số nhị phân làm thành từng bộ 3 số và 4 số liên tiếp theo thứ tự tương ứng với cách thông qua hệ 8 và hệ 16 và dùng phương pháp nhân với các thừa số bên trên tương ứng rồi cộng lại .

Ví dụ1: 

THÔNG QUA HỆ 8: Chia số nhị phân từng bộ 3 số:



THÔNG QUA HỆ 16: Chia số nhị phân thành bộ 4 số



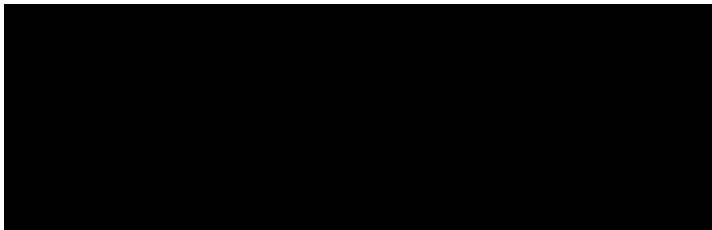
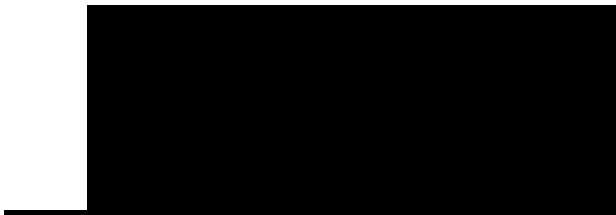


**Thông qua hệ 8 và hệ 16 để chuyển hệ 10 sang hệ 2**

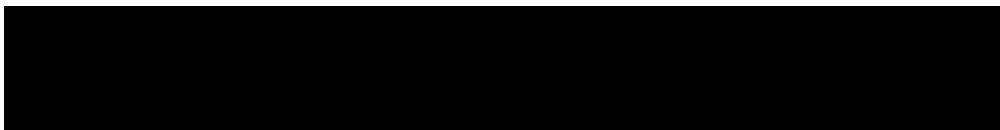
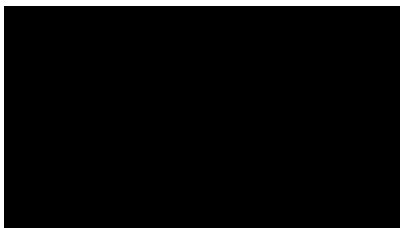
Cách làm tương tự như trên, nhưng thay phép nhân thành phép chia và lấy các số dư của phép chia ngược từ dưới lên trên để chuyển đổi.

Ví dụ2: 

THÔNG QUA HỆ 8: \_\_\_\_\_



THÔNG QUA HỆ 16: \_\_\_\_\_



Suy ra:



**Chuyển hệ 8 sang hệ 16 và ngược lại:**

Ta có thể dùng hệ 10 hoặc hệ 2 làm trung gian để chuyển đổi hệ 8 sang hệ 16 và ngược lại. Thông thường dùng hệ 2 để trung chuyển có thuận lợi hơn.

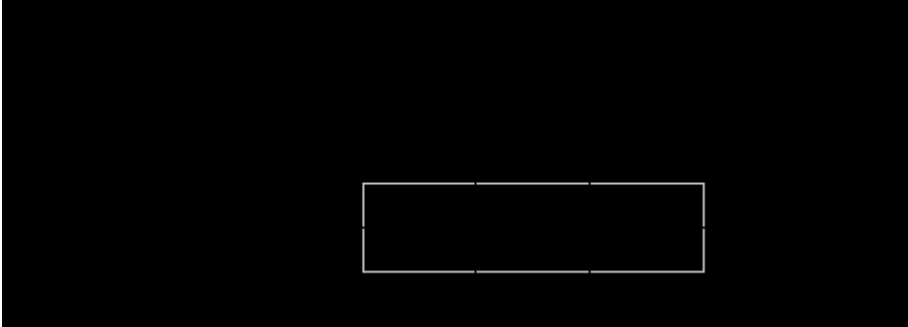
Ví dụ 3: 

Cách làm như sau:

**Bước 1: Chuyển hệ 8 thành hệ 2: biểu thị từng tri số trong hệ 8 thành từng nhóm 3 số và ghép các nhóm đó lại.**



**Bước 2: Chia dãy số hệ 2 vừa có được thành các bộ 4 số và chuyển các bộ đó sang hệ 16**



Việc chuyển từ hệ 16 sang hệ 8 ta cũng tiến hành 2 bước như vậy.

## CHƯƠNG 4

### HỆ ĐIỀU HÀNH MS-DOS

#### KHÁI NIỆM VỀ HỆ ĐIỀU HÀNH.

##### I. HỆ ĐIỀU HÀNH MS-DOS.

1. Các khái niệm cơ bản.
2. Tập lệnh nội trú, tập lệnh ngoại trú.

#### BÀI ĐỌC THÊM.

##### I. KHÁI NIỆM VỀ HỆ ĐIỀU HÀNH

Hệ điều hành (Operating System) là một tập hợp các chương trình tạo sự liên hệ giữa người sử dụng máy tính và máy tính thông qua các lệnh điều khiển. Không có hệ điều hành thì máy tính không thể chạy được. Chức năng chính của hệ điều hành là:

- Thực hiện các lệnh theo yêu cầu của người sử dụng máy,
- Quản lý, phân phối và thu hồi bộ nhớ ,
- Điều khiển các thiết bị ngoại vi như ổ đĩa, máy in, bàn phím, màn hình,...
- Quản lý tập tin,...

Hiện có nhiều hệ điều hành khác nhau như DOS, UNIX, OS2, WINDOWS,...

##### II. HỆ ĐIỀU HÀNH MS-DOS

Hệ điều hành MS-DOS (MicroSoft - Disk Operating System) là phần mềm khai thác đĩa từ (đĩa cứng hoặc đĩa mềm) rất thông dụng. MS-DOS là sản phẩm nổi tiếng của hãng Microsoft được cài đặt hầu hết trên các máy IBM PC và các máy tương thích. Phiên bản cuối cùng của DOS hiện nay là

version 7.0. Tuy nhiên, hệ điều hành MS-DOS version 6.0 - 6.22 vẫn còn được nhiều người tiếp tục sử dụng.

Một máy tính có thể có 1 hoặc 2 ổ đĩa mềm và 1 hoặc vài ổ đĩa cứng. Ổ mềm thường được đặt tên là ổ A và B (drive A, drive B), ổ cứng có tên là C, D, E,... Một đĩa cứng có thể được chia làm nhiều ổ đĩa logic nhỏ có các tên khác nhau.

Có 3 cách để khởi động máy:

- Bật công tắc khởi động khi máy chưa vào điện.
- Nhấn nút RESET khi máy đã hoạt động và cần khởi động lại.
- Nhấn tổ hợp 3 phím đồng thời là Ctrl-Alt-Del để khởi động lại máy.

Sau khi thực thi các lệnh khởi động, trên màn hình máy tính sẽ xuất hiện dấu đợi lệnh (prompt) có dạng:

C:\>\_                    hoặc                    A:\>\_

**C hoặc A là tên của ổ đĩa làm việc: C khi khởi động từ đĩa cứng và A là từ đĩa mềm.** Bộ ký tự :> là qui ước dấu đợi lệnh của DOS, qui ước này có thể thay đổi. Điểm nháy sáng \_ gọi là con trỏ (cursor) cho ta biết điểm làm việc hiện tại trên màn hình. Các ký tự gõ trên bàn phím sẽ hiện ra tại vị trí con trỏ.

**Một số qui ước gọi lệnh trong DOS**

- drive :            ổ đĩa
- path              đường dẫn bao gồm thư mục cha, các thư mục con kế tiếp
- filename        tên tập tin bao gồm phần tên và phần mở rộng
- directory        thư mục
- sub-dir          thư mục con (sub directory)
- <tên>            nội dung câu lệnh bắt buộc cần có
- [<tên>]           nội dung câu lệnh trong dấu [ ] có thể có hoặc không

■ Dấu Enter, ra lệnh thực hiện lệnh của DOS

Ghi chú: Ta có thể đánh tên lệnh và dấu /? để nhận được hướng dẫn (HELP) các chi tiết sau lệnh. Ví dụ C:\>DIR /?                    ■, DOS sẽ chỉ dẫn về lệnh DIR trên màn hình.

**1. Các khái niệm cơ bản**

**a. Tập tin (file)**

Tập tin là nơi lưu trữ thông tin bao gồm chương trình, dữ liệu, văn bản,... Mỗi tập tin có một tên riêng phân biệt. Tên tập tin thường có 2 phần: phần tên (name) và phần mở rộng (extension). Phần tên là bắt buộc phải có của một tập tin, còn phần mở rộng thì có thể có hoặc không.

- Phần tên là một dãy có từ 1 đến tối đa 8 ký tự có thể là: các ký tự chữ từ A đến Z, các chữ số từ 0 đến 9, các ký tự khác như #, \$, %, ~, ^, @, (, ), !, \_
- Phần mở rộng có từ 0 đến tối đa 3 ký tự trong số các ký tự nêu ở trên.
- Giữa phần tên và phần mở rộng có một dấu chấm (.) ngăn cách.



Tên tập tin không chấp nhận các trường hợp sau:

- Có khoảng trống trong tên file
- Trùng tên với các lệnh của DOS và lệnh điều khiển thiết bị: CON, PRN, ....
- Có chứa các ký tự như ., ?, \*, :, >, <, /, \, [, ], +, ;,

Phần mở rộng có thể được xem gần như họ trong tên người. Ta có thể căn cứ vào phần mở rộng để xác định kiểu của file:

- COM, EXE, BAT : Các file khả thi và lệnh bó chạy trực tiếp được trên MS-DOS
- TXT, DOC, ... : Các file văn bản
- PAS, BAS, ... : Các file chương trình PASCAL, BASIC, ...
- WK1, XLS, ... : Các file chương trình bảng tính LOTUS, EXCEL ...
- DBF, DAT, ... : Các file dữ liệu

Các ký tự đặc biệt trên file: DOS dùng các ký tự sao (\*) và chấm hỏi (?) để mô tả một tập hợp file. Ý nghĩa như sau:

- Dấu \* dùng để đại diện cho một chuỗi ký tự bất kỳ và thay cho phần còn lại của tên file hoặc phần mở rộng của file tại vị trí nó xuất hiện trở về sau.
- Dấu ? dùng để đại diện cho một ký tự bất kỳ tại vị trí nó xuất hiện.

Ví dụ 4.2: Trong đĩa của bạn có các tập tin:

BAOCAO1.TXT, BAOCAO2.TXT, VANBAN.TXT, VANCAO#.THO, SOLIEU.DAT

+ Ký hiệu BAOCAO?.\* đại diện cho các tập tin BAOCAO1.TXT, BAOCAO2.TXT

+ Ký hiệu \*.TXT đại diện cho BAOCAO1.TXT, BAOCAO2.TXT, VANBAN.TXT

+ Ký hiệu ???CAO?.\* đại diện cho BAOCAO1.TXT, BAOCAO2.TXT, VANCAO#.THO

+ Ký hiệu \*.\* hoặc duy nhất một dấu chấm . đại diện cho tất cả các tập tin trên đĩa

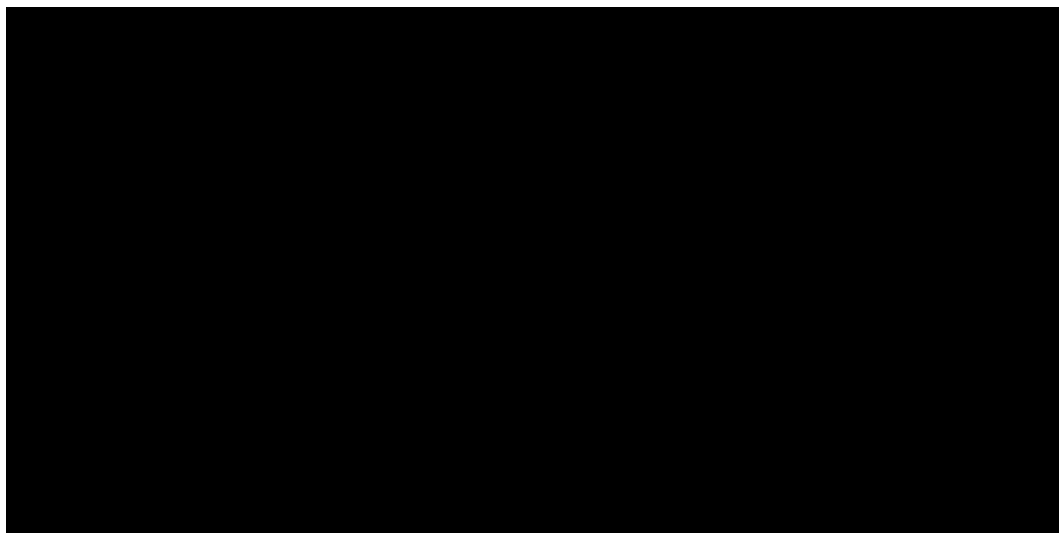
## b. Thư mục (directory)

Thư mục là nơi cất giữ các tập tin theo một chủ đề nào đó theo ý người sử dụng. Đây là biện pháp giúp ta quản lý được tập tin, dễ dàng tìm kiếm chúng khi cần truy xuất. Các tập tin có liên quan với nhau có thể được xếp trong cùng một thư mục.

Bản thân mỗi đĩa mang một thư mục chung gọi là thư mục gốc (root directory). Thư mục gốc không có tên riêng và được ký hiệu là \ (dấu xỏ phải: backslash). Dưới mỗi thư mục gốc có các file trực thuộc và các thư mục con (sub-directory). Trong các thư mục con cũng có các file trực thuộc và thư mục con của nó. Thư mục chứa thư mục con gọi là thư mục cha (parent directory).

Thư mục đang làm việc gọi là thư mục hiện hành (current directory).

Ta có thể diễn tả cấu trúc thư mục và file chứa trong nó qua hình ảnh cây thư mục (directory tree). Ví dụ trên đĩa C, ta có 2 thư mục con của thư mục gốc là VANBAN và PASCAL



### c. Đường dẫn (path)

Đường dẫn là lệnh chỉ dẫn lộ trình cho phép ta từ thư mục bất kỳ có thể đến trực tiếp thư mục cần truy xuất. Có 2 loại đường dẫn có thể sử dụng trong MS - DOS là: đường dẫn (path) và tên đường dẫn (pathname).

Đường dẫn một dãy các thư mục bắt đầu từ thư mục gốc đến các thư mục con và nối tiếp nhau bởi dấu \, thư mục đứng sau là con của thư mục đứng trước. Nói cách khác, đường dẫn dùng để chỉ định thư mục cần đến.

Cú pháp Đường dẫn: path [drive:][\directory][\sub-dir ...] ■

Ví dụ 4.3: C:\>path PASCAL\MAIN ■

Tên đường dẫn tương tự như đường dẫn nhưng kết thúc bằng tên file ở cuối cùng là. Đường dẫn kết thúc bằng tên file cần truy xuất.

Cú pháp Tên đường dẫn: path[drive:][\directory][\sub-dir ...]<\filename>

Ví dụ 4.4: C:\>path C:\PASCAL\MAIN\TURBO.EXE ■

## 2. Tập lệnh nội trú, tập lệnh ngoại trú

### a. Lệnh nội trú (internal command)

Lệnh nội trú là những lệnh nằm thường trực trong bộ nhớ máy khi đã được khởi động và sẵn sàng thực hiện lệnh khi ta gọi đến. Lệnh nội trú nằm trong phần khởi động của MS-DOS chứa trong các file COMMAND.COM, IO.SYS và MSDOS.SYS.

+ Tập tin COMMAND.COM là tập tin quan trọng nhất, có nhiệm vụ thông dịch lệnh và xử lý các lệnh nội trú. Khi khởi động máy, COMMAND.COM được nạp vào bộ nhớ RAM. Khi nhận được lệnh từ bàn phím, tập tin này sẽ nhận diện và điều khiển việc thi hành các lệnh từ người sử dụng.

+ Hai tập tin IO.SYS và MSDOS.SYS là hai tập tin ẩn, nó cũng được nạp vào RAM khi khởi động máy. Chúng có nhiệm vụ quản lý và điều khiển các thiết bị ngoại vi và các tập tin trên đĩa.

Tập lệnh nội trú gồm các lệnh chính thường dùng như:

- Các lệnh liên quan đến thư mục: DIR, CD, MD, RD, PATH, TREE, ...
- Các lệnh liên quan đến file: TYPE, DEL, COPY, REN, ...
- Các lệnh thời gian: TIME, DATE
- Các lệnh khác: PROMPT, CLS, VER, VOL,...

#### Các lệnh nội trú cơ bản:

\* Chuyển ổ đĩa:

Cú pháp : <Ký tự ổ đĩa :> ■ hoặc <drive :> ■

Với drive: là ổ đĩa cần chuyển đến.

Muốn chuyển từ ổ đĩa C sang ổ đĩa A ta gõ C:\> A: rồi nhấn nút Enter. Tương tự chuyển từ A: sang U: ta gõ lệnh U: ■

\* Liệt kê thư mục (DIR)

Hiển thị danh sách các tập tin và các thư mục con có trong thư mục.

Cú pháp: DIR [drive :] [path][/P][/W][/A : attribs][/O : sortorder] ■

Ghi chú:

/P : hiển thị từng trang màn hình (Page)

/W : hiển thị theo hàng ngang (Wide), lược bỏ bớt số liệu về kích thước byte, ngày, giờ.

/A : hiển thị thuộc tính (Attribut) của file

/Option: các ý định hiển thị trật tự sắp xếp:

N : theo alphabetic tên file;

E : theo alphabetic tên phần mở rộng

S : theo kích thước bytes (từ nhỏ đến lớn)

D : theo ngày tháng và giờ (từ trước đến nay)

G : theo nhóm thư mục trước

\* Tạo thư mục mới (Make Directory - MD)

Tạo một thư mục mới trong ổ đĩa hoặc thư mục hiện hành.

Cú pháp: MD [drive :][path]<tên thư mục mới> ■

\* Đổi thư mục (Change Directory - CD)

Cú pháp: CD [drive :] {path} ■

Ví dụ 4.5 C:\>CD PASCAL ■ sẽ có C:\PASCAL>\_

Ghi chú:

- Từ thư mục con, muốn trở về thư mục cha, ta gõ: CD.. ■

- Nếu muốn về thẳng thư mục gốc, ta gõ: CD\ ■

- Để hiển thị đường dẫn hiện hành, ta gõ: CD ■

\* Xem nội dung tập tin trên màn hình (TYPE)

Cú pháp: TYPE [drive:][path]<file name> ■

Ghi chú:

- Lệnh TYPE dùng hiển thị một tập tin văn bản chứa mã ASCII mới đọc được.

- Các file chứa mã nhị phân của chương trình như các file \*.EXE, \*.COM, \*.BIN,... thì khi gõ lệnh TYPE sẽ không đọc bình thường được.

\* Xóa thư mục (Remove Directory - RD)

Xoá bỏ một thư mục con rỗng (không chứa các tập tin và thư mục con).

Cú pháp: RD [drive :] <path> ■

\* Sao chép tập tin (COPY)

**Cú pháp 1: sao chép tập tin sang một vị trí khác.**

**COPY [drive1 :][path1]<filename1> [drive2:][path2][<filename2>] ■**

Ghi chú:



- Muốn copy một nhóm tập tin, ta có thể dùng các ký tự đại diện của tập tin là dấu \* hoặc ? trong <filename>
- Nếu không viết <filename 2> thì máy sẽ hiểu là sao chép mà không đổi tên tập tin. Khi đó ổ đĩa [drive1:] phải khác [drive2:] hoặc đường dẫn <path1> khác <path2>
- Khi không chỉ rõ Ổ đĩa và đường dẫn thì máy sẽ hiểu đang thực hiện trên ổ đĩa và thư mục hiện hành.

**Cú pháp 2: cho phép nối các tập tin có sẵn thành một tập tin mới chung.**

**COPY <file 1> + <file 2> [+ ... + <file n>] [<new\_file>] ■**

Ghi chú:

- Nếu không đặt tên tập tin mới thì tất cả các tập tin sẽ ghép chung vào <file 1> - Nếu <new\_file > đã có thì nội dung cũ sẽ được thay bằng nội dung mới.
- Tên <new\_file > không được trùng với tên các tập tin cần ghép.

**Cú pháp 3: sao chép tập tin ra máy in.**

**COPY <filename> PRN ■**

**Cú pháp 4: dùng lệnh COPY để tạo ra một tập tin văn bản đơn giản.**

**COPY CON <filename> ■**

Ghi chú:

- Sau khi gõ Enter, ta có thể đánh vào một vài đoạn văn trên bàn phím tùy ý. Muốn kết thúc, đánh tổ hợp phím Ctrl+Z hay F6 và Enter để lưu trữ.
- Khi gõ Enter để xuống dòng, ta không di chuyển con trỏ trở lên dòng trên được.
- Lỗi văn bản sai không thể sửa nội dung trực tiếp khi đã lưu, ngoại trừ phải dùng các trình soạn thảo văn bản khác.

\* Xoá tập tin (Delete - DEL)

Cú pháp: **DEL [drive:][path]<file name>[/P] ■**

Ghi chú:

- Có thể xoá một loạt <tên file> nếu dùng các ký tự \* và ?
- [/P] tạo nhắc để xác định từng tập tin muốn xoá
- Xoá tất cả các tập tin thì dùng lệnh DEL \*.\* khi đó máy sẽ hỏi lại :

All files in directory will be deleted ! (Tất cả các tập tin sẽ bị xoá !

Are you sure (y/n) ? \_                      Bạn có chắc không (y/n) ?)

Gõ y (yes) nếu muốn xóa tất cả và n (no) khi không muốn dùng lệnh xóa tất cả.

- Lệnh DEL không xóa các tập tin ẩn (hidden) và tập tin chỉ đọc (read only). Muốn xóa các tập tin này ta phải dùng lệnh ATTRIB ngoại trừ để thay đổi thuộc tính của nó.

- Nếu DEL một thư mục thì xóa hết các tập tin trong thư mục đó.

\* Đổi tên tập tin (Rename - REN)

Cú pháp: REN [drive:][path]<old\_file> <new\_file> ■

Ghi chú:

- <old\_file> : tên tập tin cũ cần đổi <new\_file> : tên tập tin mới

- Trường hợp tập tin mới đã có tên rồi hoặc không có tập tin cũ, máy sẽ báo:

Duplicate file name or file not found

\* Xóa màn hình (ClearScreen - CLS)

Cú pháp: CLS ■

\* Dấu đợi lệnh (PROMPT)

Cú pháp: PROMPT [\$text] ■

Ghi chú:

- Lệnh này định dạng lại dấu đợi lệnh của MS-DOS theo ý riêng của người sử dụng.

- \$text là chuỗi các ký tự liên tiếp nhau, trước mỗi ký tự có ký hiệu \$

\$P : ổ đĩa và thư mục hiện hành

\$G : dấu >

\$L : dấu <

\$Q : dấu =

\$T : giờ hiện hành

\$D : ngày hiện hành

\$N : ổ đĩa hiện hành

\$V : phiên bản của MS-DOS

\$\_ : xuống hàng

\* Xem và chỉnh thời gian (TIME)

Cú pháp:        TIME [hh:mm:ss] ■■■

Ghi chú:

- Sau khi Enter, trên màn hình sẽ xuất hiện giờ phút giây hiện tại của hệ thống.

Ví dụ 4.5:        TIME ■■■

Current Time is 9 : 30 : 15

Enter new time : \_

- Nếu không cần thay đổi thời gian, ta nhấn Enter để xác nhận.

- Muốn chỉnh thời gian, ta nhập giờ : phút : giây phù hợp tại vị trí con trỏ.

\* Xem và chỉnh ngày tháng (DATE)

Cú pháp:        DATE [mm-dd-yy] ■■■

Ghi chú:

- Sau khi gõ Enter, trên màn hình sẽ hiện ngày tháng năm hiện tại của hệ thống.

- Kiểu ngày tháng năm do người sử dụng thiết lập.

Ví dụ 4.6:        DATE ■■■

Current date is Sun 09-15-96

Enter new date (mm-dd-yy) : \_

- Nếu không muốn thay đổi ngày tháng của hệ thống, ta gõ Enter để xác nhận.

- Nếu muốn thay đổi ngày tháng, ta gõ theo thứ tự số : tháng-ngày-năm ■■■

Lưu ý: khi gõ sai, máy sẽ báo : Invalid date (ngày tháng không hợp lệ). Khi đó ngày tháng của hệ thống vẫn giữ nguyên và ta phải dùng lệnh DATE để nhập lại ngày tháng cho hệ thống.

Các máy tính Pentium hiện nay đều thể hiện phần năm là 4 con số, ví dụ :

Current date is Fri 08-04-2000

### **b. Lệnh ngoại trú ( external command)**

Lệnh ngoại trú cũng là những lệnh chứa các chức năng nào đó của điều hành nhưng ít được sử dụng hơn lệnh nội trú nên được để trên đĩa hay thư mục riêng để đỡ tốn bộ nhớ. Các lệnh ngoại trú phải được nạp từ đĩa vào trong bộ nhớ mới chạy được. Khi thực hiện xong câu lệnh, vùng bộ nhớ có chứa câu lệnh ngoại trú đó sẽ bị thu hồi. Các tập lệnh ngoại trú có phần mở rộng là EXE hay COM hoặc BAT. Khi gọi lệnh mà máy không tìm thấy trên màn hình hình sẽ xuất hiện câu báo lỗi Bad command or file name (Sai lệnh hoặc không có tên tập tin). Các lệnh ngoại trú như:

· Các lệnh liên quan đến ổ đĩa: FORMAT, LABEL, CHKDSK, DISKCOPY, ...

- Các lệnh liên quan đến tổ chức hệ thống: SYS, TREE, DELTREE, ...
- Các lệnh liên quan đến tập tin: UNDELETE, ATTRIB, FIND, ...
- Các lệnh khác: PRINT, GRAPHICS, SORT, ...

**Một số lệnh ngoại trú thường dùng:**

\* Tạo khuôn đĩa (FORMAT)

Lệnh FORMAT có tác dụng khởi tạo khuôn cho một đĩa mới hay tạo lại dạng khuôn cho đĩa cũ. Khi thực hiện lệnh FORMAT xong thì toàn bộ dữ liệu trong đĩa cũ hoàn toàn bị xoá sạch.

Cú pháp:       FORMAT [drive :][/S][/U][/Q][/V[:label]] ■

Ghi chú:

- Tham số /S dùng để tạo ra đĩa hệ thống. DOS sẽ tạo dạng và chép vào đĩa 3 file hệ thống là COMMAND.COM (file hiện), IO.SYS và MSDOS.SYS (file ẩn).
- Tham số /U thực hiện format không điều kiện (Unconditional)
- Tham số /Q thực hiện format nhanh (Quick)
- Tham số /V[:label ] để chỉ định tên nhãn (label) của đĩa

Ví dụ 4.7:       muốn format bình thường ổ đĩa A:

C:\>format A: ■

Insert new diskette for drive A :                   (Đưa đĩa mới vào ổ A:

and press ENTER when ready ...                   và ấn phím Enter khi sẵn sàng ...)

Sau đó là quá trình format. Ta có thể quan sát quá trình này qua số % đã thực hiện. Khi format xong, máy sẽ hiển thị dòng yêu cầu cho tên đĩa và bạn có thể cho tên đĩa với tối đa 11 ký tự và được quyền sử dụng ký tự trống, nếu không cần đặt tên, ta có thể nhấn Enter cho qua luôn.

Checking existing disk format                   (Kiểm tra sự tồn tại đĩa định dạng

Saving UNFORMAT information                   Lưu các thông tin UNFORMAT

Verifying 1.44 M                                   Thử 1.44 M

(90% percent completed)                       (90% phần trăm đã hoàn tất)

Format complete.                                 Hoàn tất định dạng

Volume label (11 characters, Enter for none) ?Tên nhãn (11 chữ, Enter nếu không)?

1,457,644 bytes total disk space               1,457,644 tổng số bytes trên đĩa

1,457,644 bytes available on disk             1,457,644 bytes dùng được trên đĩa

512 bytes in each allocation unit   512 bytes trên mỗi đơn vị allocation

2,847 allocation units available on disk 2,847 đơn vị allocation còn trống trên đĩa

Volume Serial Number is 2264 - 14E2 Số sơ-ri của đĩa là 2264 - 14E2

Format another (Y/N) ? Định dạng đĩa khác không (Y/N) ?

Gõ Y nếu muốn định dạng đĩa khác, N nếu muốn trở về.

\* Thay đổi thuộc tính của file (ATTRIB)

Mỗi tập tin đều có 4 thuộc tính (attribute) : R, S, H, A.

**Read Only** : Thuộc tính chỉ đọc, không thể thay đổi nội dung

**System** : Thuộc tính hệ thống, hệ điều hành DOS sẽ chú ý đặc biệt.

**Hidden** : Thuộc tính ẩn, khi dùng lệnh DIR sẽ không thấy.

**Archive** : Thuộc tính lưu trữ, thường dùng với các lệnh như BACKUP, XCOPY ... để tạo ra các file dự phòng.

Cú pháp:

**ATTRIB [-R|+R] [-S|+S] [-H|+H] [-A|+A] [drive:][path]<file name>** ■

Với dấu | là hoặc, hoặc chọn đặt (+) thuộc tính hoặc xóa (-) thuộc tính của tập tin.

\* Liệt kê cây thư mục TREE

Cú pháp: TREE [dirve :] [path] [/F] [/A] ■

Ghi chú:

/F : Thể hiện tên các tập tin trong từng thư mục

/A: Dùng ASCII thay cho các ký tự mở rộng

\* Xóa cây thư mục DELTREE

Lệnh này xóa thư mục được chỉ định và tất cả các thư mục con, tập tin của nó.

Cú pháp: DELTREE [/y] [dirve :] [path] <directory name> ■

Ghi chú:

directory name là tên thư mục đại diện cho một cây (nhánh) thư mục cần xóa.

Nếu có tùy chọn /y có nghĩa là người sử dụng đã xác nhận việc xóa cây thư mục này là chắc chắn.

\* Phục hồi file đã bị xóa (UNDELETE)

Ta có thể khôi phục lại các file đã lỡ bị xóa do dùng lệnh DEL, miễn là vùng chứa file này trong đĩa chưa bị các file khác chép chồng lên và nhất thiết ta phải nhớ ký tự đầu tiên của tên file.

Cú pháp: UNDELETE [drive:][path]<file name> ■

\* Di chuyển file MOVE

Để dời file từ nơi này sang nơi khác (thay vì dùng COPY và DEL)

Cú pháp 1: Di chuyển tập tin

**MOVE [drive:][path]<file name> <destination>** ■

Với destination là nơi di chuyển các files đến.

Cú pháp 2: Đổi tên thư mục

**MOVE [drive:][path]<old-directory> <new-directory>** ■

\* In văn bản ra máy in (PRINT)

Có thể in trực tiếp các file văn bản chứa mã ASCII ra máy in.

Cú pháp: PRINT </D: device>[drive:][path]<filename> ■

Nếu muốn in trực tiếp các thông tin hiển thị trên màn hình ta có thể dùng tổ hợp phím Shift + PrintScreen sau khi bật máy in lên.

Tùy chọn Device dùng để khai báo cổng máy in COM1, COM2, ... hoặc LPT1, LPT2, ...

## BÀI ĐỌC THÊM

### TẬP LỆNH XỬ LÝ THEO LÔ (BATCH FILE) VÀ TẬP LỆNH CẤU HÌNH MÁY (CONFIG.SYS)

--- oOo ---

#### I. TẬP LỆNH XỬ LÝ THEO LÔ

Để tránh việc phải gõ đi gõ lại một chuỗi lệnh mất thì giờ, MS-DOS cho phép ta đặt các chuỗi lệnh trong một tập tin, gọi là tập lệnh xử lý theo lô (Batch File) còn gọi là file BAT vì đều có phần mở rộng của file là BAT. Khi sử dụng ta chỉ cần gọi tên file BAT và nhấn Enter thì một loạt các chuỗi lệnh trong nó sẽ tuần tự thi hành. Trong file BAT, ta có thể dùng các lệnh nội trú và ngoại trú của DOS, các câu lệnh, tham số của BAT và các chương trình khác.

Ta có thể sử dụng các lệnh hoặc chương trình tạo văn bản của DOS như COPY CON, EDIT.COM, SIDEKICK, WINWORD,... để tạo ra các file BAT.\_

Ví dụ: Dùng lệnh của MS-DOS

C:\>COPY CON NHAPDATA.BAT ■

hay dùng chương trình EDIT :

C:\>EDIT NHAPDATA.BAT ■

Tập tin AUTOEXEC.BAT trong đĩa hệ thống là một file BAT đặc biệt. Khi khởi động máy, tuần tự các lệnh chứa trong nó sẽ được thi hành. Tập tin AUTOEXEC.BAT thường có các lệnh báo đường dẫn, thiết lập cấu hình, tạo dấu nhắc, báo ngày giờ và một số thông báo nhắc nhở người sử dụng.

Ta có thể tạo ra một tập tin AUTOEXEC.BAT đơn giản sau:

```
A:\>COPY CON AUTOEXEC.BAT
REM TAP TIN KHOI DONG MAY
ECHO OFF
TIME
DATE
PATH C:\; C:\DOS; C:\NC; C:\PASCAL; C:\WINDOW;
PROMPT PG
ECHO == PHONG MAY TINH CHAO CAC BAN ==
Gõ phím chức năng F6 (hoặc Ctrl + Z)
```

1 file(s) copied

**Lập trình cho file BAT :**

Ta có thể lập trình cho file BAT bằng cách sử dụng các lệnh FOR, GOTO, IF, ... với các tham số hình thức %0, %1, %2, ... (cho phép sử dụng 10 tham số hình thức từ %0 đến %9). Nếu ta dùng ký hiệu % như một thành phần trong file BAT thì phải ghi dấu % hai lần. Tham số %0 tương ứng với tên file BAT đang được thực hiện và %1, %2,... tương ứng với các tham số được đưa vào.

Ví dụ: Tạo một file BAT đơn giản như sau:

```
A:\>COPY CON VANBAN.BAT
ECHO %0
MD %1
CD %1
COPY C:\%2.TXT
DIR A:
Gõ phím F6 (hoặc Ctrl + Z)
```

1 file(s) copied

Khi dùng lệnh:

A:>VANBAN TEXT BAOCAO ■

Sau khi Enter, tên VANBAN sẽ thay vào %0, tên TEXT sẽ thay vào vị trí %1 và BAOCAO sẽ vào vị trí %2. Do vậy, máy sẽ hiểu cần thi hành tuần tự các công việc sau:

- Hiện thị câu: VANBAN
- Mở thư mục: MD TEXT
- Vào thư mục: CD TEXT
- Chép file BAOCAO.TXT từ ổ đĩa C
- Liệt kê tất cả các file trong đĩa A

### Các câu lệnh chính dùng trong file BAT:

Trong file BAT cho phép sử dụng dấu @ (a sign), nếu đặt @ trước một lệnh thì lệnh đó không hiển thị ra màn hình mà chỉ thi hành lệnh.

#### ☒ **ECHO**

Cú pháp: ECHO [on/off]<dòng thông báo>

Ghi chú:

- Lệnh Echo cho phép đưa ra (on) hay không đưa ra (off) màn hình câu lệnh. <dòng thông báo> xuất hiện không cần để ý trạng thái của Echo là on/off

#### ☒ **CALL**

Cú pháp: CALL [drive:][path]<batch file>[argument]

Ghi chú:

- Lệnh Call cho phép ta gọi một file BAT khác chạy trong file BAT hiện tại.
- <batch file> là tên tập tin BAT khác mà ta muốn gọi.
- [argument] là lệnh sẽ thi hành tiếp theo sau khi file BAT được gọi.

#### ☒ **PAUSE**

Cú pháp: PAUSE [comment]

Ghi chú:

- Lệnh Pause tạm thời dừng thi hành file BAT
- [comment] dòng chú thích hiển thị ra màn hình, không dài quá 123 ký tự.
- Các lệnh sau lệnh Pause sẽ tiếp tục khi ta gõ một nút bất kỳ trên bàn phím.



✉ **REM**

Cú pháp: REM [comment]

Ghi chú:

- Lệnh REM(remark) ghi các nhắc nhở [comment ] dài không quá 123 ký tự.

- Các comment này sẽ xuất hiện trên màn hình nếu trước đó ta có ECHO ON và ngược lại, chúng sẽ không xuất hiện khi có ECHO OFF.

✉ **IF**

Cú pháp 1: IF [NOT] EXIST [drive:][path]<file name><Lệnh>

Ghi chú:

- Lệnh IF là lệnh điều kiện với ý nghĩa: NẾU ... thì thi hành <Lệnh>

- IF [NOT] là phủ định (not tùy chọn): NẾU KHÔNG ... thì thi hành <Lệnh>

Ví dụ: Trong 1 file BAT ta có dòng:

```
COPY C:\DOS*.COM A:

IF EXIST A: EDIT.COM EXIT

DIR A:
```

Lệnh trên có nghĩa là copy ở C:\DOS tất cả các file .COM vào đĩa A.

Nếu đĩa A đã có file EDIT.COM thì thôi không thực hiện nữa (để khỏi mất thì giờ) và thoát ra (EXIT) thực hiện dòng tiếp là liệt kê các file ở đĩa A.

Cú pháp 2: IF [NOT] String1 == String2 <Lệnh>

Ghi chú: Nếu chuỗi ký tự String1 giống chuỗi ký tự String2 thì <Lệnh> được thực hiện.

Ví dụ: IF %1 == IN ECHO MO MAY IN RA !

Nghĩa là nếu chương trình chứa tham số %1 đúng là IN thì hiển thị (ECHO) dòng chữ MO MAY IN RA !. Ngược lại, nếu tham số %1 không là IN thì không thi hành lệnh ECHO theo sau.

Cú pháp 3: IF [NOT] ERRORLEVEL # <Lệnh>

Ghi chú: Lệnh điều kiện này để báo mức độ lỗi. Tùy theo mức độ lỗi đã định mà thực hiện <Lệnh> tương ứng.

Ví dụ: COPY C:\PRG\\*.PAS A:

```
IF ERRORLEVEL 1 ECHO KHONG COPY DUOC
```

Nghĩa là copy tất cả các file .PAS ở C:\PRG qua A. Vì một lý do nào đó ta không copy được (không có đĩa A, hay đĩa A bị hư hay đĩa A đã đầy ...) thì chương trình sẽ báo dòng chữ KHONG COPY DUOC.

✉ **FOR**

Cú pháp: FOR %%<Biến> IN (Tập hợp) DO <Lệnh> [Tham số Lệnh}

Ghi chú:

- Lệnh FOR thực hiện vòng lặp với từng file trong tập hợp file.

- Nghĩa là, <Biến> (variable) sẽ lần lượt nhận các giá trị trong <Tập hợp> (Set) để thực hiện <Lệnh> (comamnd) theo [Tham số Lệnh] (Command parameters).

Ví dụ: FOR %%T IN (C:\VANBAN\\*.TXT) DO COPY %%T A:\

Sao chép lần lượt tất cả các file .TXT trong C:\VANBAN vào A:\

✉ **GOTO**

CÚ PHÁP: GOTO : <NHÃN>

Ghi chú:

- Lệnh GOTO là lệnh di chuyển tới dòng lệnh đứng sau <Nhãn> (Label).

- Trước <Nhãn> phải có dấu hai chấm (:), chiều dài có giá trị của <Nhãn> chỉ có 8 ký tự mặc dầu ta có thể viết dài hơn tùy ý.

- Nếu <Nhãn> không xác định được trong file BAT thì khi thi hành lệnh GOTO máy sẽ báo:

Label not found (Nhãn không tìm thấy)

Ví dụ: FORMAT A:/S

IF ERRORLEVEL 0 GOTO ECHO KHONG FORMAT A: DUOC

✉ **SHIFT**

Cú pháp: SHIFT

Ghi chú:

- Lệnh SHIFT dùng để thay đổi vị trí tham số có thể thay thế được trong .BAT

- File BAT có tối đa 10 biến có thể thay thế từ %0 đến %9, khi dùng SHIFT thì sẽ có một sự thay đổi biến như sau: %0 mất đi, %1 chép đè lên %0, và %2 chép đè lên %1, và %3 chép đè lên %2, ...

- Lệnh SHIFT có thể dùng quá 10 tham số.

Ví dụ: Ta có file mang tên SAPCHU.BAT trong ổ A với nội dung sau:

ECHO OFF

ECHO %0 %1 %2 %3

SHIFT

ECHO %0 %1 %2 %3

SHIFT

ECHO %0 %1 %2 %3

SHIFT

Khi cho chạy thử với dòng lệnh:

A:\>SAPCHU Toi thích an kem lam ████

Kết quả hiện ra màn hình là:

A:\> ECHO OFF

SAPCHU Toi thích an

Toi thích an kem

Thich an kem lam

## II. TẬP LỆNH CẤU HÌNH MÁY CONFIG.SYS

Tập tin CONFIG.SYS là file tạo cấu hình hệ thống, nó cho phép ta thay đổi các cấu hình chuẩn (default) của MS-DOS. Hầu hết các máy vi tính hiện nay đều có 2 tập tin đồng thời là AUTOEXEC.BAT và CONFIG.SYS. Hai file này chứa các thông tin liên quan đến cấu hình hệ thống và đặt tại thư mục gốc của đĩa khởi động máy.

Tập tin CONFIG.SYS có thể tạo bằng lệnh COPY CON CONFIG.SYS, hoặc các trình lập văn bản như EDIT.COM, WINWORD, .... Sau khi khởi động máy, DOS sẽ tìm CONFIG.SYS trong thư mục gốc đĩa khởi động hệ thống của nó. Nếu không tìm thấy CONFIG.SYS, DOS sẽ lập cấu hình của máy theo trị mặc nhiên.

### Một số lệnh trong CONFIG.SYS

#### ☒ **BREAK**

Cú pháp: BREAK = ON/OFF

Ghi chú:

Lệnh BREAK dùng để định chế độ kiểm tra bàn phím, BREAK = ON cho phép ta dùng tổ hợp phím Ctrl +C (hay Ctrl+Break) để ngưng việc thi hành lệnh, muốn tắt chế độ BREAK ta sử dụng lệnh BREAK = OFF.

#### ☒ **BUFFERS**

Cú pháp: BUFFERS = <n>

Ghi chú:

- Lệnh BUFFERS là lệnh cấp phát bộ nhớ đệm, nghĩa là định số sectors cho vùng nhớ đệm của hệ thống. Vùng nhớ đệm dùng để lưu trữ nội dung của file mà chương trình hiện hành đang dùng.

- <n> là giá trị vùng nhớ đệm có thể có giá trị từ 2 đến 255. Khi cấu trúc thư mục trở nên phức tạp, thì nên tăng cường số buffers để tạo hiệu quả thêm cho DOS. Tuy nhiên, quá nhiều buffers thì vùng nhớ trống còn lại của chương trình và dữ liệu sẽ ít đi.

- Thông thường, giá trị vùng nhớ đệm được khai báo là buffers = 30.

✉ **COUNTRY**

Cú pháp: COUNTRY = xxx[, [yyy], [drive:]<filename>]

Ghi chú:

- Lệnh COUNTRY chỉ định các thông tin cho MD-DOS về ngày, giờ, tiền tệ,... phù hợp với mỗi quốc gia. Lệnh này cũng chỉ ra bộ mã ký tự của nước đó.

- xxx là mã điện thoại quốc tế của một nước : 001 của Mỹ, 084 của Việt nam, ...

- yyy là trang mã (code page) của một nước.

- filename là tập tin thông tin quốc gia, nếu không có DOS sẽ sử dụng COUNTRY.SYS của hệ thống.

✉ **FILES**

Cú pháp: FILES = <x>

Ghi chú:

- Lệnh Files dùng để xác định số tập tin mà MS-DOS có thể mở ra đồng thời để xử lý. Trường hợp, số files mở ra nhiều hơn số đã định thì gặp dòng thông báo:

Too many files open (Mở quá nhiều tập tin)

- Số file <x> có thể từ 8 đến 255. Giá trị mặc nhiên là 8, thường thì Files = 30

✉ **DEVICE**

Cú pháp: DEVICE = [drive:][path]<file name>[argument]

Ghi chú:

- Lệnh DEVICE dùng để cài module điều khiển các thiết bị cho MS-DOS

- <file name> là tên tập tin dùng để điều khiển thiết bị, thí dụ như:

ANSI.SYS            DISPLAY.SYS            DRIVER.SYS  
CONFIG.SYS        PRINTER.SYS            RAMDRIVE.SYS (VDISK.SYS) ...

- [argument] chỉ ra các thông tin của dòng lệnh mà chương trình điều khiển thiết bị yêu cầu.

Ví dụ: Lệnh cho phép xác định một phần bộ nhớ của máy tính như là một ổ cứng, vùng bộ nhớ này gọi là RAM disk và làm việc như một ổ đĩa ảo (virtual disk) :

DEVICE = C:\DOS\RAMDRIVE.SYS 100/E

câu lệnh trên sẽ tạo ra một đĩa ảo có kích thước 100 KB nằm trong vùng nhớ mở rộng (/E : Expanded Memory). Tên đĩa ảo sẽ là tên alphabetic tiếp theo tên của đĩa cứng. Ví dụ đĩa cứng chia thành 2 ổ có tên là C: và D: thì đĩa ảo sẽ có tên là E:

☒ **SHELL**

Cú pháp: SHELL =[drive:][path]<filename>

Ghi chú:

- Lệnh SHELL cho phép sử dụng bộ xử lý lệnh cấp cao (top-level command processor) thay thế lệnh COMMAND.COM.

- <filename> là tên của chương trình xử lý lệnh.

## CHƯƠNG 5

### CÁC PHẦN MỀM HỖ TRỢ

- I. [PHẦN MỀM TIỆN ÍCH NORTON COMMANDER.](#)
  1. [Giới thiệu.](#)
  2. [Các thao tác căn bản trong NC.](#)
- II. [VIRUS TIN HỌC VÀ CÁCH PHÒNG CHỐNG.](#)
  1. [Virus tin học là gì?.](#)
  2. [Cách phòng chống Virus.](#)
  3. [Các chương trình kiểm tra và diệt Virus thông dụng.](#)

#### I. PHẦN MỀM TIỆN ÍCH NORTON COMMANDER

##### 1. Giới thiệu

Norton Commander (viết tắt là NC) là một trong những chương trình tiện ích chạy trong môi trường DOS. NC được sử dụng nhiều ở Việt nam và nhiều quốc gia khác. Ưu điểm của NC là đơn giản, dễ sử dụng và không chiếm nhiều không gian, bộ nhớ mà vẫn đủ các chức năng cơ bản và nâng cao. Sử dụng được NC, ta dễ dàng khai thác các trình ứng dụng tương tự như Norton Utilities, PC Tools, Xtree Gold, File Manager trong Windows, v.v...

Phiên bản Norton Commmander được dùng phổ biến hiện nay tại Việt nam là NC version 4.0 và version 5.0. Chúng ta có thể cài đặt toàn bộ hay một phần chương trình. Hai file quan trọng nhất của NC là NC.EXE và MCMAN.EXE. Thường NC được cài trong phần Path của Autoexec.bat.

Để đi vào NC, từ dấu nhắc của DOS, ta gõ NC rồi nhấn Enter. Khi đó, trên màn hình sẽ xuất hiện hai cửa sổ gọi là Panels, như sau:



Hình 5.1: Panel làm việc của Norton Commander

Panel có viền sáng được gọi là cửa sổ làm việc hiện thời. Muốn chuyển cửa sổ làm việc ta dùng phím Tab. Trên các panels, có thể đọc được các thông tin:

- Nội dung của các thư mục và files hiện thời
- Cấu trúc thư mục trên đĩa
- Các thông tin về chương trình, bộ nhớ, file trống trên đĩa hiện hành
- Dòng cuối cùng là các phím nóng (HotKey) chỉ dẫn các thao tác nhanh.

- F1 Help : Hướng dẫn
- F2 Menu : Gọi Thực đơn của người sử dụng
- F3 View : Xem nội dung file có vệt sáng hiện hành
- F4 Edit : Soạn thảo, sửa chữa nội dung file
- F5 Copy : Sao chép files sang nơi khác
- F6 RenMov : Di chuyển files hoặc đổi tên
- F7 Mkdir : Tạo thư mục con
- F8 Delete : Xóa files hoặc thư mục
- F9 PullDn : Mở menu làm việc của NC
- F10 Quit : Thoát khỏi chương trình

Tổ hợp phím:

- Ctrl + F1 : dùng để ẩn/ hiện Panel trái
- Ctrl + F2 : dùng để ẩn/ hiện Panel phải
- phím Tab : chuyển đổi vệt sáng giữa các cửa sổ

## 2. Các thao tác căn bản trong NC



\* **Xem cây thư mục**

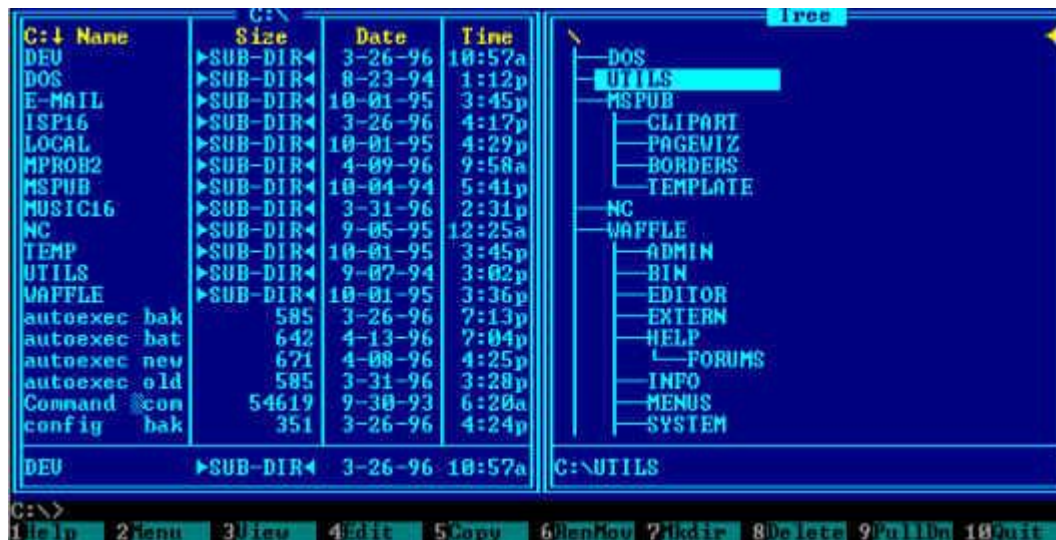
Nhấn F9 - ta có dòng Menu ở hàng trên, vào Right (Left) nếu muốn cây thư mục xuất hiện ở Panel Phải (Trái tương ứng), gõ Enter. Một bảng như sau xuất hiện:

Trong bảng này có nhiều option cho ta lựa chọn, muốn vào mục nào ta kéo vệt sáng (highlight) xuống dòng đó (hoặc nhấn phím chữ có đánh dấu màu) rồi nhấn Enter.

Ví dụ, muốn xem cây thư mục (Tree) thì ta kéo vệt sáng xuống vị trí Tree (hoặc nhấn phím T) rồi nhấn Enter.

Hình 5.2: Option của Thực đơn Right

Trên màn hình sẽ cho ta hình ảnh sau:



Hình 5.3: Hình ảnh cây thư mục ở Panel phải

\* **Chọn thư mục làm việc hoặc file**

Thư mục (Directory) và file là 2 đối tượng chính của NC. Thư mục thể hiện bằng chữ in hoa, trong khi file thể hiện bằng chữ thường. Ta có thể dùng các phím di chuyển trên bàn phím như **PageUp**, **PageDown** để đi đến các thư mục và file để chọn. Vệt sáng nằm ở thư mục hay file nào là NC đang làm việc tại thư mục hay file đó. Khi đưa vệt sáng đến các files có phần mở rộng là .BAT, .COM và .EXE và nhấn Enter thì NC sẽ thực hiện ngay lệnh đó trong môi trường của nó. Khi hoàn tất việc thực thi các file đó, chương trình quay trở về NC.

\* **Chọn một nhóm files**

Muốn chọn một nhóm files hoặc từng thư mục, ta di chuyển vệt sáng đến các files hoặc thư mục đó và gõ vào nút <Insert> trên bàn phím để chọn, khi đó tên file đã chọn sẽ đổi sang màu sáng trên màn hình panel. Trường hợp muốn hủy việc chọn lựa, ta cũng di chuyển vệt sáng đến chỗ file cần hủy chọn và nhấn <Insert> một lần nữa, khi đó màu sáng của file biến mất.

\* **Chọn một nhóm hoặc tất cả các files**

Muốn chọn tất cả các files trong một panel hiện hành hoặc một nhóm các files dựa trên ký hiệu đại diện, ta gõ nút <+> ở vùng phím số, khi đó trên màn hình sẽ xuất hiện hộp thoại:



Khi gõ Enter, tất cả các file trong panel sẽ được chọn và đồng loạt đổi sang màu sáng. Nếu ta chỉ muốn chọn một nhóm tập tin có cùng một kiểu đặt tên nào đó thì có thể sử dụng các ký tự đại diện như ? hay \* và gõ Enter khi đã định xong.

**\* Sao chép, di chuyển và xoá files**

Khi các nhóm file đã được chọn thì ta có thể sao chép, di chuyển hoặc xóa bỏ tùy ý. Ở chế độ mặc định, files hoặc thư mục sẽ theo câu lệnh mà sao chép hoặc di chuyển sang thư mục đối diện. Có thể dùng các thao tác sau :

Nhấn F5 (Copy) : sao chép files qua thư mục khác

Nhấn F6 (RenMov) : đổi tên hay di chuyển files

Nhấn F8 (Delete) : xóa files hay thư mục

Để chọn thư mục đích, ta dùng phím Alt-F10 và các phím mũi tên để chỉ định đường dẫn hay gõ trực tiếp tên đường dẫn.

Để bảo vệ files khi bị xóa hoặc chép đè lên, NC luôn đưa ra các thông báo cần thiết để người sử dụng khẳng định lại ý đồ của mình.

**\* Tạo thư mục**

Dùng phím F7 để tạo thư mục (Make Directory), sau đó gõ tên thư mục và nhấn Enter. Trường hợp, gõ tên thư mục mà không gõ đường dẫn thì NC mặc nhiên tạo thư mục con trong thư mục của panel hiện hành.

**\* Chuyển thư mục**

Cách đơn giản nhất là đưa vệt sáng đến thư mục cần chuyển và ấn Enter.

Tổ hợp phím <Alt + F10> sẽ hiện thị cây thư mục và ta dùng các phím mũi tên để di chuyển đến thư mục cần chuyển và gõ Enter.

**\* Chuyển ổ đĩa**

Dùng tổ hợp phím <Alt + F1> và các phím mũi tên để chọn ổ đĩa thích hợp và gõ Enter, khi đó nội dung ổ đĩa vừa chọn sẽ xuất hiện ở panel bên trái. Khi dùng tổ hợp phím <Alt + F2> thì nội dung ổ đĩa sẽ ở panel phải màn hình.

**\* Tìm kiếm file**

Để tìm kiếm nhanh một file nào đó trong thư mục hiện hành, ta nhấn giữ phím Alt và đồng thời gõ các ký tự đầu tiên của tên file. Nếu cho các ký tự đầu của tên file càng nhiều thì mức độ tìm kiếm càng nhanh.

Trường hợp không rõ thư mục chứa file hoặc tên file nằm ở các thư mục khác nhau, ta có thể sử dụng tổ hợp phím <Alt + F7>, sau đó gõ tên file cần tìm và nhấn Enter.



Khi gõ tên file ta có thể sử dụng các ký tự ? hoặc \* như trên DOS. NC sẽ tìm kiếm cho đến khi ta chọn Stop và Enter để ngưng. Các phím mũi tên được dùng để chọn một file nào đó trong nhóm vừa tìm thấy. Nếu chọn tiếp ChDir và Enter, chương trình sẽ đưa ta vào làm việc với thư mục chứa file vừa tìm được. Chức năng New search của NC cho phép bắt đầu lại việc tìm kiếm. Chức năng Change drive dùng để chuyển ổ đĩa cần tìm kiếm.

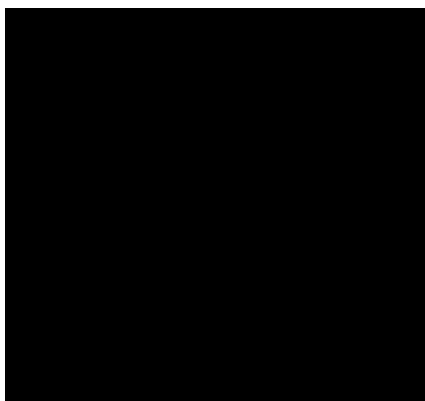
\* **Cài đặt thuộc tính cho files**

Trong MS-DOS, các files có thể có các thuộc tính (attributes) sau :

- Archive : thuộc tính lưu trữ
- Read Only : thuộc tính chỉ đọc
- Hidden: thuộc tính ẩn
- System : thuộc tính hệ thống

NC có thể giúp ta thay đổi thuộc tính cho file hoặc một nhóm file.

- Đánh dấu các files cần đặt/hủy thuộc tính
- Nhấn lần lượt phím F9 và các thực đơn con /File/File attributes. Hộp thoại như sau sẽ xuất hiện:



- Dùng các phím mũi tên và phím chọn SPACE để đánh dấu. Cột set là thiết lập, cột clear là hủy bỏ thuộc tính.
- Chọn chức năng set ở hàng dưới cùng để thực hiện lệnh.

## II. VIRUS TIN HỌC VÀ CÁCH PHÒNG CHỐNG

### 1. Virus tin học là gì ?

Virus tin học hay còn gọi là virus máy tính (computer virus) là một đoạn chương trình ẩn, có kích thước nhỏ, có khả năng tự sao chép, sinh sôi nảy nở như một virus sinh học, lây lan nhanh và có thể gây nguy cơ phá hoại các phần mềm hệ của hệ điều hành, xóa một phần hay toàn bộ các tập tin dữ liệu, tạo ra các tập tin vô nghĩa chiếm lấy không gian bộ nhớ của máy, hoặc chỉ đơn thuần trên chọc hay đe dọa suông người sử dụng, ...

Thoạt đầu, virus tin học là vũ khí của những người viết phần mềm muốn bảo vệ sản phẩm của họ chống việc sao chép bất hợp pháp. Sau một số người tạo ra các virus khác nhằm mục đích phá hoại hay đơn thuần chỉ để tự khẳng định mình và cấy vào trong các máy khác gây lây nhiễm trên một diện

rộng. Năm 1987, virus Lehigh lần đầu tiên được phát hiện trên máy IBM PC. Cùng lúc đó, virus Brain từ Pakistan xâm nhập vào các trường Đại học của Mỹ. Một virus khác từ Đại học Hebrew, Israel cũng xuất hiện. Từ đó đến nay, hàng ngàn chủng loại virus được thông báo phát hiện gây phá hoại, tê liệt nhiều hệ thống máy trên toàn thế giới gây thiệt hại nhiều triệu dollars.

Virus được chia thành 2 nhóm chính:

+ B-virus (Boot-virus) : là các virus chuyên tấn công lên các Boot sector hay Master boot, điển hình như : Brain, Nops, Clock, Stone, Sài Gòn 01/91, ...

+ F-virus (File-virus) : là các virus chỉ tấn công lên các file .COM và .EXE, điển hình như : Friday 13th, Yankee Doodle, DataLock, Keypress, VN01, ... Hay một số chuyên phá hoại các file có phần mở rộng .DB\* của các chương trình dBase, FoxBase hay chuyên phá hoại các file văn bản có phần mở rộng là .DOC như virus Concept, Macro, ....

Ngoài ra, một số virus vừa mang cả 2 tính phá hoại Boot và File như virus Dark Avenger .... Người ta còn lo ngại một số người đang cố tình tạo ra các virus còn có thể xâm nhập vào các hệ thống mạng máy tính qua đường truyền thư điện tử (E-mail) hoặc mạng Internet, mạng của các ngân hàng lớn gây đảo lộn các số liệu tài khoản, hoặc xâm nhập vào các hệ thống máy tính quân sự tạo nên những báo động giả rất nguy hiểm ...

Hiện nay, một số nước đã ra những điều luật xem những kẻ cố tình tạo ra các virus tin học phá hoại, lấy cắp mật mã, ... (gọi chung là những kẻ tin tặc - hackers) như là những tội phạm hình sự.

## 2. Cách phòng chống virus

- Không chép các chương trình mà bạn không rõ nguồn gốc và đặc biệt chưa qua thủ tục kiểm tra virus.
- Các file dữ liệu quan trọng cần phải được COPY lưu trữ dự phòng và có thể dùng các chương trình PKZIP.EXE hay ARJ.EXE .... để nén lại, tiết kiệm đĩa.
- Có thể sử dụng nhiều phần mềm BACKUP để lưu.
- Dùng lệnh BACKUP và RESTORE của DOS như sau:

Mẫu lệnh:      BACKUP source destination - drive : [Option]

Các option là:

- /S      :      lưu trữ cả thư mục con
- /M      :      chỉ lưu các file bị đổi so với lần lưu cuối cùng
- /A      :      lưu thêm các file vào đĩa lưu trữ
- /F:[ size] :      chỉ định kích thước đĩa cần định dạng
- /D: date    :      Chỉ lưu trữ các file bị thay đổi kể từ ngày ...
- /T : time    :      Chỉ lưu trữ các file bị thay đổi kể từ thời điểm ...
- /L:[drive:][path]logfile ] : Tạo nên một file nhật ký vào đầu vào để ghi các thao tác lưu trữ

Ví dụ muốn lưu trữ thư mục DOS và các thư mục con của nó vào ổ B, ta gõ:

```
C:\>BACKUP C:\DOS B:/S
```

Máy sẽ thông báo dòng tin:

Insert backup diskette 01 in drive B :

WARNING ! Files in the target drive

B:\ root directory will be erased

Press any key to continue ..

Sau khi lưu trữ vào B, ta có thể xem trên đĩa có nội dung:

```
C:\>dir B:
```

```
Volume in drive B is BACKUP 001
```

```
Volume Serial Number is 001F- 012B
```

```
Directory of B:\
```

```
BACKUP 001 1,432,246 04-20-96 6:20p
```

```
1 file(s) 1,432,246 bytes
```

```
7,754 bytes free
```

Chương trình RESTORE là chương trình khôi phục các file trong BACKUP

Mẫu lệnh như sau:

```
RESTORE drive1 : drive2 : [path[filename]] [Option]
```

trong đó: drive1 : chỉ ổ đĩa chứa file backup

drive2 :[path[filename]] : chỉ rõ các file cần phục hồi

các options là:

/S : Phục hồi tất cả file trong các thư mục con trong đường dẫn

/P : Báo cho ta biết trước khi khôi phục các read-only-files hoặc các file đã thay đổi từ lần backup cuối cùng (nếu các thuộc tính phù hợp được thiết lập).

/M : Chỉ khôi phục các file có thay đổi kể từ lần backup cuối

/N : Chỉ khôi phục các file không có trên đĩa nhận

/D : Hiển thị các file trên đĩa backup phù hợp với đặc tả

/B:date: Chỉ khôi phục các file có thay đổi kể từ trước ngày ....

/A:date: Chỉ khôi phục các file có thay đổi kể từ sau ngày ....

/E:time: Chỉ khôi phục các file có thay đổi kể từ trước thời điểm ....

/L:time: Chỉ khôi phục các file có thay đổi kể từ sau thời điểm ....

Với ví dụ trên, muốn khôi phục thư mục DOS, ta gõ:

```
C:\>RESTORE B: C : /S <Enter>
```

Insert backup diskette 01 in drive B :

Press any key to continue ...

```
*** Files were backed up 04 - 20 - 96 ***
```

```
*** Restoring files from drive B : ***
```

Diskette : 01

Chú ý: Chúng ta không cần chỉ định rõ thư mục con.

### 3. Các chương trình kiểm tra và diệt virus thông dụng

Ta cần có một số đĩa khác phòng ngừa sự mất mát hệ thống:

· Chương trình NORTON UTILITIES có file RESCUE.EXE giúp bạn tạo một đĩa cấp cứu lưu trữ các bảng FAT, Boot Sector, CONFIG.SYS ... của máy vi tính.

· Tạo một đĩa khởi động máy, chứa các tập tin:

COMMAND.COM

IO.SYS

MSDOS.SYS

AUTOEXEC.BAT

CONFIG.SYS

SYS.COM

NDD.EXE

...

· và một số đĩa chứa chương trình quét và diệt virus có phiên bản (version) mới nhất, như:

- SCAN.EXE và CLEAN.EXE của Công ty Mc-Afee, Mỹ

- VSAFE.EXE của MS-DOS, Mỹ

- MSAV.EXE của MS-DOS, Mỹ

- NORTON ANTIVIRUS Của NORTON, Mỹ
- ATV.EXE của Ngô Anh Vũ và Phạm Dũ Liêm, ĐH Kinh tế TP. HCM
- TAV.EXE của Dương Hồng Tấn, Trần Thanh Sơn, Unisoft, TP. HCM
- D2.COM của Trương Minh Nhật Quang, ĐH Tại chức TP. Cần Thơ
- BKAV của Đại học Bách khoa Hà Nội
- v.v..

## CHƯƠNG 6

### CƠ BẢN VỀ NGÔN NGỮ LẬP TRÌNH PASCAL

#### GIỚI THIỆU.

- I. CÁC PHẦN TỬ CƠ BẢN CỦA NGÔN NGỮ PASCAL.
  1. Bộ Ký Tự.
  2. Từ khóa.
  3. Tên chuẩn.
  4. Danh hiệu tự đặt.
- II. CẤU TRÚC MỘT CHƯƠNG TRÌNH PASCAL.
- III. CÁC KIỂU DỮ LIỆU CƠ SỞ : INTEGER, REAL, BOOLEAN, CHAR.
  1. Khái niệm.
  2. Kiểu số nguyên (Integer type).
  3. Kiểu số thực (Real type).
  4. Kiểu logic (Boolean type).
  5. Kiểu ký tự (Char type).
- IV. CÁC KHAI BÁO HẰNG, BIẾN, KIỂU, BIỂU THỨC, ....
  1. Hằng.
  2. Biến.
  3. Kiểu.
  4. Biểu thức.
- V. CÁC THỦ TỤC XUẤT/NHẬP.
  1. Câu lệnh.
  2. Cấu trúc tuần tự.
  3. Cấu trúc rẽ nhánh.
  4. Cấu trúc lặp.

#### BÀI ĐỌC THÊM NHẬP & XUẤT DỮ LIỆU TRONG TURBO PASCAL.

##### **I. GIỚI THIỆU**

Pascal là tên của một trong các ngôn ngữ lập trình cấp cao thông dụng. Ngôn ngữ lập trình Pascal được giáo sư Niklaus Wirth ở trường Đại học Kỹ thuật Zurich (Thụy sĩ) thiết kế và công bố vào năm 1970. Niklaus Wirth đặt tên cho ngôn ngữ này là Pascal để tưởng nhớ đến nhà Toán học và Triết học Pháp ở thế kỷ 17 là Blaise Pascal, người đã phát minh ra một máy tính cơ khí đơn giản đầu tiên của con người.

Ngôn ngữ Pascal được dùng hiện nay có nhiều điểm khác biệt với chuẩn Pascal nguyên thủy của Giáo sư Wirth. Tùy theo quốc gia hoặc công ty đã phát triển cho ra đời các chương trình biên dịch ngôn ngữ Pascal như:

- ISO PASCAL (International Standards Organization) của Châu Âu
- ANSI PASCAL (American National Standards Institute) của Mỹ
- TURBO PASCAL của hãng BORLAND (Mỹ)
- IBM PASCAL của hãng Microsoft (Mỹ)
- v.v...

Đến nay, ngôn ngữ Pascal đã phát triển đến phiên bản Turbo Pascal Version 7. Các diễn giải và ví dụ trong giáo trình này chủ yếu sử dụng chương trình Turbo Pascal 5.5 - 7.0, hiện đang được sử dụng rộng rãi ở Việt Nam.

## II. CÁC PHẦN TỬ CƠ BẢN CỦA NGÔN NGỮ PASCAL

### 1. Bộ ký tự

- Bộ 26 chữ Latin:

Chữ in: A, B, C, ..., X, Y, Z

Chữ thường: a, b, c, ..., x, y, z

- Bộ chữ số thập phân: 0, 1, 2, 3, ..., 8, 9

- Ký tự gạch nối dưới: \_

- Các ký hiệu toán học: +, -, \*, /, =, <, >, (, ), [, ]

### 2. Từ khóa

Là các từ riêng của Pascal, có ngữ nghĩa đã được xác định, không được dùng nó vào các việc khác hoặc đặt tên mới trùng với các từ khóa.

- Từ khóa chung:

PROGRAM, BEGIN, END, PROCEDURE, FUNCTION

- Từ khóa để khai báo:

CONST, VAR, TYPE, ARRAY, STRING, RECORD, SET, FILE, LABEL

- Từ khóa của lệnh lựa chọn:

IF ... THEN ... ELSE, CASE ... OF

- Từ khóa của lệnh lặp:

FOR... TO... DO, FOR... DOWNTO... DO, WHILE... DO,  
REPEAT... UNTIL

- Từ khóa điều khiển:

WITH, GOTO, EXIT, HALT

- Từ khóa toán tử:

AND, OR, NOT, IN, DIV, MOD

### 3. Tên chuẩn

Tên chuẩn là tên đã được định nghĩa sẵn trong Pascal, nhưng người ta có thể định nghĩa lại nếu muốn. Trong Pascal ta có các tên chuẩn sau đây:

Boolean, Char, Integer, Word, Byte, Real, Text

False, True, MaxInt

Abs, Arctan, Chr, Cos, Sin, Eof, Eoln

Exp, Ln, Odd, Ord

Round, Trunc, Sqr, Pred, Succ

Dispose, New, Get, Put, Read, Readln,

Write, Writeln

Reset, Rewrite

### 4. Danh hiệu tự đặt

Trong Pascal để đặt tên cho các biến, hằng, kiểu, chương trình con ta dùng các *danh hiệu (identifier)*. *Danh hiệu của Pascal được bắt đầu bằng một chữ cái, sau đó có thể là các chữ cái, chữ số hay là dấu nối, không được có khoảng trắng và độ dài tối đa cho phép là 127.*

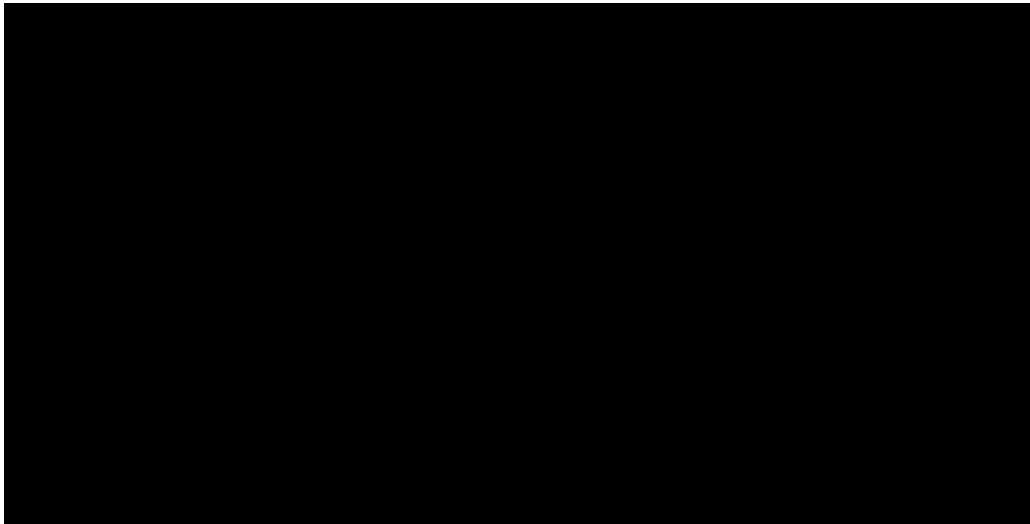
Ví dụ 6.1: Sau đây là các danh hiệu: x; S1; Delta; PT\_bac\_2

*Pascal không phân biệt chữ thường và chữ hoa trong một danh hiệu.*

Ví dụ 6.2: aa và AA là một; XyZ\_aBc và xyZ\_AbC là một

Khi viết chương trình ta nên đặt các danh hiệu sao cho chúng nói lên các ý nghĩa của đối tượng mà chúng biểu thị. Điều này giúp chúng ta viết chương trình dễ dàng và người khác cũng dễ hiểu nội dung chương trình.

## III. CẤU TRÚC MỘT CHƯƠNG TRÌNH PASCAL



Hình 6.1: Sơ đồ cấu trúc chương trình Pascal

Ví dụ 6.3:

```
PROGRAM Hello; { Dòng tiêu đề }

USES Crt; { Lời gọi sử dụng các đơn vị chương trình }

VAR Name : string; { Khai báo biến }

PROCEDURE Input; { Có thể có nhiều Procedure và Function }

 Begin

 ClrScr; { Lệnh xóa màn hình }

 Write(' Hello ! What is your name ?... ');Readln(Name);

 End;

BEGIN { Thân chương trình chính }

 Input;

 Writeln (' Welcome to you, ', Name) ;

 Writeln (' Today, we study PASCAL PROGRAMMING ... ');

 Readln;

 END.
```

Một chương trình Pascal có các phần:

\* Phần tiêu đề:

Phần này bắt đầu bằng từ khóa Program rồi tiếp đến là tên của chương trình và chấm dứt bằng dấu chấm phẩy (;)



Tên chương trình phải được đặt theo đúng qui cách của danh hiệu tự đặt. Phần tiêu đề có hay không cũng được.

\* Phần khai báo dữ liệu:

Trước khi sử dụng biến nào phải khai báo biến đó, nghĩa là xác định rõ xem biến đó thuộc kiểu dữ liệu nào. Một chương trình Pascal có thể có một số hoặc tất cả các khai báo dữ liệu sau:

CONST : khai báo hằng

...

TYPE : định nghĩa kiểu dữ liệu mới

...

VAR : khai báo các biến

...

\* Phần khai báo chương trình con:

Phần này mô tả một nhóm lệnh được đặt tên chung là một chương trình con để khi thân chương trình chính gọi đến thì cả nhóm lệnh đó được thi hành.

Phần này có thể có hoặc không tùy theo nhu cầu.

\* Phần thân chương trình:

Phần thân chương trình là phần quan trọng nhất và bắt buộc phải có, phần này luôn nằm giữa 2 từ khoá là BEGIN và END. Ở giữa là lệnh mà các chương trình chính cần thực hiện. Sau từ khóa END là dấu chấm (.) để báo kết thúc chương trình.

\* Dấu chấm phẩy (;):

Dấu ; dùng để ngăn cách các câu lệnh của Pascal và không thể thiếu được.

\* **Lời chú thích:**

Lời chú thích dùng để chú giải cho người sử dụng chương trình nhớ nhằm trao đổi thông tin giữa người và người, máy tính sẽ không để ý đến lời chú thích này. Lời chú thích nằm giữa ký hiệu: { } hoặc (\* \*)

#### **IV. CÁC KIỂU DỮ LIỆU CƠ SỞ: INTEGER, REAL, BOOLEAN, CHAR**

##### **1. Khái niệm**

Dữ liệu (data) là tất cả những gì mà máy tính phải xử lý.

Theo Niklaus Wirth:

**CHƯƠNG TRÌNH = THUẬT TOÁN + CẤU TRÚC DỮ LIỆU**

Một kiểu dữ liệu (data type) là một qui định về hình dạng, cấu trúc và giá trị của dữ liệu cũng như cách biểu diễn và cách xử lý dữ liệu.

Trong Pascal các kiểu dữ liệu gồm các loại sau:

- Kiểu đơn giản (Simple type): bao gồm kiểu số nguyên (Integer), kiểu số thực (Real), kiểu logic (Boolean), kiểu ký tự (Char).
- Kiểu có cấu trúc (Structure type): bao gồm mảng (Array), chuỗi (String), bản ghi (Record), tập hợp (Set), tập tin (File).
- Kiểu chỉ điểm (pointer):

Trong chương này, chúng ta chỉ xét các kiểu dữ liệu đơn giản.

## 2. Kiểu số nguyên (Integer type)

### a. Kiểu số nguyên thuộc Z chứa trong Turbo Pascal

Được định nghĩa với các từ khóa sau:

| TỪ KHÓA  | SỐ BYTE | PHẠM VI                    |
|----------|---------|----------------------------|
| BYTE     | 1       | 0 .. 255                   |
| SHORTINT | 1       | - 128 .. 127               |
| INTEGER  | 2       | - 32768 .. + 32767         |
| WORD     | 2       | 0 .. 65535                 |
| LONGINT  | 4       | - 2147483648 .. 2147483647 |

### b. Các phép toán số học đối với số nguyên

| KÝ HIỆU  | Ý NGHĨA                              |
|----------|--------------------------------------|
| +        | Cộng                                 |
| -        | Trừ                                  |
| *        | Nhân                                 |
| /        | Chia cho kết quả là số thực          |
| DIV      | Chia lấy phần nguyên                 |
| MOD      | Chia lấy phần dư                     |
| SUCC (n) | n + 1                                |
| PRED (n) | n - 1                                |
| ODD (n)  | TRUE nếu n lẻ<br>và FALSE nếu n chẵn |

## 3. Kiểu số thực (Real type)

Ở Turbo Pascal, kiểu số thực thuộc tập hợp R chứa trong 6 bytes, được định nghĩa với từ khóa REAL:

$$R = ([2.9 \times 10^{-39}, 1.7 \times 10^{38}])$$

Hay viết theo dạng số khoa học:  $R = ([2.9E-39, 1.7E38])$

Số thực có thể viết theo kiểu có dấu chấm thập phân bình thường hoặc viết theo kiểu thập phân có phần mũ và phần định trị.

Các phép toán số học cơ bản +, -, \*, / dĩ nhiên được sử dụng trong kiểu real.

Bảng dưới đây là các hàm số học cho kiểu số thực:

| KÝ HIỆU | Ý NGHĨA |
|---------|---------|
|---------|---------|

|            |                                                  |
|------------|--------------------------------------------------|
| ABS (x)    | x  : lấy giá trị tuyệt đối của số x              |
| SQR (x)    | x2 : lấy bình phương trị số x                    |
| SQRT(x)    | ■ : lấy căn bậc 2 của số x                       |
| SIN(x)     | sin (x) : lấy sin của x                          |
| COS (x)    | cos (x) : lấy cos của x                          |
| ARCTAN (x) | arctang (x)                                      |
| LN (x)     | ln x : lấy logarit nepe của trị x (e ( 2.71828)) |
| EXP (x)    | e <sup>x</sup>                                   |
| TRUNC (x)  | lấy phần nguyên lớn nhất không vượt quá trị số x |
| ROUND (x)  | làm tròn giá trị của x, lấy số nguyên gần x nhất |

#### 4. Kiểu logic (Boolean)

Một dữ liệu thuộc kiểu BOOLEAN là một đại lượng được chứa trong 1 byte ở Turbo Pascal và chỉ có thể nhận được một trong hai giá trị logic là TRUE (đúng) và FALSE (sai).

Qui ước: TRUE > FALSE

Các phép toán trên kiểu Boolean:

| A     | B     | NOT A | A AND B | A OR B | A XOR B |
|-------|-------|-------|---------|--------|---------|
| TRUE  | TRUE  | FALSE | TRUE    | TRUE   | FALSE   |
| TRUE  | FALSE | FALSE | FALSE   | TRUE   | TRUE    |
| FALSE | TRUE  | TRUE  | FALSE   | TRUE   | TRUE    |
| FALSE | FALSE | TRUE  | FALSE   | FALSE  | FALSE   |

Nhận xét:

- Phép AND (và) chỉ cho kết quả là TRUE khi cả 2 toán hạng là TRUE
- Phép OR (hoặc) chỉ cho kết quả là FALSE khi cả 2 toán hạng là FALSE
- Phép XOR (hoặc triệt tiêu) luôn cho kết quả là TRUE khi cả 2 toán hạng là khác nhau và ngược lại.

Các phép toán quan hệ cho kết quả kiểu Boolean:

| KÝ HIỆU | Ý NGHĨA           |
|---------|-------------------|
| < >     | khác nhau         |
| =       | bằng nhau         |
| >       | lớn hơn           |
| <       | nhỏ hơn           |
| > =     | lớn hơn hoặc bằng |
| < =     | nhỏ hơn hoặc bằng |

#### 5. Kiểu ký tự (Char type)

Tất cả các dữ liệu viết ở dạng chữ ký tự được khai báo bởi từ khóa CHAR.

Một ký tự được viết trong hai dấu nháy đơn ( ). Để tiện trao đổi thông tin cần phải sắp xếp, đánh số các ký tự, mỗi cách sắp xếp như vậy gọi là bảng mã. Bảng mã thông dụng hiện nay là bảng mã ASCII (xem lại chương 3).

Để thực hiện các phép toán số học và so sánh, ta dựa vào giá trị số thứ tự mã ASCII của từng ký tự, chẳng hạn: 'A' < 'a' vì số thứ tự mã ASCII tương ứng là 65 và 97.

Trong Turbo Pascal mỗi ký tự được chứa trong 1 byte.

Các hàm chuẩn liên quan đến kiểu ký tự:

| KÝ HIỆU       | Ý NGHĨA                                 |
|---------------|-----------------------------------------|
| ORD(x)        | Cho số thứ tự của ký tự x trong bảng mã |
| CHR(n) hay #n | Cho ký tự có số thứ tự là n             |
| PRED(x)       | Cho ký tự đứng trước x                  |
| SUCC(x)       | Cho ký tự đứng sau x                    |

## V. CÁC KHAI BÁO HẰNG, BIẾN, KIỂU, BIỂU THỨC, ...

### 1. Hằng (constant)

#### a. Định nghĩa

Hằng là một đại lượng có giá trị không đổi trong quá trình chạy chương trình. Ta dùng tên hằng để chương trình được rõ ràng và dễ sửa đổi.

#### b. Cách khai báo

##### CONST

<Tên hằng> = <giá trị của hằng> ;

Ví dụ 6.4: CONST

Siso = 100;

X = 'xxx ';

### 2. Biến (variable)

#### a. Định nghĩa

Biến là một cấu trúc ghi nhớ có tên (đó là tên biến hay danh hiệu của biến).

Biến ghi nhớ một dữ liệu nào đó gọi là giá trị (value) của biến. Giá trị của biến có thể được biến đổi trong thời gian sử dụng biến.

Sự truy xuất của biến nghĩa là đọc giá trị hay thay đổi giá trị của biến được thực hiện thông qua tên biến.

Ví dụ 6.5: Readln (x) ;

Writeln (x) ;

x := 9 ;

Biến là một cấu trúc ghi nhớ dữ liệu vì vậy nó phải tuân theo qui định của kiểu dữ liệu : một biến phải thuộc một kiểu dữ liệu nhất định.

#### b. Cách khai báo

## VAR

<Tên biến> : <Kiểu biến> ;

Ví dụ 6.6: VAR

a : Real ;

b, c : Integer ;

TEN : String [20]

X : Boolean ;

Chon : Char ;

Cần khai báo các biến trước khi sử dụng chúng trong chương trình. Khai báo một biến là khai báo sự tồn tại của biến đó và cho biết nó thuộc kiểu gì.

### 3. Kiểu (Type)

#### a. Định nghĩa

Ngoài các kiểu đã định sẵn, Pascal còn cho phép ta định nghĩa các kiểu dữ liệu khác từ các kiểu căn bản theo qui tắc xây dựng của Pascal.

#### b. Cách khai báo

## TYPE

<Tên kiểu> = <Mô tả xây dựng kiểu> ;

Ví dụ 6.7:

## TYPE

SoNguyen = Integer ;

Diem = Real;

Tuoi = 1 .. 100 ;

Color = (Red, Blue, Green) ;

Thu = (Sun, Mon, Tue, Wed, Thu, Fri, Sat) ;

và khi đã khai báo kiểu gì thì ta có quyền sử dụng để khai báo biến như ở ví dụ sau:

Ví dụ 6.8: VAR

i, j : SoNguyen ;

Dtb : Diem ;

T : tuoi ;

Mau : Color ;

Ngay\_hoc : Thu;

#### 4. Biểu thức (Expression)

##### a. Định nghĩa

Một biểu thức là một công thức tính toán bao gồm các phép toán, hằng, biến, hàm và các dấu ngoặc.

Ví dụ 6.9:  $5 + A * \text{SQRT}(B) / \text{SIN}(X)$

$(A \text{ AND } B) \text{ OR } C$

##### b. Thứ tự ưu tiên

Khi tính giá trị của một biểu thức, ngôn ngữ Pascal qui ước thứ tự ưu tiên của các phép toán từ cao đến thấp như sau:

| Mức ưu tiên: | Các phép toán:                |
|--------------|-------------------------------|
| 1.           | Biểu thức trong ngoặc đơn ( ) |
| 2.           | Phép gọi hàm                  |
| 3.           | Not, -                        |
| 4.           | *, /, DIV, MOD, AND           |
| 5.           | +, -, OR, XOR                 |
| 6.           | =, <>, <=, >=, <, >, IN       |

Ví dụ 6.10:  $(4+5)/3 + 6 - (\sin((/2)+3))*2 = (9)/3 + 6 - (1+3)*2 = 3 + 6 - 8 = 1$

##### c. Qui ước tính thứ tự ưu tiên

Khi tính một biểu thức có 3 qui tắc về thứ tự ưu tiên như sau:

**Qui tắc 1 :** Các phép toán nào có ưu tiên cao hơn sẽ được tính trước.

**Qui tắc 2 :** Trong các phép toán có cùng thứ tự ưu tiên thì sự tính toán sẽ được thực hiện từ trái sang phải.

**Qui tắc 3 :** Phần trong ngoặc từ trong ra ngoài được tính toán để trở thành một giá trị đơn.

##### d. Kiểu của biểu thức

Là kiểu của kết quả sau khi tính biểu thức.

Ví dụ 6.11: Biểu thức sau được gọi là biểu thức Boolean:

$\text{not } (('a' > 'c') \text{ and } ('c' > 'C')) \text{ or } ('B' = 'b')$  có giá trị TRUE

## VI. CÁC THỦ TỤC XUẤT/NHẬP

### 1. Câu lệnh (statement)

**a. Trong một chương trình Pascal, sau phần mô tả dữ liệu là phần mô tả các câu lệnh.** Các câu lệnh có nhiệm vụ xác định các công việc mà máy tính phải thực hiện để xử lý các dữ liệu đã được mô tả và khai báo.

**b. Câu lệnh được chia thành câu lệnh đơn giản và câu lệnh có cấu trúc.**

(xem phần bài đọc thêm)

- Câu lệnh đơn giản

+ Vào dữ liệu : Read, Readln

+ Ra dữ liệu : Write, Writeln

+ Lệnh gán : :=

+ Lệnh gọi chương trình con (gọi trực tiếp tên của chương trình con)

+ Xử lý tập tin : RESET, REWRITE, ASSIGN ...

- Câu lệnh có cấu trúc

+ Lệnh ghép : BEGIN .. END

+ Lệnh chọn : IF .. THEN .. ELSE

CASE .. OF .

+ Lệnh lặp : FOR .. TO .. DO

REPEAT .. UNTIL

WHILE .. DO

**c. Các câu lệnh phải được ngăn cách với nhau bởi dấu chấm phẩy ( ; ) và Các câu lệnh có thể viết trên một dòng hay nhiều dòng.**

### 2. Cấu trúc tuần tự

#### a. Lệnh gán (Assignment statement)

Một trong các lệnh đơn giản và cơ bản nhất của Pascal là lệnh gán. Mục đích của lệnh này là gán cho một biến đã khai báo một giá trị nào đó cùng kiểu với biến.

\* Cách viết:

<Tên\_biến> := <biểu thức> ;

Ví dụ 6.12: Khi đã khai báo

VAR

c : Char ;

i,j : Integer ;

x, y : Real ;

p, q : Boolean ;

thì ta có thể có các phép gán sau :

c := 'A' ;

c := Chr(90) ;

i := (35+7)\*2 mod 4 ;

i := i div 7 ;

x := 0.5 ;

x := i + 1 ;

q := i > 2\*j + 1 ;

q := not p ;

**\* Ý nghĩa:**

Biến và các phát biểu gán là các khái niệm quan trọng của một họ các ngôn ngữ lập trình mà Pascal là một đại diện tiêu biểu. Chúng phản ánh cách thức hoạt động của máy tính hiện nay, đó là:

- Lưu trữ các giá trị khác nhau vào một ô nhớ tại những thời điểm khác nhau.

- Một quá trình tính toán có thể coi như là một quá trình làm thay đổi giá trị của một (hay một số) ô nhớ nào đó, cho đến khi đạt được giá trị cần tìm.

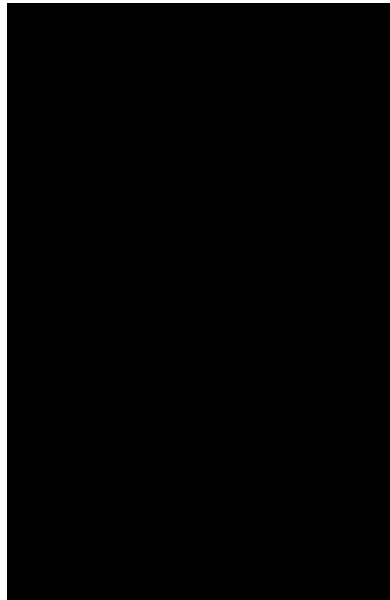
**b. Lệnh ghép (Compound statement)**

Một nhóm câu lệnh đơn được đặt giữa 2 chữ BEGIN và END sẽ tạo thành một câu lệnh ghép.

Trong Pascal ta có thể đặt các lệnh ghép con trong các lệnh ghép lớn hơn bao ngoài của nó và có thể hiểu tương tự như cấu trúc ngoặc đơn ( ) trong các biểu thức toán học.

\* Sơ đồ:





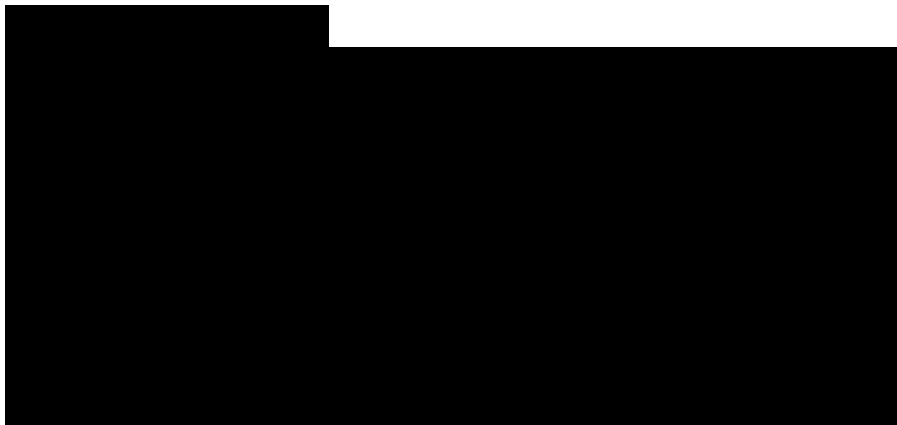
Hình 6.2: Sơ đồ cấu trúc BEGIN .. END;

Ở hình minh họa trên ta dễ thấy các nhóm lệnh thành từng khối (block). Một khối lệnh bắt đầu bằng BEGIN và chấm dứt ở END; . Trong một khối lệnh cũng có thể có các khối lệnh con nằm trong nó. Một khối chương trình thường được dùng để nhóm từ 2 lệnh trở lên để tạo thành một <Công việc> của các lệnh có cấu trúc, ta có thể gặp khái niệm này trong nhiều ví dụ ở các phần sau.

### 3. Cấu trúc rẽ nhánh

#### a. Lệnh IF .. THEN .. và Lệnh IF .. THEN .. ELSE..

\* Lưu đồ diễn tả các lệnh và ý nghĩa cách viết:



Hình 6. 3: Lệnh IF <Điều kiện> THEN <Công việc>;



Hình 6. 4: Lệnh IF .. THEN .. ELSE ..;

Chú ý:

- Điều kiện là một biểu thức Boolean.

- Nếu <Công việc>sau THEN hoặc ELSE có nhiều hơn một lệnh thì ta phải gói lại trong BEGIN .. END;

- Toàn bộ lệnh IF .. THEN .. ELSE xem như 1 lệnh đơn.

Ví dụ 6.13: Tính căn bậc 2 của một số

```
PROGRAM Tinh_can_bac_hai ;
```

```
VAR
```

```
 a : Real ;
```

```
BEGIN
```

```
 Write (' Nhập số a = ') ;
```

```
 Readln(a) ;
```

```
 IF a < 0 THEN
```

```
 Write (' a : 10 : 2 , là số âm nên không lấy căn được !!! ')
```

```
 ELSE
```

```
 Writeln (' Căn số bậc 2 của , a : 2 : 2 , là , SQRT(a) :10 : 3 ');
```

```
 Writeln (' Nhấn ENTER để thoát ... ');
```

```
 Readln; {Dừng màn hình để xem kết quả}
```

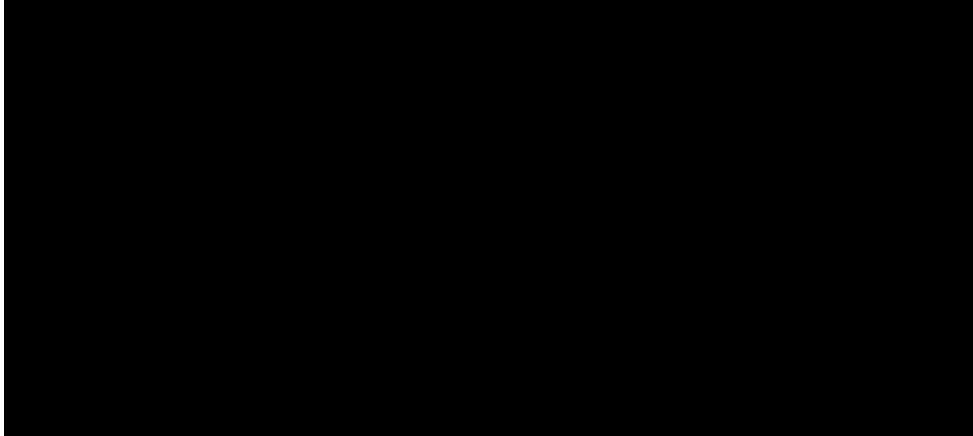
```
END.
```

Ghi chú:

Trong chương trình trên, a ta thấy có dạng a :m :n với ý nghĩa m là số định khoảng mà phần nguyên của a sẽ chiếm chỗ và n là khoảng cho số trị phần thập phân của a.

**b. Lệnh CASE .. OF**

\* Lưu đồ biểu diễn:



Hình 6.5: Lưu đồ lệnh CASE .. OF

\* **Cách viết, ý nghĩa:**

| Cách viết            | Ý nghĩa                                                 |
|----------------------|---------------------------------------------------------|
| CASE <Biểu thức > OF | Xét giá trị của biểu thức chọn                          |
| GT1 : Công việc 1 ;  | Nếu có giá trị 1 (GT1) thì thi hành Công việc 1         |
| .....                | .....                                                   |
| GTi : Công việc i ;  | Nếu có giá trị i (GT i) thì thi hành Công việc i        |
| .....                | .....                                                   |
| ELSE Công việc 0 ;   | Nếu không có giá trị nào thỏa thì thực hiện Công việc 0 |
| END;                 |                                                         |

Ghi chú:

- Lệnh CASE .. OF có thể không có ELSE
- Biểu thức chọn là kiểu rời rạc như Integer, Char, không chọn kiểu Real
- Nếu muốn ứng với nhiều giá trị khác nhau của biểu thức chọn vẫn thi hành một lệnh thì giá trị đó có thể viết trên cùng một hàng cách nhau bởi dấu phẩy (,) : Giá trị k1, k2, ..., kp : Lệnh k ;

Ví dụ 6.14: PROGRAM Chon\_mau ;

VAR color : char ;

BEGIN

write (' Chọn màu theo một trong 3 ký tự đầu là R / W / B ');

readln ( color );

CASE color OF

'R' , 'r' : write ( ' RED = màu đỏ ');

'W' , 'w' : write ( ' WHITE = màu trắng ');

'B' , 'b' : write ( ' BLUE = màu xanh dương ');

END ;

Readln;

END.

#### 4. Cấu trúc lặp

##### a. Lệnh FOR

Cấu trúc FOR cho phép lặp lại nhiều lần một dãy lệnh. Số lần lặp lại dãy lệnh đã biết trước. Phát biểu FOR có 2 dạng:

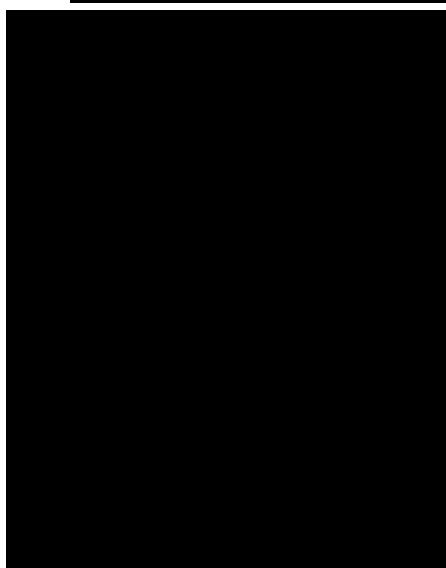
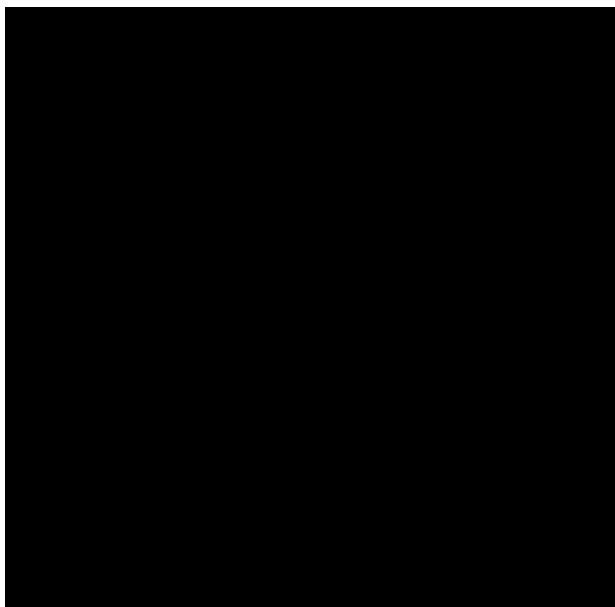
FOR .. TO .. DO                      đếm lên

FOR .. DOWNTO ..DO                đếm xuống

*\* Cú pháp tổng quát là:*

**FOR <biến đếm> := <trị đầu> TO/DOWNTO <trị cuối> DO <Công việc>;**

\* Lưu đồ:



Hình 6. 6: Lưu đồ phát biểu FOR .. TO .. DO

Chú ý: Trị đầu, trị cuối là các biến hoặc hằng và biến đếm phải là kiểu rời rạc.

Ví dụ 6.15: Chương trình in một dãy số từ 0 đến 9

```
PROGRAM Day_So ;

VAR
 i : Integer ;

BEGIN
 FOR i := 0 TO 9 DO Write (i) ;
 Readln ;

END.
```

#### b. Lệnh WHILE .. DO

*\* Lưu đồ của lệnh*



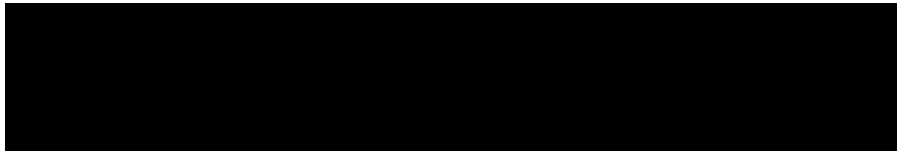
Hình 6. 7: Lưu đồ cấu trúc WHILE .. DO

**\* Ý nghĩa lưu đồ:**

**Trong khi mà điều kiện còn đúng thì cứ thực hiện Công việc, rồi quay trở về kiểm tra điều kiện lại.** Vòng lặp được tiếp tục, đến khi điều kiện đặt ra không còn đúng nữa thì đi tới thực hiện lệnh tiếp theo

**\* Cú pháp**

**WHILE <điều kiện> DO <Công việc>**



Hình 6.8: Sơ đồ cú pháp lệnh WHILE .. DO

**Ghi chú:**

- Điều kiện trong cấu trúc lặp WHILE .. DO là một biểu thức logic kiểu Boolean chỉ có 2 giá trị là Đúng (True) hoặc Sai (False)
- Nếu điều kiện Đúng thì chương trình sẽ chạy trong cấu trúc WHILE .. DO.
- Sau mỗi lần lặp, chương trình trở lại kiểm tra điều kiện. Tùy theo biểu thức logic của điều kiện là Đúng hay Sai thì chương trình sẽ thực hiện Công việc tương ứng.
- Nếu Sai thì chuyển xuống dưới cấu trúc WHILE .. DO

Ví dụ 6.16: Chương trình tính trung bình n số:  $x_1 + x_2 + x_3 + \dots + x_n$

Program Trung\_binh\_Day\_So ;

VAR

n, count : Integer ;

x, sum, average : real ;

BEGIN

count := 1 ;

sum := 0 ;

```
Write (' Nhập n = ') ;

readln (n) ;

WHILE count < n+1 DO

 BEGIN

 Write (' Nhập giá trị thứ' , count,' của x = ') ;

 readln (x) ;

 sum := sum + x ;

 count := count + 1 ;

 END ;

average := sum/n ;

Writeln (' Trung bình là =' , average : 10 : 3) ;

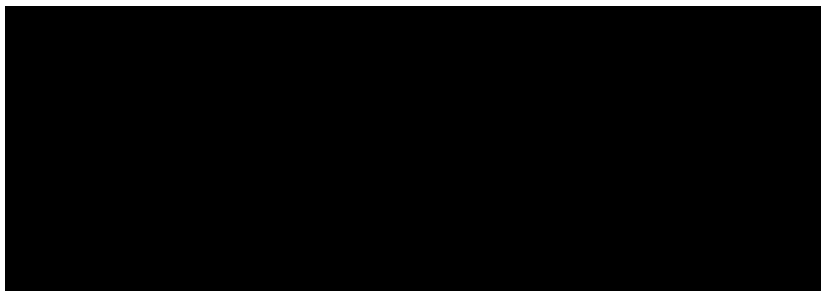
Writeln (' Nhấn Enter để thoát ...') ;

Readln ;

END.
```

### c. Lệnh REPEAT .. UNTIL

Câu lệnh REPEAT .. UNTIL dùng trong các trường hợp khi biến điều khiển không có kiểu rời rạc và đặc biệt trong các trường hợp số lần lặp không biết trước.



Hình 6.9: Lưu đồ cấu trúc của REPEAT .. UNTIL

#### \* Ý nghĩa câu lệnh:

Nếu điều kiện logic là Sai (False) thì lặp lại lệnh cho đến khi điều kiện Đúng thì mới thoát ra khỏi cấu trúc REPEAT .. UNTIL.

Nếu có nhiều câu lệnh thì mỗi lệnh ngăn cách nhau bằng dấu chấm phẩy (;) Công việc của REPEAT và UNTIL không nhất thiết phải dùng lệnh ghép để nhóm từ 2 lệnh đơn trở lên thành công việc.



Hình 6.10: Sơ đồ cú pháp REPEAT .. UNTIL

Ví dụ 6.17: Với bài toán trung bình cộng một dãy số ở ví dụ trước có thể viết theo cấu trúc REPEAT .. UNTIL như sau:

```
PROGRAM Trung_binh_Day_So ;

VAR n, count : Integer ;

 x, sum : real ;

BEGIN

 count := 1 ;

 sum := 0 ;

 Write := (' Nhập n = ') ; readln (n) ;

 REPEAT

 Write (' Nhập giá trị thứ' , count, ' của x = ') ;

 readln(x) ;

 sum := sum + x ;

 count := count + 1 ;

 UNTIL count > n ;

 Writeln (' Trung bình là = ' , sum/n : 8 :2) ;

 Readln ;

END.
```

Ghi chú:

So sánh 2 cách viết WHILE .. DO và REPEAT .. UNTIL ta thấy có sự khác biệt:

- Trong cấu trúc WHILE .. DO thì <Điều kiện> được kiểm tra trước, nếu thỏa <Điều kiện> thì mới thực hiện <Công việc>.
- Ngược lại, trong cấu trúc REPEAT .. UNTIL thì <Công việc> sẽ được thực thi trước sau đó mới kiểm tra <Điều kiện>, nếu không thỏa <Điều kiện> thì tiếp tục thi hành <Công việc> cho đến khi <Điều kiện> là đúng.



Lệnh REPEAT .. UNTIL thường được sử dụng trong lập trình, nhất là lúc người sử dụng muốn tiếp tục bài toán ở trường hợp thay đổi biến mà không phải trở về chương trình và nhấn tổ hợp phím Ctrl + F9 lại.

Ví dụ 6.18: Nhân 2 số a và b

```
PROGRAM Tich;

VAR a, b : integer ;

 CK : char ;

BEGIN

 REPEAT

 Write (' Nhập số a = '); Readln (a) ;

 Write (' Nhập số b = '); Readln (b) ;

 Writeln (' Tích số của a x b là :', a*b : 10) ;

 Writeln (' Tiếp tục tính nữa không (CK) ? ');

 Readln (CK) ;

 UNTIL upcase(CK) = K; {hàm chuyển đổi ký tự trong biến}
 {CK thành ký tự in hoa}

END.
```

## BÀI ĐỌC THÊM

### NHẬP VÀ XUẤT DỮ LIỆU TRONG TURBO PASCAL

--- oOo ---

Thông thường, chương trình Turbo Pascal được đặt trong một thư mục riêng rẽ có tên TP. Để sử dụng Turbo Pascal, ta cần có các tập tin tối thiểu:

- TURBO.EXE
- TURBO.TPL
- TURBO.TP
- GRAPH.TPU
- Các file đồ họa : \*.BGI
- Các Font chữ trong đồ họa : \*.CHR

Sử dụng câu lệnh Turbo và nhấn Enter, màn hình sẽ xuất hiện :



Để trợ giúp người sử dụng, phím chức năng F10 có tác dụng mở các Menu với nhiều Options khác nhau. Ta cũng có thể kích hoạt trên thanh Menu chính bằng cách kết hợp phím <Alt - Ký tự mục tương ứng>, ví dụ để kích hoạt mục File, ta nhấn đồng thời phím Alt- F, sau đó dùng các phím mũi tên và nút Enter để chọn lựa và ra lệnh thi hành. Phím F1 trợ giúp thể hiện các thông tin trên màn hình.

Ta có thể sử dụng các tổ hợp phím để tạo ra các khối chữ hoặc câu lệnh (trên màn hình thấy có sự thay đổi màu) để ta có thể sao chép, cắt dán, xóa bỏ ...

- Ctrl-K-B      Đánh dấu đầu khối
- Ctrl-K-K      Đánh dấu cuối khối
- Ctrl-K-C      Chép khối tại sau vị trí con trỏ
- Ctrl-K-V      Di chuyển khối tới sau vị trí con trỏ
- Ctrl-K-Y      Xóa khối hiện hành
- Ctrl-K-W      Ghi khối hiện hành vào đĩa như một tập tin
- Ctrl-K-R      Đọc khối tập tin đã ghi vào đĩa vào sau vị trí con trỏ
- Ctrl-K-H      Tắt/ Mở khối

Một chương trình máy tính, sẽ có các bước căn bản sau:



Trong thảo chương Turbo Pascal, các thủ tục nhập dữ liệu được dùng:

**THỦ TỤC NHẬP**

**Ý NGHĨA**

READ(x1, x2, ..., xn)      Nhập các biến x1, x2, ..., xn theo hàng ngang từ bàn phím (con trỏ không xuống hàng).

|                            |                                                                                            |
|----------------------------|--------------------------------------------------------------------------------------------|
| READLN(x1, x2, ..., xn)    | Nhập các biến x1, x2, ..., xn theo hàng dọc từ bàn phím (mỗi lần nhập con trỏ xuống hàng). |
| READLN;                    | Dừng chương trình, đợi Enter mới tiếp tục.                                                 |
| ASSIGN(F, File_Name);      | Mở tập tin F có tên là File_Name                                                           |
| RESET(F);                  | Chuẩn bị đọc tập tin                                                                       |
| READ(F, x1, x2, ..., xn) ; | Đọc các giá trị trên tập tin F ra các biến x1, x2, ..., xn tương ứng                       |
| CH := ReadKey ;            | Đọc một ký tự từ bàn phím vào biến ký tự CH                                                |
| KEYPRESSED                 | Một hàm có giá trị là TRUE nếu có một phím được bấm và là FALSE nếu ngược lại.             |

### **THỦ TỤC XUẤT**

### **Ý NGHĨA**

|                                |                                                                                                               |
|--------------------------------|---------------------------------------------------------------------------------------------------------------|
| WRITE(x1, x2, ..., xn)         | Viết giá trị trong các biến x1, x2, ..., xn ra màn hình theo hàng ngang (con trỏ không xuống hàng).           |
| WRITELN(x1, x2, ..., xn)       | Viết giá trị trong các biến x1, x2, ..., xn ra màn hình theo hàng dọc (mỗi lần viết trị x có xuống hàng).     |
| WRITELN;                       | Xuống hàng                                                                                                    |
| WRITELN(I : n);                | Viết ra giá trị của biến nguyên I vào n chỗ tính từ phải sang trái. Nếu dư chỗ (chữ số của I < n) sẽ để trống |
| WRITELN(R : n : m);            | Viết ra giá trị của biến thực R vào n chỗ, chỉ lấy m số thập phân.                                            |
| WRITELN( abc... );             | Viết ra nguyên văn chuỗi ký tự abc...                                                                         |
| WRITELN (LST, x1, x2, ..., xn) | Viết ra máy in các trị biến x1, x2, ..., xn                                                                   |
| ASSIGN(F, File_Name)           | Mở tập tin F có tên là File_Name                                                                              |
| REWRITE(F) ;                   | để chuẩn bị viết vào                                                                                          |
| WRITE (F, x1, x2, ..., xn) ;   | Viết các giá trị x1, x2, ..., xn vào tập tin F                                                                |
| CLOSE (F) ;                    | Đóng tập tin F                                                                                                |

Cần lưu trữ chương trình ta dùng phím F2.

Mở một file đã có ta dùng phím F3.

Để thay đổi kích thước/Di chuyển cửa sổ chương trình, dùng phím F5 và Ctrl+F5.

Trường hợp mở nhiều chương trình, ta dùng phím F6 và Ctrl+F6 để đi đến/trở về trước chương trình hiện hành.

Để biên dịch và kiểm tra lỗi, ta dùng phím F9.

Để chạy chương trình đã soạn thảo xong, đánh Ctrl+F9

Muốn thoát khỏi Turbo Pascal và trở về DOS, đánh Alt+X.

## CHƯƠNG 7

### CHƯƠNG TRÌNH CON VÀ ĐƠN VỊ CHƯƠNG TRÌNH

- I. [KHÁI NIỆM VỀ CHƯƠNG TRÌNH CON.](#)
- II. [THỦ TỤC VÀ HÀM.](#)
  1. [Thủ tục.](#)
  2. [Hàm.](#)
- III. [TRUYỀN THAM SỐ CHO CHƯƠNG TRÌNH CON.](#)
- IV. [TÍNH ĐỀ QUI TRONG CHƯƠNG TRÌNH CON.](#)
- V. [UNIT.](#)
  1. [Khái niệm.](#)
  2. [Các Unit chuẩn.](#)
- VI. [UNIT TỰ TẠO.](#)
  1. [Các bước để tạo ra một Unit.](#)
  2. [Ví dụ.](#)

#### [BÀI ĐỌC THÊM LẬP TRÌNH THEO CẤU TRÚC TRÊN - XUỐNG.](#)

### I. KHÁI NIỆM VỀ CHƯƠNG TRÌNH CON

Khi lập trình, chúng ta thường có những đoạn chương trình hay phép tính lặp lại nhiều lần. Nếu mỗi lần lặp lại, ta phải viết những đoạn lệnh như nhau thì chương trình của chúng ta trở nên dài dòng, rối rắm và mất thời gian vô ích. Để giải quyết những trường hợp như vậy, Pascal cho phép chúng ta tạo ra các module, mỗi module mang một đoạn chương trình gọi là chương trình con (subroutine hay subprogram). Mỗi chương trình con sẽ mang một cái tên khác nhau. Một module chỉ cần viết một lần và sau đó chúng ta có thể truy xuất nó nhiều lần, bất kỳ nơi nào trong chương trình chính. Khi cần thiết, chúng ta chỉ việc gọi tên chương trình con đó ra để thi hành lệnh.

Nhờ sử dụng chương trình con, chương trình có thể tiết kiệm được ô nhớ. Đồng thời, ta có thể kiểm tra tính logic trong tiến trình lập trình cho máy tính điện tử, có thể nhanh chóng loại bỏ những sai sót khi cần hiệu chỉnh hay cải tiến chương trình. Đây là khái niệm cơ bản trong ý tưởng lập chương trình có cấu trúc. Một quá trình tính cũng có thể có nhiều chương trình con lồng ghép vào nhau.

Trong Pascal, chương trình con được viết dưới dạng thủ tục (procedure) và **hàm (function)**. **Cấu trúc của 2 kiểu chương trình con này thì tương tự với nhau, mặc dầu cách truy xuất của chúng có khác nhau và cách trao đổi thông tin trong mỗi kiểu cũng có điểm khác nhau.** Hàm (function) trả lại một giá trị kết quả vô hướng thông qua tên hàm và hàm được sử dụng trong biểu thức.

Ví dụ hàm chuẩn, như hàm  $\sin(x)$  mà chúng ta đã biết trong chương trước có thể được xem như một chương trình con kiểu function với tên là  $\sin$  và tham số là  $x$ . Trong khi đó, thủ tục (procedure) không trả lại kết quả thông qua tên của nó, do vậy, ta không thể viết các thủ tục trong biểu thức. Các lệnh `Writeln`, `Readln` trong chương trước được xem như các thủ tục chuẩn.

Một chương trình có chương trình con tự thiết lập có 3 khối (block) :

- \* Khối khai báo
- \* Khối chương trình con
- \* Khối chương trình chính

## II . THỦ TỤC VÀ HÀM

\* Một số khái niệm biến:

· **Biến toàn cục (global variable):** Còn được gọi là **biến chung**, là biến được khai báo ở đầu chương trình, nó được sử dụng bên trong chương trình chính và cả bên trong chương trình con. Biến toàn cục sẽ tồn tại trong suốt quá trình thực hiện chương trình.

· **Biến cục bộ (local variable):** Còn được gọi là **biến riêng**, là biến được khai báo ở đầu chương trình con, và nó chỉ được sử dụng bên trong thân chương trình con hoặc bên trong thân chương trình con khác nằm bên trong nó (các chương trình con lồng nhau). Biến cục bộ chỉ tồn tại khi chương trình con đang hoạt động, nghĩa là biến cục bộ sẽ được cấp phát bộ nhớ khi chương trình con được gọi để thi hành, và nó sẽ được giải phóng ngay sau khi chương trình con kết thúc.

· **Tham số thực (actual parameter)** là một tham số mà nó có thể là một biến toàn cục, một biểu thức hoặc một giá trị số (cũng có thể biến cục bộ khi sử dụng chương trình con lồng nhau) mà ta dùng chúng khi truyền giá trị cho các tham số hình thức tương ứng của chương trình con.

· **Tham số hình thức (formal parameter)** là các biến được khai báo ngay sau **Tên chương trình con**, nó dùng để nhận giá trị của các tham số thực truyền đến. Tham số hình thức cũng là một biến cục bộ, ta có thể xem nó như là các đối số của hàm toán học.

\* Lời gọi chương trình con (thủ tục và hàm):

Để chương trình con được thi hành, ta phải có lời gọi đến chương trình con, lời gọi chương trình con thông qua tên chương trình con và danh sách các tham số tương ứng (nếu có). Các qui tắc của lời gọi chương trình con:

· Trong thân chương trình chính hoặc thân chương trình con, ta chỉ có thể gọi tới các chương trình con trực thuộc nó.

· Trong chương trình con, ta có thể gọi các chương trình con ngang cấp đã được thiết lập trước đó.

### 1. Thủ tục (Procedure):

Thủ tục là một đoạn cấu trúc chương trình được chứa bên trong chương trình Pascal như là một chương trình con. Thủ tục được đặt tên và có thể chứa danh sách tham số hình thức (formal parameters). Các tham số này phải được đặt trong dấu ngoặc đơn ( ). Ta có thể truy xuất thủ tục bằng cách gọi tên của thủ tục. Chương trình sẽ tự động truy xuất thủ tục đúng tên đã gọi và thực hiện các lệnh chứa trong thủ tục đó. Sau khi thực hiện thủ tục xong, chương trình sẽ trở lại ngay lập tức sau vị trí câu lệnh gọi thủ tục đó.

Có 2 loại thủ tục:

+ thủ tục không tham số

+ và thủ tục có tham số.

**a. Cấu trúc của thủ tục không tham số**

```
PROCEDURE < Tên thủ tục > ;
 { Các khai báo hằng, biến, kiểu cục bộ... }
BEGIN
 { ... các lệnh trong nội bộ thủ tục ... }
END ;
```

Ví dụ 7.1: Tìm số lớn nhất trong 3 trị số nguyên

PROGRAM Largest ; (\* Xác định số lớn nhất trong 3 trị số nguyên được nhập vào \*)

```
VAR a, b, c : integer ;
```

```
yn : char ;
```

```
PROCEDURE maximum ;
```

```
VAR max : integer ;
```

```
BEGIN
```

```
IF a > b THEN max := a ELSE max := b ;
```

```
IF c > max THEN max := c ;
```

```
Writeln (' Số lớn nhất là' , max) ;
```

```
END ;
```

```
BEGIN (* Đoạn chương trình chính *)
```

```
yn := 'Y' ;
```

```
WHILE (upcase(yn) = 'Y') DO
```

```
BEGIN
```

```
Writeln (' Nhập 3 số nguyên : ') ;
```

```
Readln (a, b, c) ;
```

```
maximum ; (* --- Lờ gọi thủ tục maximum --- *)
```

```
Write (' Tiếp tục nhập 3 số mới không (y/n) ? ') ;
```

Readln (yn) ;

END ;

END.

Chú ý:

Trong chương trình trên, thủ tục maximum được khai báo trước khi nó được truy xuất, các biến a, b, c được gọi nhập vào ở chương trình chính và biến max được định nghĩa bên trong thủ tục. Điều này cho ta thấy, không phải lúc nào cũng cần thiết khai báo biến ngay đầu chương trình chính.

### b. Cấu trúc của thủ tục có tham số

**PROCEDURE < Tên thủ tục > (<danh sách tham số hình thức : kiểu biến>);**

**{ Các khai báo hằng, biến, kiểu cục bộ ... }**

**BEGIN**

**{ ... các lệnh trong nội bộ thủ tục ... }**

**END ;**

Khi viết một thủ tục, nếu có các tham số cần thiết, ta phải khai báo nó (kiểu, số lượng, tính chất, ...). Các tham số này gọi là tham số hình thức (formal parameters).

Một thủ tục có thể có 1 hoặc nhiều tham số hình thức. Khi các tham số hình thức có cùng một kiểu thì ta viết chúng cách nhau bởi dấu phẩy (.). Trường hợp các kiểu của chúng khác nhau hoặc giữa khai báo tham số truyền bằng tham biến và truyền bằng tham trị (sẽ học ở phần sau ) thì ta phải viết cách nhau bằng dấu chấm phẩy (;).

Ví dụ 7.2: Tính giai thừa của một số

PROGRAM Tinh\_Giai\_thua ;

VAR

n : integer ; gt : real ; {các biến chung}

PROCEDURE giaiithua (m : integer) ;

VAR i : integer ; {i là biến riêng}

BEGIN

gt := 1 ;

FOR i := 1 TO m DO gt := gt \* i ;

END ;

BEGIN (\* Thân chương trình chính \*)

Write('Nhập số nguyên n (0 <= n < 33) = '); Readln (n) ;

```
If n>=0 then
 Begin
 giaithua (n) ;
 Writeln ('Giai thừa của , n, là : ' , gt: 10 : 0) ;
 End
Else Writeln(' Không tính giai thừa của một số âm! ') ;
Readln;
```

END.

Trong chương trình trên m là các tham số hình thức của thủ tục giaithua.

Khi gọi thủ tục giaithua (n) thì tham số thực n được truyền tương ứng cho tham số hình thức m.

Ví dụ 7.3: Giải phương trình  $ax^2 + bx + c = 0$ , theo dạng chương trình con lồng nhau:

```
PROGRAM Giai_PTB2;
VAR hsa, hsb, hsc:real; {các biến toàn cục}
PROCEDURE Ptb2(a,b,c:real); {a, b, c là các tham số hình thức của Ptb2}
 Var delta:real; {biến cục bộ}
 PROCEDURE Ptb1(a1,b1:real); {a,b là các tham số hình thức của Ptb1}
 Begin
 if a1=0 then
 if b1=0 then
 writeln('Phương trình vô số nghiệm')
 else
 writeln('Phương trình vô nghiệm')
 else
 writeln('Phương trình có nghiệm =',-b1/a1:8:2);
 End; {kết thúc thủ tục Ptb1}
 Begin {bắt đầu thủ tục Ptb2}
 (3) if a=0 then ptb1(b,c) {b, c là các tham số thực cho Ptb1}
 (4) else
```



```
begin
 delta:=sqr(b)-4*a*c;
 if delta>0 then
 begin
 writeln('Nghiem x1= ',(-b+sqr(delta))/(2*a):8:2);
 writeln('Nghiem x2= ',(-b-sqr(delta))/(2*a):8:2);
 end
 else
 if delta=0 then
 writeln('Nghiem kép x1=x2= ',-b/(2*a):8:2)
 else
 writeln('delta <0 => Phương trình vô nghiệm');
 end;
 End; {kết thúc thủ tục Ptb2}
```

Begin {chương trình chính}

- (1) write('Nhập các hệ số a, b, c = ');readln(hsa, hsb, hsc);
  - (2) Ptb2(hsa,hsb,hsc); {hsa, hsb, hsc là các tham số thực cho Ptb2}
  - (5) readln;
- End. {kết thúc chương trình}

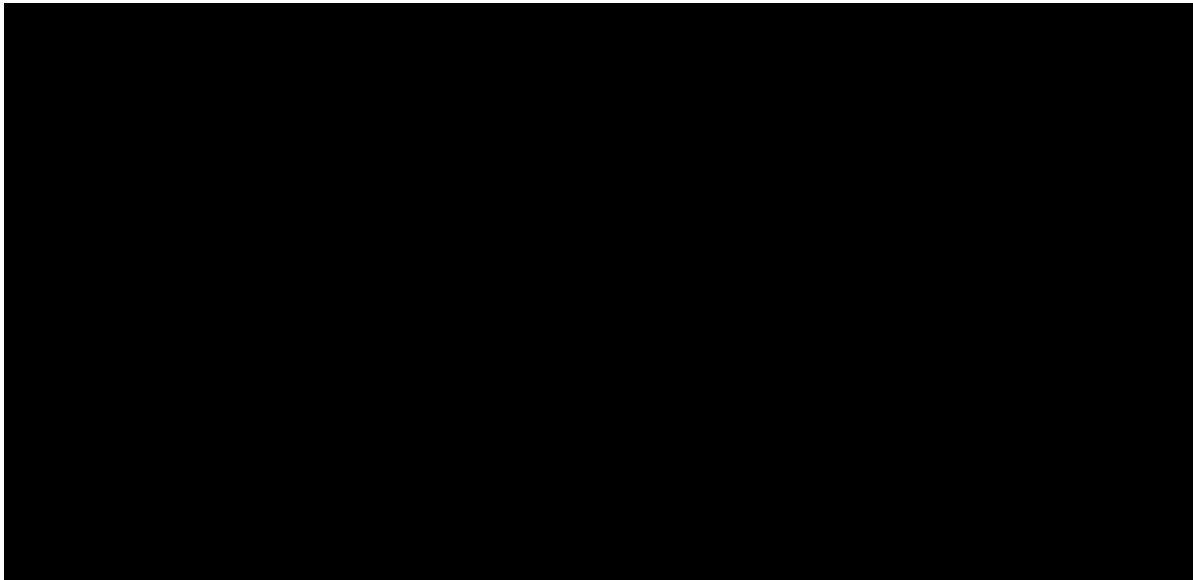
Ở ví dụ trên, thì thủ tục Ptb2 và thủ tục Ptb1 được gọi là thủ tục lồng nhau.

Ở dòng (4), ta thấy hsa, hsb, hsc lại được hiểu là các tham số thực, chúng truyền giá trị biến cho các tham số hình thức a, b, c tương ứng trong thủ tục Ptb2.

Nếu ta lại xét đến thủ tục con của thủ tục Ptb2 là Ptb1 thì các tham số a, b, c này (chính xác là b và c) lại là tham số thực đối với Ptb1, với b và c được truyền tương ứng cho các tham số hình thức a, b của thủ tục Ptb1.

Như vậy ta nhận thấy rằng, vấn đề xác định được đâu là biến toàn cục, đâu là biến cục bộ, đâu là tham số thực và đâu là tham số hình thức (tham số biến và tham số trị) là ứng bước nào mà chương trình đang thực hiện? Đây là phần then chốt để nắm được cách vận hành và kết quả của chương trình xử lý.

Sơ đồ minh họa cách vận hành và quản lý biến của chương trình:



## 2. Hàm (Function) :

Hàm là một chương trình con cho ta 1 giá trị kiểu vô hướng. Hàm tương tự như thủ tục nhưng trả về một giá trị thông qua tên hàm và lời gọi hàm tham gia trong biểu thức.

Cấu trúc một hàm tự đặt gồm:

**FUNCTION <Tên hàm> (<Tham số hình thức : kiểu biến>) : <Kiểu kết quả> ;**

**{ các khai báo hằng, biến, kiểu cục bộ... }**

**BEGIN**

**{ ... các khai báo trong nội bộ hàm ... }**

**END ;**

Trong đó:

- Tên hàm là tên tự đặt cần tuân thủ theo nguyên tắc đặt tên trong Pascal.
- Kiểu kết quả là một kiểu vô hướng, biểu diễn kết quả giá trị của hàm.

- Một hàm có thể có 1 hay nhiều tham số hình thức, khi có nhiều tham số hình thức cùng một kiểu giá trị thì ta có thể viết chúng cách nhau bằng dấu phẩy (.). Trường hợp các tham số hình thức khác kiểu thì ta viết chúng cách nhau bằng dấu chấm phẩy (;).

- Trong hàm có thể sử dụng các hằng, kiểu, biến đã được khai báo trong chương trình chính nhưng ta có thể khai báo thêm các hằng, kiểu, biến dùng riêng trong nội bộ hàm. Chú ý là phải có một biến trung gian có cùng kiểu kết quả của hàm để lưu kết quả của hàm trong quá trình tính toán để cuối cùng ta có 1 lệnh gán giá trị của biến trung gian cho tên hàm.

Ví dụ 7.4:      **FUNCTION TINH (x, y : integer ; z : real ) : real ;**

Đây là một hàm số có tên là TINH với 3 tham số hình thức x, y, z. Kiểu của x và y là kiểu số nguyên integer còn kiểu của z là kiểu số thực real. Hàm TINH sẽ cho kết quả kiểu số thực real.

Ví dụ 7.5:      Bài toán tính giai thừa (factorials)

```
PROGRAM giaithua ;

VAR x : integer ;

FUNCTION factorial (n : integer) : integer ;

VAR heso, tichso : integer ;

 BEGIN

 tichso := 1 ;

 IF n <= 1 THEN factorial := 1

 ELSE BEGIN

 FOR heso := 2 TO n DO

 tichso := tichso * heso ;

 factorial := tichso;

 END ;

 END ;

BEGIN

 Write (' Nhập vào một số nguyên dương x = '); Readln (x) ;

 Writeln (' Với x = , x , thì giai thừa sẽ là : x ! = ' , factorial(x))

 Readln;

END.
```

Ghi chú :

Khi khai báo kiểu dữ liệu cho các tham số hình thức trong thủ tục và hàm, ta cần phải chú ý điểm sau:

Nếu kiểu dữ liệu của các tham số hình thức là các kiểu dữ liệu có cấu trúc (kiểu array, string, kiểu record,... ) thì việc khai báo kiểu dữ liệu cho các tham số hình thức nên được khai báo theo cách gián tiếp, tức là phải thông qua từ khóa TYPE.

Ví dụ 7.6:     Procedure Xuat1(hoten : string[25]);  
  
                  Procedure Xuat2(mang: array[1..10] of integer);

Hai chương trình con Xuat1 và Xuat2 đều bị lỗi ở phần khai báo kiểu dữ liệu cho hai tham số hình thức là hoten và mang.

Để khắc phục lỗi này, ta sẽ khai báo gián tiếp một kiểu dữ liệu str25 và M10 thông qua từ khóa TYPE như sau:

TYPE

Str25=string[25]; {Str25 là một kiểu chuỗi có độ dài 25}

M10=Array[1..10] of integer; {M10 là một kiểu dữ liệu mảng có 10 phần tử nguyên}

Tiếp đến, dùng 2 kiểu dữ liệu mới định nghĩa Str25 và M10 để định kiểu cho các tham số hình thức hoten và mang như sau:

Procedure Xuat1(hoten : Str25);

Procedure Xuat2(mang: M10);

**III. TRUYỀN THAM SỐ CHO CHƯƠNG TRÌNH CON**

Khi truyền tham số trong Pascal, đòi hỏi phải có sự tương ứng về tên của kiểu dữ liệu của các tham số hình thức và tham số thực. Một số định nghĩa và qui tắc về truyền tham số trong Pascal:

- Những tham số hình thức nằm sau từ khóa VAR gọi là tham số biến (variable parameter). Với tham số biến, các tham số thực bắt buộc phải là biến chứ không được là giá trị. Khi giá trị của tham số biến thay đổi thì nó sẽ làm thay đổi giá trị của tham số thực tương ứng và khi ra khỏi chương trình con đó, tham số thực vẫn giữ giá trị đã được thay đổi đó.

- Những tham số hình thức không đứng sau từ khóa VAR gọi là tham số trị (value parameter), khi đó các tham số thực có thể là một biến, một biểu thức, một hằng, hoặc một giá trị số. Các tham số trị nhận giá trị từ tham số thực khi truyền như là giá trị ban đầu, khi giá trị của tham số trị thay đổi thì nó sẽ không làm thay đổi giá trị của tham số thực, nghĩa là giá trị của tham số thực sau khi thoát khỏi chương trình con vẫn luôn bằng với giá trị của tham số thực trước khi truyền đến chương trình con đó. Do vậy một tham trị không bao giờ là kết quả tính toán của chương trình con.

Một vài thí dụ về tham số biến:

Ví dụ 7.7: Viết chương trình tính lập phương.

PROGRAM Parameter1;

VAR num: integer; {num là biến toàn cục}

PROCEDURE LapPhuong(var a:integer); {a là một tham số biến}

Begin

a:=a\*a\*a;

End;

Begin

write('Nhập số cần tính lập phương num = ');

readln(num);

LapPhuong(num); {tham số thực num được truyền cho tham số biến a}

writeln('Lập phương của số vừa nhập =', num);

readln;

End.

Ví dụ 7.8:

```
PROGRAM parameter2;
VAR a, b : integer ; {biến toàn cục }
PROCEDURE thamso (x : integer ; VAR y : integer) ;
BEGIN
 { x: là tham số trị , còn y là tham số biến}
 x := x + 1 ;
 y := y + 1 ;
 Writeln ('Trong procedure thamso, ... ');
 Writeln (' Hai số của bạn là a = , x : 3, và b = , y : 3) ;
END ;
BEGIN
 Write (' Nhập vào 2 trị số nguyên a, b : ') ; Readln (a, b) ;
 Writeln (' Ban đầu, Bạn đã nhập vào a =', a : 3, 'và b =' , b : 3) ;
 thamso (a, b) ; {tham số thực a truyền cho tham số trị x
 tham số thực b truyền cho tham số biến y }
 Writeln (' Ngoài procedure thamso, ... ');
 Writeln (' Hiện nay, số a là , a : 3, và b là , b : 3 ') ;
 Writeln (' Ta thấy, a không đổi và b thay đổi ! ') ;
 Readln;
END.
```

#### IV. TÍNH ĐỆ QUI CỦA CHƯƠNG TRÌNH CON

Một chương trình con mà trong quá trình thiết lập, nó sẽ gọi chính bản thân nó thì chương trình con có tính đệ qui (recursion).

Ví dụ 7.9: Bài toán tính giai thừa (factorials) theo cách đệ qui. Bài toán này có phần chương trình chính giống như đã có ở ví dụ trước:

```
PROGRAM Giaiithua ; (*Tính giai thừa của số n theo phương pháp đệ qui *)
VAR x : integer ;
```

```
FUNCTION factorial (n : integer) : longint ;
```

```
 BEGIN
```

```
 IF n <= 1 THEN factorial := 1 {điều kiện neo}
```

```
 ELSE factorial := n * factorial (n -1);
```

```
 END ;
```

```
BEGIN
```

```
 Write (' Nhập vào một số nguyên dương x = ');
```

```
 Readln (x) ;
```

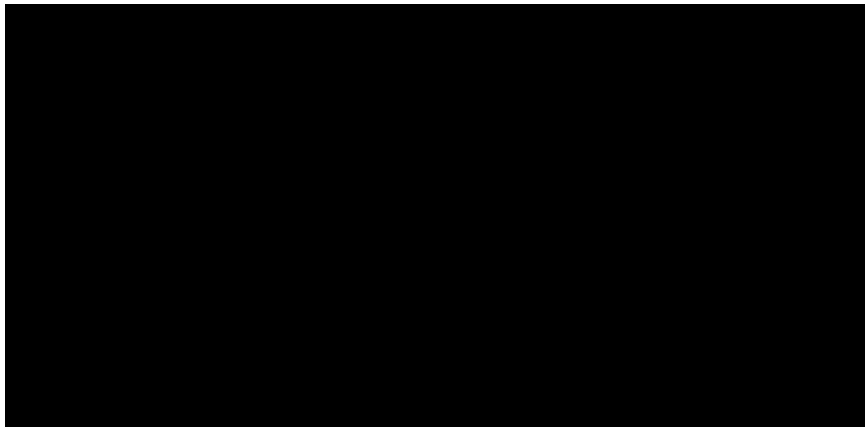
```
 Writeln ;
```

```
 Writeln (' Kết quả ',x,'! = ', factorial(x));
```

```
 Readln;
```

```
END.
```

Giả sử ta nhập  $x = 4$ , thì  $4! = \text{factorial}(n)$ , với  $n = 4$ , ta có sơ đồ minh họa như sau:



Chú ý:

- Ưu điểm của thuật toán đệ qui là ngắn gọn. Nó có khả năng định nghĩa một tập hợp rất lớn các đối tượng bằng một số các câu lệnh hữu hạn. Thuật toán đệ qui có vẻ thích hợp cho các bài toán mà tự thân cấu trúc dữ liệu của nó đã được định nghĩa theo lối đệ qui.

- Có một số thuật toán đệ qui sử dụng cho các bài toán đơn giản có thể được thay thế bằng một thuật toán khác không tự gọi chúng, sự thay thế đó được gọi là khử *đệ qui*.

- Trong một số bài toán ta có thể giải theo 2 cách: thuật toán lặp (xem chương trước) và thuật toán đệ qui. Thông thường, cách giải theo thuật toán lặp (WHILE .. DO) thì tốt hơn so với thuật toán đệ qui vì đệ qui đòi hỏi thêm bộ nhớ và thời gian. Khi đó các thanh ghi được sử dụng cho lưu trữ và khi quay trở về phải khôi phục lại trạng thái cũ trong mỗi lần gọi đến chương trình con. Mức độ phức tạp có thể gia tăng khi trong chương trình con theo thuật toán đệ qui có chứa những chương trình con khác. Vì vậy, khi dùng đệ qui ta cần thận trọng, nhất là thuật toán này thường không cho ta thấy rõ trực tiếp toàn bộ quá trình giải các bước. Nói chung, chỉ khi nào không thể dùng thuật toán *lặp ta mới nên sử dụng thuật toán đệ qui*.

## V. ĐƠN VỊ CHƯƠNG TRÌNH (UNIT)

### 1. Khái niệm

Lập trình một bài toán lớn rất phức tạp và vất vả nếu nó phải sử dụng nhiều thuật toán lập đi lập lại. Việc tạo ra nhiều mẫu chương trình con nhằm giảm nhẹ công việc của một lập trình viên (programmer). Tuy nhiên, mỗi chương trình con chỉ có ứng dụng được trong chính chương trình chứa nó mà thôi. Để khỏi mất thời gian để viết lại chúng, người ta biến mỗi chương trình con thành các module độc lập, được biên dịch sẵn và lưu trữ trên đĩa như một thư viện. Khi cần ta cứ việc gọi các module này ra mà không cần phải viết lại chúng. Mỗi module như vậy được gọi là một đơn vị chương trình, hay gọi tắt là UNIT.

Khái niệm Unit đã được vào sử dụng từ chương trình Pascal version 4.0 trở đi. Có hai loại Unit là các Unit chuẩn do Turbo Pascal tạo sẵn và các Unit tự tạo do người lập trình tự viết ra.

### 2. Các Unit chuẩn

#### a. Một số Unit chuẩn trong Turbo Pascal 5.5 trở đi

- \* Unit SYSTEM : gồm các hằng, biến, kiểu, hàm, thủ tục trong version 3.0
- \* Unit CRT : gồm các hằng, biến, kiểu, hàm, thủ tục liên quan đến chế độ Text của version 5.5
- \* Unit PRINTER : gồm các hằng, biến, kiểu, hàm, thủ tục liên quan đến chế độ in ấn.
- \* Unit GRAPH : gồm các hằng, biến, kiểu, hàm, thủ tục liên quan đến chế độ đồ thị của version 5.5
- \* Unit TURBO3 : gồm các hằng, biến, kiểu, hàm, thủ tục liên quan đến chế độ Text của version 3.0
- \* Unit GRAPH3 : gồm các hằng, biến, kiểu, hàm, thủ tục liên quan đến chế độ đồ thị của version 3.0
- \* Unit DOS : gồm các hằng, biến, kiểu, hàm, thủ tục liên quan đến hệ điều hành MS-DOS
- \* Unit OVERLAY : gồm các hằng, biến, kiểu, hàm, thủ tục liên quan đến việc truy xuất đĩa phủ lấp khi chạy chương trình.

Các Unit trên được lưu trữ trong tập tin TURBO.TPL của Turbo Pascal.

Chúng ta có thể sử dụng chương trình TPUMOVER.EXE để lấy ra hoặc đưa vào một hay nhiều Unit nhằm tiết kiệm bộ nhớ hay tăng cường tiện ích sử dụng.

#### b. Cách gọi Unit

Muốn sử dụng UNIT thì trong đầu chương trình ta phải khai báo Unit theo cú pháp sau:

```
USES <Tên Unit> [{ , <Tên Unit> }] ;
```

Ví dụ 7.10                    USES CRT, GRAPH ;

Đặc biệt, chỉ riêng Unit SYSTEM thì không cần phải khai báo.

**c. Một số thủ tục và hàm trong Unit CRT**

- \* ClrScr : thủ tục xóa màn hình
- \* ClrEol : thủ tục xóa ký tự bên phải con trỏ màn hình, sau khi xóa con trỏ vẫn ở tại chỗ
- \* InsLine : thủ tục xen vào một hàng ở vị trí con trỏ màn hình
- \* DelLine : thủ tục xóa bỏ một hàng ở vị trí con trỏ màn hình
- \* GotoXY(XPos, Ypos): đưa con trỏ màn hình về vị trí có tọa độ Xpos và Ypos. X có giá trị từ 1 - 80, và Y có giá trị từ 1 - 25
- \* Delay(time): tạo thời gian trễ tính theo milisecond. Time là một số nguyên dùng để làm chậm chương trình cho ta kịp quan sát dữ liệu
- \* Sound(F) : thủ tục tạo ra âm thanh với tần số F (hz). F là số nguyên
- \* NoSound : thủ tục tắt âm thanh
- \* LowVideo và NormVideo: thủ tục màn hình, khi gọi LowVideo thì mọi ký tự viết ra màn hình có độ sáng yếu dần đi cho đến khi nhận thủ tục NormVideo mới về độ sáng bình thường.
- \* TextBackGround (color): thủ tục chọn màu nền, color cho ở bảng (từ 1 -7)
- \* KeyPressed: hàm cho giá trị kiểu kết quả Boolean, cho giá trị là True nếu có một phím được bấm.
- \* TextColor (color): thủ tục chọn màu chữ, color lấy ở bảng

Các hằng số màu của CRT unit

| Hằng số color | Màu hiển thị | Giá trị |
|---------------|--------------|---------|
| Black         | Đen          | 0       |
| Blue          | Xanh da trời | 1       |
| Green         | Xanh lá cây  | 2       |
| Cyan          | Xanh lơ      | 3       |
| Red           | Đỏ           | 4       |
| Magenta       | Tím          | 5       |
| Brown         | Nâu          | 6       |
| LightGray     | Xám nhạt     | 7       |
| DarkGray      | Xám đậm      | 8       |



|              |                   |    |
|--------------|-------------------|----|
| LightBlue    | Xanh da trời nhạt | 9  |
| LightGreen   | Xanh lá cây nhạt  | 10 |
| LightCyan    | Xanh lơ nhạt      | 11 |
| LightRed     | Đỏ nhạt           | 12 |
| LightMagenta | Tím nhạt          | 13 |
| Yellow       | Vàng              | 14 |
| White        | Trắng             | 15 |

Để tạo ra màu nhấp nháy, ta cộng thêm hằng số Blink của CRT vào bất kỳ màu chữ mà ta chọn. Ta không thể làm nhấp nháy màu nền.

Ví dụ để trình bày chữ vàng trên nền xanh, ta viết:

```
TextColor(Yellow) ; TextBackground(Blue) ;
```

Muốn có một dòng chữ đỏ nhấp nháy thì ra lệnh:

```
TextColor(Red + Blink) ;
```

\* **ReadKey** : hàm có kiểu kết quả là Char, khi nhấn các phím chức năng trên bàn phím thì có kết quả là 1 ký tự mã ASCII. Nếu ký tự đầu tiên do ReadKey trả về bằng ASCII 0 thì ký tự kế theo sẽ chỉ định phím như ở bảng dưới. Các phím Home, phím mũi tên, ... luôn tạo nên một ký tự đi theo ASCII 0. Các phím chức năng từ F1 đến F10 sinh ra một trong 4 ký tự tùy theo ta dùng với tổ hợp phím Alt, Ctrl hay Shift hay dùng một mình.

### Các phím chức năng đặc biệt

| Tên phím | Bình thường | Alt     | Ctrl    | Shift  |
|----------|-------------|---------|---------|--------|
| F1       | : (59)      | h (104) | ^ (94)  | T (84) |
| F2       | < (60)      | i (105) | - (95)  | U (85) |
| F3       | = (61)      | j (106) | . (96)  | V (86) |
| F4       | > (62)      | k (107) | a (97)  | W (87) |
| F5       | ? (63)      | l (108) | b (98)  | X (88) |
| F6       | @ (64)      | m (109) | c (99)  | Y (89) |
| F7       | A (65)      | n (110) | d (100) | Z (90) |
| F8       | B (66)      | o (111) | e (101) | [ (91) |
| F9       | C (67)      | p (112) | f (102) | \ (92) |
| F10      | D (68)      | q (113) | g (103) | ] (93) |

|        |        |  |  |  |
|--------|--------|--|--|--|
| Home   | G (71) |  |  |  |
| .↑     | H (72) |  |  |  |
| PageUp | I (73) |  |  |  |
| .←     | K (75) |  |  |  |
| →.     | M (77) |  |  |  |
| End    | O (79) |  |  |  |
| ↓.     | P (80) |  |  |  |
| PageDn | Q (81) |  |  |  |
| Ins    | R (82) |  |  |  |
| Del    | S (83) |  |  |  |

## VI. UNIT TỰ TẠO

### 1. Một số bước để tạo ra Unit

Để tạo ra một Unit của mình cần đi qua các bước sau:

**Bước 1 :** *Tạo ra một file Unit có phần mở rộng là .PAS với bố cục sau:*

UNIT <Tên Unit> ; (\* Chú ý : Tên Unit phải trùng với tên File \*)

INTERFACE (\* Chú ý : Phần giao diện với bên ngoài, không có dấu ; ở đây \*)

[Uses <danh sách các unit>]; {Khai báo các unit dùng trong chương trình }

[Khai báo các hằng, kiểu, biến dùng chung] ;

[Khai báo các thủ tục, hàm (tên, danh sách tham số của thủ tục và hàm) ;

IMPLEMENTATION(\* Cài đặt các hàm, thủ tục của Unit, không có dấu ; ở đây \*)

[Các khai báo kiểu, hằng, biến cục bộ ];

[Nội dung cài đặt các thủ tục, hàm của unit ];

[BEGIN] (\* Phần khởi tạo : Initialization Part \*)

[Các lệnh khởi tạo ];

END. (\* Dù có BEGIN để khởi tạo hay không, ở đây vẫn có END. \*)

**Bước 2 :** *Dịch file này lên đĩa theo trình tự sau:*

i). Gõ Alt - C để vào Menu COMPILE

- ii). Đi đến mục Destination và nhấn Enter để chương trình tự động đổi **Memory thành Disk**
- iii). Gõ C (hoặc F9) để biên dịch chương trình tạo nên một file .TPU
- iv). Khi dịch xong gõ một phím bất kỳ. Sau đó ta có thể lập lại bước a và b để chuyển Destination từ Disk sang Memory.

## 2. Ví dụ

Ví dụ 7.11: Tạo một UNIT tính Cộng, Trừ, Nhân, Chia cho học sinh tiểu học.

Tên file Unit là TTIEUHOOC.PAS với nội dung sau:

```
UNIT TTieuHoc ; {Phần đầu : Chương trình Toán Tiểu học }
INTERFACE {Phần giao diện}

 PROCEDURE Cong (Var So1, So2, So3 : Real) ;
 PROCEDURE Tru (Var So1, So2, So3 : Real) ;
 PROCEDURE Nhan (Var So1, So2, So3 : Real) ;
 PROCEDURE Chia (Var So1, So2, So3 : Real) ;

IMPLEMENTATION {Phần cài đặt }

 PROCEDURE Cong ;

 BEGIN

 IF So1 + So2 = So3 THEN Writeln ('Giỏi lắm ! Em đã làm đúng! ')
 ELSE Writeln (' Rất tiếc, em đã làm sai ! ');

 END;

 PROCEDURE Tru ;

 BEGIN

 IF So1 - So2 = So3 THEN Writeln (' Giỏi lắm ! Em đã làm đúng!')
 ELSE Writeln (' Rất tiếc, em đã làm sai ! ');

 END;

 PROCEDURE Nhan ;

 BEGIN

 IF So1 * So2 = So3 THEN Writeln ('Giỏi lắm ! Em đã làm đúng!')
 ELSE Writeln (' Rất tiếc, em đã làm sai ! ');
```

```
END;

PROCEDURE Chia ;

BEGIN

 IF So2 = 0 THEN Writeln ('Số chia phải khác 0')

 ELSE

 IF So1 / So2 = So3 THEN Writeln ('Giỏi lắm! Em đã làm đúng! ')

 ELSE Writeln (' Rất tiếc, em đã làm sai ! ');

 END;

END. {Chấm dứt UNIT }
```

Sau khi gõ chương trình Unit trên, đổi Compile Destination thành Disk, biên dịch và tạo tập tin TTIEUHOOC.TPU trên đĩa.

Chương trình Pascal cho bài toán Cộng, trừ, Nhân, Chia dùng Unit TTIEUHOOC:

```
PROGRAM Toan_Tieu_Hoc ;

USES CRT, TTieuHoc ;

VAR

 chon : Integer ;

 So1, So2, So3 : Real ;

PROCEDURE Menu (Var chon : integer) ;

 BEGIN

 ClrScr ;

 Writeln (' == TOÁN TIỂU HỌC == ');

 Writeln (' = 0. Chấm dứt = ');

 Writeln (' = 1. Toán cộng = ');

 Writeln (' = 2. Toán trừ = ');

 Writeln (' = 3. Toán nhân = ');

 Writeln (' = 4. Toán chia = ');

 Writeln (' ===== ');

 Write (' Bạn chọn số mấy ? ');
```

Readln (chon);

END ;

PROCEDURE nhapso (Var So1, So2, So3 : real);

BEGIN

Write (' Nhập vào số thứ 1 : ' ); Readln(So1) ;

Write (' Nhập vào số thứ 2 : ' ); Readln(So2) ;

Write (' Kết quả là : ' ); Readln (So3) ;

END ;

{=====Chương Trình Chính =====}

BEGIN

CLRSCR;

REPEAT

Menu (chon) ;

CASE chon OF

1 : BEGIN

Writeln ;

Writeln (' == Toán cộng == ' );

Nhapso(So1, So2, So3) ;

Cong(So1, So2, So3) ;

END;

2 : BEGIN

Writeln ;

Writeln (' == Toán trừ == ' );

Nhapso(So1, So2, So3) ;

Tru(So1, So2, So3) ;

END;

3 : BEGIN

Writeln ;

```
 Writeln (' == Toán nhân == ');
 Nhapso(So1, So2, So3);
 Nhan(So1, So2, So3);
 END;
4 : BEGIN
 Writeln ;
 Writeln (' == Toán chia == ');
 Nhapso(So1, So2, So3);
 Chia(So1, So2, So3);
 END;
END; { case }
Writeln (' Gõ bất kỳ phím nào để tiếp tục ... ');
Readln;
UNTIL chon = 0;
END. { Ngung làm toán }
```

---

## BÀI ĐỌC THÊM

### TOP - DOWN STRUCTURED PROGRAMMING

hay

### LẬP TRÌNH THEO CẤU TRÚC TRÊN - XUỐNG

--- oOo ---

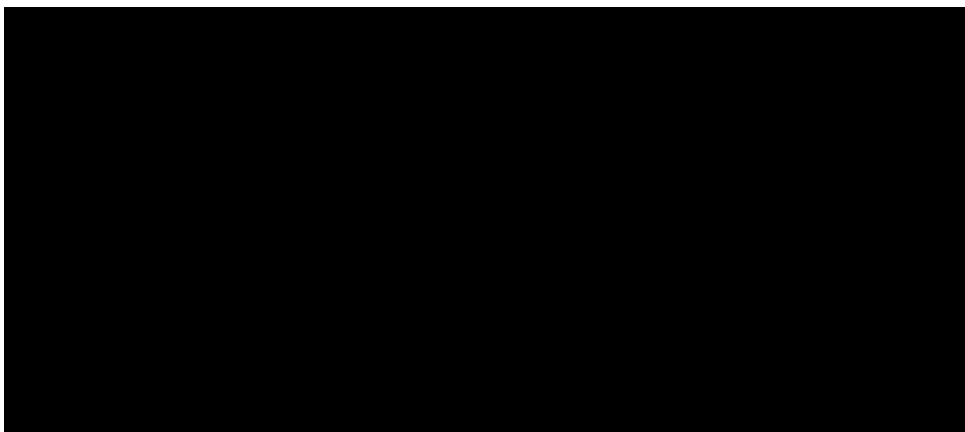
Nếu các bạn là một người mới bắt đầu khởi sự thực hành lập trình một bài toán nào đó, các bạn sẽ thường tự hỏi: Ta phải bắt đầu bằng việc gì đây? Đây là một câu hỏi không phải ai cũng trả lời chung được. Tuy nhiên, dựa vào kinh nghiệm thu thập được của những người lập trình tài tử và của những lập trình viên chuyên nghiệp, tác giả Francis Scheid, trong tác phẩm Computer and Programming của mình, đã cho một số lời khuyên sau :

1. Ôn lại những kinh nghiệm đã qua để xem coi vấn đề của bạn có chút gì tương tự đến các vấn đề mà bạn đã từng chạm trán trước đây không;
2. Trước tiên, thử làm một phiên bản đơn giản. Nếu có thể, đưa ngay vào một số trường hợp đặc biệt mà bạn có, nhằm tạo chương trình bạn có một vẻ gì sâu sắc.

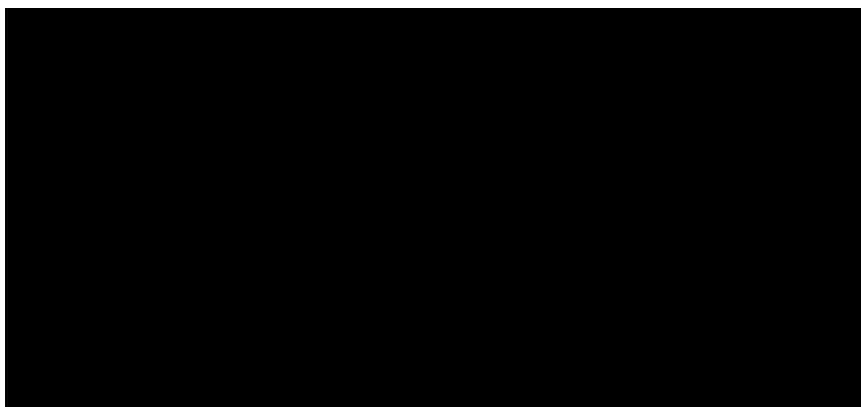
3. Chia bài toán ra thành những bài nhỏ, rồi tiếp tục chẻ những bài nhỏ này thành những phần nhỏ hơn, nếu được, chẻ tiếp những phần nhỏ này thành những mảnh nhỏ hơn nữa, sau đó giải quyết từng phần hay mảnh nhỏ này.

Mỗi thuật toán có thể thể hiện bằng lưu đồ (flow chart). Lưu đồ chính là bản đồ lộ trình của thuật toán. Không hẳn tất cả những chuyên viên máy tính phải thực hiện lưu đồ trước khi lập trình nhưng nhờ có lưu đồ mà công việc của bạn trở nên rõ ràng và mang tính logic hơn. Bạn sẽ không cảm thấy bối rối khi cần phải trình bày tiến trình giải toán của bạn cho người khác hiểu. Bạn có thể mất một ít thời gian cho lưu đồ nhưng nó có giá trị hơn cả ngàn từ nếu phải cất nghĩa thuật toán.

Chúng ta hãy tưởng tượng bài toán to lớn của chúng ta như một cây cổ thụ nhiều cành lá rậm rạp. Ta muốn đốn cây này về nhà làm củi chụm và dĩ nhiên, ta không thể nào chặt ngang gốc cây mà vác về nhà (thí dụ này không có ý khuyến khích phá hoại môi trường đâu nhé ! ). Vậy tại sao ta không tỉa từng cành nhỏ rồi dần dần thanh toán luôn cả cây ? Giải quyết vấn đề của chúng ta cũng vậy. Bạn cứ xem bài toán của chúng ta như một gốc cây lộn ngược đầu. Chia nhỏ bài toán ra thành những vấn đề nhỏ hơn, rồi nhỏ hơn nữa nếu nó còn phức tạp, như minh họa ở hình sau đây:



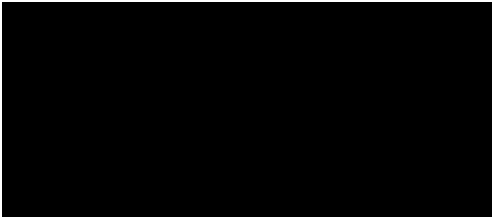
Trong hình vẽ trên, bài toán được phân thành 4 vấn đề nhỏ hơn là A, B, C và D. Vấn đề B và D có thể giải quyết được ngay. Riêng vấn đề A và C thì lại tiếp tục chia nhỏ hơn nữa để thành những mảnh nhỏ có thể giải quyết được. Ở đây các nhánh cây không dài ngắn như nhau, dễ hiểu bởi vì mức độ phức tạp của mỗi vấn đề không thể như nhau và tùy theo thuật toán cần giải quyết mà ta phân nhỏ nó ra. Bài toán của chúng ta sẽ đi từ vấn đề trừu tượng đến cụ thể. Cách giải quyết như vậy giống như hệ thống phân quyền trong tổ chức chính phủ:



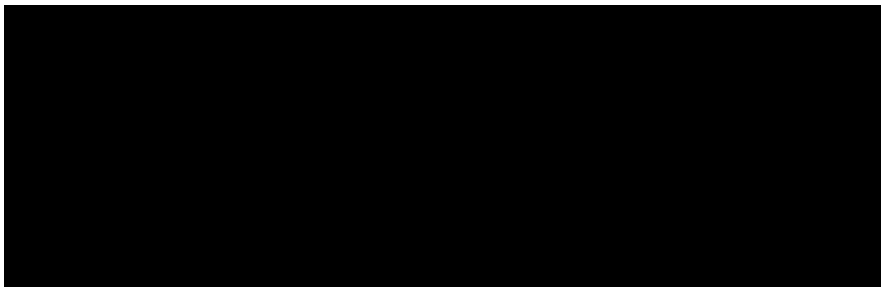
Lập trình cấu trúc là một trường phái lập trình xuất hiện vào thập niên 1970 và đã nhanh chóng được nhiều người hưởng ứng. Điểm cơ bản trong lập trình cấu trúc là tổ chức chương trình thành một hệ phân cấp (hierarchy) và phải điều khiển sao cho các mối tương tác giữa các thành phần trong hệ là tối thiểu. Đây chính là ý tưởng của một phép tinh chế từng bước (stepwise refinement) hay phương pháp chia để trị trong giải bài toán theo cách top-down. Một khi ta đã thực hiện việc phân tích top-

down xong, những mảnh bài toán chi tiết nhất sẽ được giải theo cách của 1 trong 3 thành phần thuật toán sau :

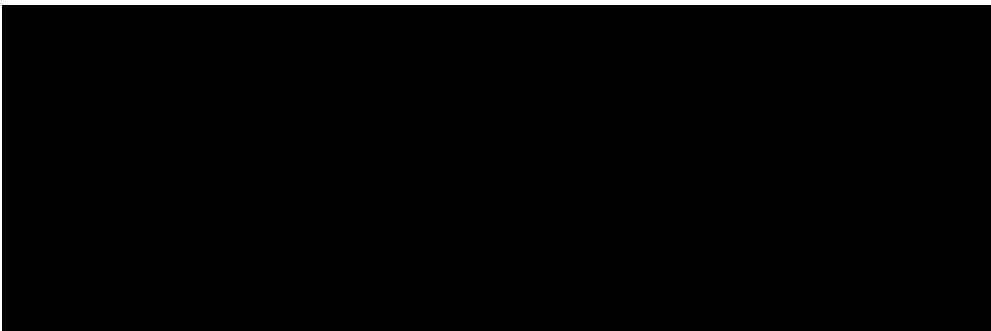
\* Dòng tuần tự (Sequential Flow) : Trong thuật giải này, các bước giải được thể hiện ở trong một luồng lệnh tuần tự như hình vẽ sau:



\* Dòng điều kiện (Conditional Flow): Trong thực tế ở nhiều bài toán máy tính, ta sẽ đi đến việc chọn lựa một trong hai điều kiện. Mỗi điều kiện (Đúng hoặc Sai) sẽ dẫn đến một quyết định khác nhau. Minh họa ở hình sau:



\* Dòng lặp (Repetitive Flow) : Trong nhiều trường hợp, ta cần thực hiện nhiều lần liên tiếp một hay nhiều thao tác cho đến khi một điều kiện được thỏa.



Cha đẻ của ngôn ngữ lập trình Pascal là N.Wirth đã phát triển phương pháp tinh chế từng bước, xem nó như một phương pháp khoa học cần thiết cho việc phân tích vấn đề và lập trình.

Khởi đầu, chương trình phải được thể hiện khái quát vấn đề, nêu bậc sự phân tích tổng thể bài toán. Ở từng bước kế tiếp sau đó, sẽ có các giải pháp giải quyết vấn đề một cách chi tiết hơn, mỗi giải pháp như vậy là một sự đặc tả (specification) công việc. Như vậy, từng bước một, ta dần dần tinh chế bài toán. Sự tinh chế này phải hướng đến các thuật toán của ngôn ngữ lập trình. Nếu các bài toán nhỏ trở nên đơn giản thì ta thay nó bằng các câu lệnh. Nếu nó tỏ ra còn phức tạp thì ta xem đó như một thủ tục và tiếp tục tìm cách tinh chế nó.

Trong quá trình tinh chế, cần thiết đưa ra những cách biểu diễn dữ liệu đi song song với cách chi tiết hoá việc giải quyết bài toán. Đây là một phương pháp khoa học nhưng cũng mang một phần tính nghệ thuật thể hiện sự nhạy bén trong tư duy của người lập trình.



## CHƯƠNG 8

### MỘT SỐ CẤU TRÚC DỮ LIỆU CƠ SỞ

- I. KIỂU LIỆT KÊ, KIỂU MIỀN CON.
  - 1. Kiểu vô hướng liệt kê.
  - 2. Kiểu miền con.
- II. KIỂU MẢNG, KIỂU CHUỖI.
  - 1. Dữ liệu kiểu mảng.
  - 2. Dữ liệu kiểu chuỗi.
- III. KIỂU TẬP HỢP (SET).
  - 1. Định nghĩa và khai báo.
  - 2. Mô tả một tập hợp.
  - 3. Các phép toán trên tập hợp.
  - 4. Viết và đọc dữ liệu kiểu tập hợp.
- IV. KIỂU BẢN GHI (RECORD).
  - 1. Định nghĩa và khai báo.
  - 2. Truy xuất một Record.
  - 3. Các Record lồng nhau.
  - 4. Câu lệnh WITH.

#### I. KIỂU LIỆT KÊ, KIỂU MIỀN CON

##### 1. Kiểu vô hướng liệt kê (enumerated scalar type)

Chương trước chúng ta đã đi qua các kiểu dữ liệu đơn giản là các dữ liệu kiểu dữ liệu vô hướng chuẩn (Standard Scalar-type Data) như Integer, Real, Char, Boolean. Các kiểu này đã được định nghĩa sẵn trong mọi chương trình cài đặt trong máy. Ngôn ngữ Pascal cho phép người lập trình có thể tự đặt ra các kiểu vô hướng mới bằng cách tự liệt kê các giá trị của kiểu vô hướng mới và phải khai báo định nghĩa kiểu. Danh sách các giá trị này được đặt trong ngoặc đơn ( ) và được mô tả bằng một tên kiểu (như phần mô tả kiểu TYPE). Kiểu vô hướng theo cách này gọi là kiểu vô hướng liệt kê (Enumerated Scalar Type).

##### a. Cách khai báo

Có 2 cách khai báo một biến kiểu liệt kê:

+ *Khai báo gián tiếp: Định nghĩa kiểu (dựa vào từ khóa type) trước khi khai biến (var)*

**TYPE**

<tên kiểu liệt kê> = (<danh sách giá trị kiểu liệt kê>);

**VAR**

<danh sách biến> : <tên kiểu liệt kê>;

Ví dụ 8.1:

TYPE

Days = (Sun, Mon, Tue, Wed, Thu, Fri, Sat) ;

Colors =(Red, Yellow, Green, White, Blue, Black) ;

Subjects = (Mathematics, Physics, Chemistry, Biology) ;

VAR

Ngày : Days ;

MauVe : Colors ;

MonThi, Kiemtra : Subjects ;

+ *Khai báo trực tiếp: Kiểu sau biến được định nghĩa trực tiếp.*

VAR

< danh sách biến > : (< danh sách giá trị kiểu liệt kê >) ;

Ví dụ 8.2:

VAR

Ngày : (Sun, Mon, Tue, Wed, Thu, Fri, Sat) ;

MauVe : (Red, Yellow, Green, White) ;

Ta có thể gán cho biến các giá trị của kiểu tương ứng:

Ngày := Mon ;

MauVe := Red ;

Biến theo định nghĩa của kiểu nào chỉ nhận giá trị của kiểu đó mà thôi.

Theo khai báo như ví dụ 8.2. ở trên, ta không thể có MauVe := Mon ;

Kiểu vô hướng liệt kê là một kiểu đếm được.

Theo định nghĩa kiểu vô hướng liệt kê, thứ tự danh sách giá trị liệt kê được ngầm đánh số tăng tuyến tính bắt đầu từ số 0 trở đi theo thứ tự từ trái sang phải. Như vậy, ở ví dụ trên: Sun < Mon < Tue < Wed .... và Red < Yellow < Green ...

**b. Một số hàm chuẩn áp dụng cho kiểu vô hướng**

\* Hàm thứ tự ORD (X)

Hàm này cho ta thứ tự của giá trị x trong kiểu vô hướng đếm được. Hàm ORD thực chất là hàm biến đổi một giá trị kiểu vô hướng đếm được sang giá trị kiểu số nguyên.

Theo ví dụ trên:

ORD (Sun) = 0      là Đúng vì Sun có thứ tự là 0

ORD (Mon) = 1      là Đúng vì Mon có thứ tự là 1

ORD (Green) = 3    là Sai vì Green có thứ tự là 2

ORD (n) = n        trong đó n là một giá trị kiểu Longint

**\* Hàm PRED (X)**

Hàm này cho giá trị đứng trước x trong định nghĩa kiểu của x.

Theo ví dụ trên :

PRED (Mon) = Sun

PRED (Green) = Yellow

PRED (n) = n - 1

**\* Hàm SUCC (X)**

Hàm này cho giá trị đứng sau x trong định nghĩa kiểu của x.

Theo ví dụ trên:

SUCC (Mon) = Tue

SUCC (Green) = White

SUCC (n) = n + 1

**\* Hàm chuyển một số nguyên thành một giá trị vô hướng**

Tên hàm này chính là tên kiểu vô hướng mà ta đã khai báo trước.

Theo ví dụ trên:

Days(2) = Tue

Colors(3) = White

LONGINT (n) = n

**c. Viết ra và đọc vào kiểu liệt kê**

Viết và đọc theo kiểu liệt kê thì khác với kiểu vô hướng chuẩn.

**\* Viết ra kiểu liệt kê**

Thủ tục Write và Writeln chỉ chấp nhận đưa ra các giá trị thuộc kiểu vô hướng chuẩn (Real, Integer, Byte, Char, Boolean) mà không chấp nhận viết ra một giá trị kiểu vô hướng liệt kê, ví dụ cách viết sau là không đúng:

Writeln(Color(4))

Writeln(Red)

Writeln(Days)

mà chỉ có thể chấp nhận nếu viết:

```
Writeln (Char(78)) vì Char(78) = N là giá trị vô hướng chuẩn.
```

Để viết ra một giá trị của biến vô hướng liệt kê, ta có thể áp dụng thủ thuật sau:

```
IF MauVe = Red THEN Writeln('Red') ;
```

\* Đọc vào kiểu liệt kê

Thủ tục Read và Readln cũng chỉ chấp nhận đọc vào một giá trị kiểu vô hướng chuẩn mà không chấp nhận đọc trực tiếp các giá trị kiểu vô hướng liệt kê, ví dụ không thể đọc Readln(Days). Để đọc vào một giá trị kiểu liệt kê ta có thể dùng phương pháp sau: đọc số thứ tự của giá trị biến vô hướng rồi biến đổi kiểu dữ liệu thêm:

Ví dụ 8.3:

```
TYPE Days = (Sun, Mon, Tue, Wed, Thu, Fri, Sat) ;

VAR i : Integer ;

BEGIN

 Write('Nhập số từ 0 . .6 tương ứng cho ngày:'); Readln(i) ;

 Case Days(i) of

 Sun: writeln('Ngày Chủ nhật');

 Mon: writeln('Ngày thứ hai');

 Tue: writeln('Ngày thứ ba');

 Wed: writeln('Ngày thứ tư');

 Thu: writeln('Ngày thứ năm');

 Fri: writeln('Ngày thứ sáu');

 Sat: writeln('Ngày thứ bảy');

 Else writeln('Nhập sai');

 end;

 Readln;

END.
```

Mục 2 ở phần II ở phía sau, sẽ giới thiệu chuỗi String, ta có thể dùng thủ thuật sau để đọc kiểu liệt kê:

Ví dụ 8.4: Readln(St) ;

```
IF St = 'Mon' THEN Ngay := Mon ;
```

## 2. Kiểu miền con (Sub-range type)

### a. Khái niệm

Khi khai báo một số trường hợp, ví dụ Tuổi của người hoặc Điểm thi học sinh, nếu ta viết:

VAR

TuoiTho : Integer ; {Integer có miền xác định -32 768 .. 32 767}

Hay Diem : Real ; {Real có miền xác định 2.9 E-39 .. 1.7 E38}

Nếu viết như vậy sẽ tốn ô nhớ vì Integer có kích thước 2 bytes hoặc Real có kích thước đến 6 bytes. Làm như vậy sẽ không cần thiết vì Tuổi con người chỉ biến thiên trong khoảng từ 0 đến 200 là lớn nhất và điểm thi học sinh thì chỉ trong khoảng từ 0 đến 10 chẳng hạn.

Trong Pascal cho phép ta xác định một biến lấy giá trị trong một khoảng nào đó được giới hạn ở một hằng cận dưới (first data item) và một hằng cận trên (last data item). Hai giá trị này phải cùng một kiểu vô hướng đếm được và hằng cận trên có giá trị lớn hơn hằng cận dưới. Khai báo như vậy gọi là khai báo kiểu miền con (Sub-range type) và biến của nó chỉ chiếm 1 byte trong ô nhớ mà thôi. Trong lúc chạy chương trình, ta có thể kiểm tra giá trị của biến không được vượt ra khỏi giới hạn của khoảng con.

### b. Cách khai báo

Miền con là một tập hợp con của một kiểu đếm được. Có 2 cách khai báo:

+ Khai báo gián tiếp:

TYPE

<Tên kiểu miền con> = <hằng cận dưới> .. <hằng cận trên> ;

VAR

<danh sách biến> : < Tên kiểu miền con> ;

Ví dụ 8.5:

TYPE

TuoiTho = 0 .. 200 ;

VAR Tho : TuoiTho ;

+ Khai báo trực tiếp:

VAR

<danh sách biến> : <hằng cận dưới> .. <hằng cận trên> ;

Ví dụ 8.6:

VAR Tuoi : 0 .. 200 ;

## II. KIỂU MẢNG, KIỂU CHUỖI

Pascal có 4 kiểu cấu trúc dữ liệu là kiểu mảng (ARRAY), tập hợp (SET), bản ghi (RECORD) và tập tin (FILE). Sau đây ta lần lượt tìm hiểu từng kiểu cấu trúc.

### 1. Dữ liệu kiểu mảng (Array-Type Data)

Một mảng dữ liệu là một tập hợp số hữu hạn phần tử có giống như các biến, có cùng kiểu, gọi là kiểu cơ bản.

Mảng được tổ chức theo một trật tự xác định. Số phần tử của mảng được khai báo ngay từ khi định nghĩa ra mảng.

#### a. Mảng một chiều (One-Dimensional Array)

Mảng một chiều có thể được hiểu như một danh sách các phần tử (theo cột), có cùng kiểu. Mỗi phần tử của mảng được xác định được truy nhập trực tiếp thông qua tên mảng cùng với chỉ dẫn truy nhập được để giữa hai ngoặc vuông [ ].

Ví dụ 8.7:

List là một mảng 1 chiều có n phần tử. Các phần tử của List có thể mang các tên List[1], List[2], List[3], ..., List[n], và có thể minh họa như hình sau:



Hình 8.1: Minh họa mảng một chiều

+ *Khai báo gián tiếp:*

**TYPE**

**<Kiểu mảng> = ARRAY [Kiểu chỉ số ] OF <Kiểu phần tử > ;**

**VAR**

**<Danh sách biến> : Kiểu mảng ;**

+ *Khai báo trực tiếp :*

**VAR**

**< Danh sách biến > : ARRAY [ Kiểu chỉ số] OF < Kiểu phần tử > ;**

\* Chú ý: Kiểu chỉ số phải là kiểu rời rạc (đếm được).

Ví dụ 8.8:

**TYPE**

**KM1 = ARRAY [1.. 100] OF INTEGER ;**

**KM2 = ARRAY [1 .. 20 ] OF CHAR ;**

DAY = (Sun, Mon, Tue, Wed, Thu, Fri, Sat) ;

VAR

TUOI : KM1 ;

TEN : KM2 ;

NGAY : ARRAY [DAY] OF BOOLEAN ;

Ý nghĩa:

- KM1 là kiểu mảng gồm 100 phần tử được đánh số từ 1 đến 100 thông qua kiểu chỉ dẫn là một miền con các số nguyên từ 1 .. 100. TUOI là biến có kiểu là KM1.
- KM2 là kiểu mảng gồm 20 phần tử đánh số từ 1 .. 20 có kiểu là các ký tự. Biến TEN có kiểu là KM2.
- NGAY là một biến mảng gồm 7 phần tử kiểu Boolean được đánh dấu qua kiểu chỉ dẫn là tên của 7 ngày trong tuần.

Chú ý:

Khi khai báo mảng, kiểu chỉ dẫn chỉ có thể là:

- Kiểu miền con của các loại dữ liệu vô hướng đếm được như ký tự, số nguyên
- Kiểu liệt kê do người viết định nghĩa (như NGAY trong tuần)
- Kiểu Boolean

Kiểu chỉ dẫn không thể là kiểu không đếm được như REAL

Viết như sau là SAI : X1 : ARRAY [Real] OF Integer ;

Ta cũng không thể khai báo như: X2 : ARRAY [Integer] OF Integer ;

Mặc dầu Integer là kiểu vô hướng đếm được do giới hạn của vùng nhớ dành cho dữ liệu, số lượng phần tử của 1 mảng cũng bị hạn chế tùy theo kích thước của kiểu dữ liệu của các phần tử, ta nên dùng kiểu miền con để khai báo số phần tử của mảng.

**+ Truy xuất các phần tử của mảng:**

Mỗi phần tử của mảng được truy xuất thông qua Tên Biến Mảng cùng với chỉ số của mảng trong dấu ngoặc vuông [ ]. Ví dụ tên biến mảng là A, khi viết A[7], ta hiểu nó là phần tử thứ 7 của mảng A.

Ví dụ 8.9: Lập trình giải một bài toán tính trung bình một dãy số x[i] :

$x[1], x[2], x[3], \dots, x[n]$

sau đó tiếp tục tính độ lệch (deviation) của từng phần tử so với trị trung bình, theo công thức:

$$\text{độ\_lệch} = x[i] - \text{trung\_bình}$$

Giả sử dãy số của chúng ta có giới hạn  $n = 100$  phần tử trở lại,  $n$  là một biến số để khai báo số phần tử muốn tính [redacted]. Sau đó ta lần lượt nhập tính giá trị của phần tử kiểu số thực (real) từ phần tử thứ 1 đến phần tử thứ  $n$ . Trong chương trình sẽ tạo ra một mảng 1 chiều  $x$  với  $n$  các phần tử. Tính trung bình của  $n$  phần tử và độ lệch. In kết quả ra màn hình.

PROGRAM Average\_deviations ;

{ Nhập  $n$  số phần tử kiểu số thực, tính trị trung bình của chúng, sau đó tính tiếp độ lệch của từng phần tử số so với trị trung bình }

VAR

n, count : integer ;

sum, average, deviation : real ;

x : ARRAY [1 .. 100] OF real ;

BEGIN

(\* Nhập số phần tử và tính trung bình\*)

Write (' Nhập bao nhiêu số  $n$  để tính trung bình ? ');

Readln (n) ;

Writeln ;

sum := 0 ;

FOR count := 1 TO n DO

BEGIN

Write (' i = ', count : 3, ' x = ');

Readln (x [count] ) ;

sum := sum + x[count];

END ;

average := sum/n ;

Writeln (' Trung bình của dãy số là = , average ');

Writeln ;

(\* Tính độ lệch so với trị trung bình \*)

FOR count := 1 TO n DO

BEGIN

deviation := x[count] - average ;



Write ( ' i = ', count : 3, ' x = ', x[count] ) ;

Writeln ( ' Độ lệch d = ', deviation );

END ;

Readln;

END.

Giả sử, ta nhập vào 5 số hạng (các số có gạch dưới là phần của người nhập):

x[1] = 3.0    x[2] = -2.0    x[3] = 12.0    x[4] = 4.4    x[5] = 3.5

Khi chạy chương trình (nhấn Ctrl + F9), trên màn hình ta sẽ thấy :

Nhập bao nhiêu số n để tính trung bình ? 5

i = 1                    x = 3.0

i = 2                    x = -2.0

i = 3                    x = 12.0

i = 4                    x = 4.4

i = 5                    x = 3.5

Trung bình của dãy số là = 4. 1800000E+00

i = 1            x = 3. 0000000E+00    Độ lệch d = - 1. 1800000E+00

i = 2            x = -2. 0000000E+00    Độ lệch d = - 6. 1800000E+00

i = 3            x = 1. 2000000E+00    Độ lệch d = 7. 8200000E+00

i = 4            x = 4. 4000000E+00    Độ lệch d = 2. 2000000E - 01

i = 5            x = 3. 5000000E+00    Độ lệch d = - 6. 8000000E - 01

Ta có thể định khoảng chứa kết quả và phần lẻ thập phân, dùng lệnh : m : n

Ví dụ 8.10: Sắp xếp một dãy số theo thứ tự từ nhỏ đến lớn

Tiến trình của bài toán:

- Giả sử chuỗi số của ta có n phần tử XXXXXXXXXX. Lần lượt cho chương trình đọc giá trị của các phần tử nhập được.
- Một thủ tục (Procedure) sẽ làm công việc sắp xếp như sau : đầu tiên đưa phần tử thứ nhất so sánh với các phần tử tiếp theo, nếu nó lớn hơn phần tử so sánh thì đem đổi chỗ giá trị của hai phần tử với nhau. Sau đó tiếp tục đem phần tử thứ 2 so sánh các phần tử tiếp theo theo trình tự như vậy, ... và cứ như thế cho đến phần tử thứ n - 1.
- In kết quả ra màn hình

Chương trình Pascal như sau:

```
PROGRAM Reorder ;

 (* Sắp xếp một mảng các phần tử số thực từ nhỏ đến lớn*)

VAR n, i, loc: 1 .. 100 ;

 x : ARRAY [1 .. 100] OF real ;

 temp : real ;

PROCEDURE interchange ;

 (* Đổi chỗ các phần tử mảng từ nhỏ đến lớn*)

BEGIN

 FOR loc := 1 TO n-1 DO

 FOR i := loc + 1 TO n DO

 IF x[i] < x [loc] THEN

 BEGIN

 temp := x[loc] ;

 x[loc] := x[i] ;

 x[i] := temp ;

 END ;

 END ;

 END ;

 END ;

BEGIN

 Write (' Có bao nhiêu phần tử số ? ') ; Readln (n) ;

 FOR i := 1 TO n DO

 BEGIN

 Write (' x[' , i : 3, '] = ? ') ;

 Readln(x[i]) ;

 END ;

 interchange ;

 Writeln ;

 Writeln (' Số liệu đã sắp xếp : ') ;
```

```
Writeln ;
FOR i := 1 TO n DO
 Writeln ('x[', i : 3, '] = ', x[i] : 4 : 1);
Readln;
END.
```

Khi chạy chương trình, giả sử ta có 5 số liệu như phần nhập :

(các số có gạch dưới là phần nhập từ bàn phím)

Có bao nhiêu phần tử số ? 5

$$x[ 1] = ? \underline{4.7}$$

$$x[ 2] = ? \underline{-2.3}$$

$$x[ 3] = ? \underline{12.9}$$

$$x[ 4] = ? \underline{8.8}$$

$$x[ 5] = ? \underline{6.0}$$

*Kết quả là :*

Số liệu đã sắp xếp :

$$x[ 1] = ? - 2.3$$

$$x[ 2] = ? 4.7$$

$$x[ 3] = ? 6.0$$

$$x[ 4] = ? 8.8$$

$$x[ 5] = ? 12.9$$

### **b. Mảng nhiều chiều (Multi-Dimensional Array)**

Trong một số bài toán thực tế, người ta sử dụng các mảng nhiều hơn 1 chiều, gọi là mảng nhiều chiều.

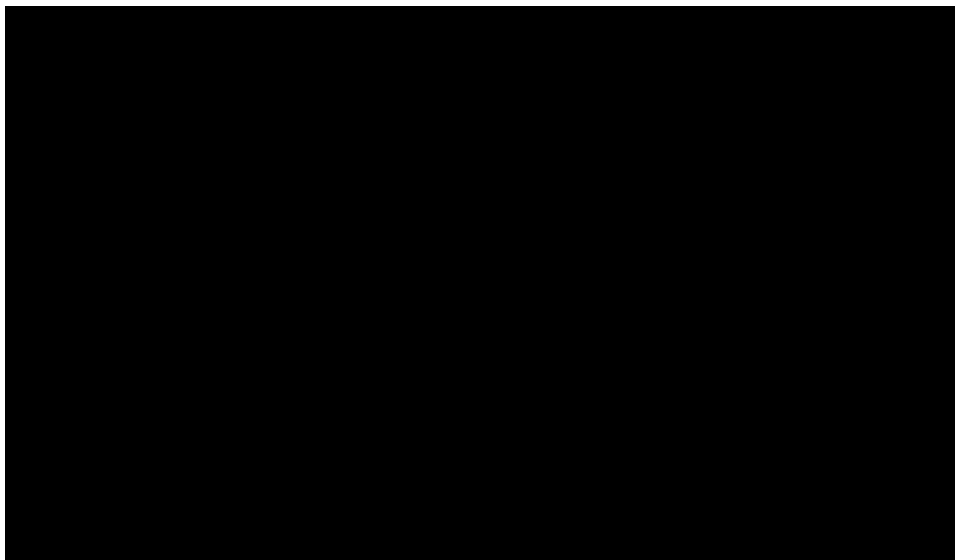
Ví dụ 8.11: Phòng Đào tạo quản lý điểm của sinh viên. Trong khoá 22 chẳng hạn, người ta tạo ra một mảng 2 chiều: ví dụ một chiều là số thứ tự của sinh viên, chiều còn lại là các môn học (dạng kiểu vô hướng liệt kê), ta có thể hình dung dạng của mảng ghi điểm (tên mảng là ghi\_diem) như sau:



*Lưu ý: Thực tế, danh sách tên sinh viên lưu lại trong máy tính thường được ghi bằng cách gán mã số sinh viên (coding) cho mỗi sinh viên ngay từ năm đầu vào học.*

Với ví dụ trên, muốn nhập điểm một sinh viên nào đó ta phải khai báo 2 tham số là số thứ tự sinh viên và môn học.

Tương tự, cũng với các khoá kế tiếp theo học những môn như vậy, ta sẽ tạo ra mảng nhiều chiều như hình vẽ minh họa sau:



Trong trường hợp này, muốn biết điểm một sinh viên nào đó ta phải khai báo 3 tham số: Khoá học, số thứ tự sinh viên và môn học, chẳng hạn:

ghi\_diem[K22,0001,AV] nhập điểm 10,...

Khai báo cũng có 2 cách như đối với mảng 1 chiều:

+ *Khai báo gián tiếp:*

**TYPE**

<Kiểu mảng> = ARRAY [Kiểu\_chỉ\_số\_1, ..., Kiểu\_chỉ\_số\_n] OF <Kiểu phần tử>;

**VAR**

<Danh sách biến>:<Kiểu mảng>;

Ví dụ 8.12:

TYPE matrix = ARRAY [1 .. 20, 1 .. 30] OF integer ;

VAR A:matrix;

Lệnh trên khai báo một kiểu tên matrix. Đây là một mảng 2 chiều, chiều thứ nhất có các chỉ số từ 1 đến 20, chiều thứ hai có các chỉ số từ 1 đến 30, tổng cộng ta có (20 x 30) phần tử số nguyên. Và ta có một biến A là biến có kiểu matrix.

Ví dụ trên cũng có thể được khai báo tương đương với:

```
TYPE matrix = ARRAY [1 .. 20] OF ARRAY [1 .. 30] OF integer ;
```

```
VAR A:matrix;
```

+ **Khai báo gián tiếp:**

**VAR**

**<Danh sách biến>: ARRAY [Kiểu\_chỉ\_số\_1, ..., Kiểu\_chỉ\_số\_n] OF <Kiểu phần tử>;**

Khai báo một biến A có 5 dòng và 10 cột kiểu phần tử là Integer như sau:

```
VAR A : ARRAY [1 .. 5, 1 .. 10] OF integer ;
```

+ **Truy xuất các phần tử mảng:**

Tương tự như cách truy xuất phần tử của mảng 1 chiều, mảng nhiều chiều cũng được truy xuất thông qua tên biến mảng kết hợp với các chỉ số của nó được đặt trong cặp dấu ngoặc vuông.

Mảng 2 chiều là một ma trận, như ví dụ trên ta có một ma trận 5 dòng và 10 cột. Các phần tử của ma trận A được ký hiệu là  $a[i,j]$  với  $i$  là vị trí cột và  $j$  là dòng. Khi viết  $a[2, 7]$  thì hiểu đây là phần tử ở dòng 2 và cột 7.

Trong Pascal, ta có thể viết  $a[i,j]$  thành  $a[i] [j]$  với ý nghĩa hoàn toàn như nhau.

Chú ý: Trên nguyên tắc, ta có thể khai báo một mảng có đến 255 chiều. Tuy vậy, một điều cần lưu ý là kích thước bộ nhớ của máy tính có hạn nên thường chỉ khai báo mảng từ 1 đến 3 chiều. Khai biến quá nhiều thì phải cần máy lớn hơn.

Chẳng hạn khi báo 1 mảng [1.. 10] các phần tử số nguyên đã lấy 10 bytes bộ nhớ

- Mảng 2 chiều  $10 \times 10 = 100$  bytes bộ nhớ.
- Mảng 3 chiều  $10 \times 10 \times 10 = 1\,000$  bytes bộ nhớ
- Mảng 4 chiều  $10 \times 10 \times 10 \times 10 = 10\,000$  bytes bộ nhớ
- Mảng 5 chiều  $10 \times 10 \times 10 \times 10 \times 10 = 100\,000$  bytes bộ nhớ
- v.v...

Ví dụ 8.13:

Viết một chương trình Pascal để đọc một bảng các số thực được nhập vào máy tính dưới dạng một mảng 2 chiều. Tính tổng các giá trị số theo hàng và theo cột. Kết quả được in ra màn hình theo vị trí hàng và cột tương ứng.

Trước tiên, ta bắt đầu bằng định nghĩa các biến:

table = mảng 2 chiều chứa số thực dưới dạng bảng gồm các số nhập và kết quả

nrows = một biến số nguyên chỉ số hàng

ncols = một biến số nguyên chỉ số cột

row = một số đếm nguyên chỉ số hàng

col = một số đếm nguyên chỉ số cột

Để đơn giản, chúng ta giả sử rằng kích thước số liệu nhập vào bảng tính không vượt quá 10 hàng và 10 cột. Ta sẽ thêm vào 1 hàng cộng phía dưới và 1 cột cộng bên phải vào bảng để ghi kết quả tính cộng các phần tử hàng và cột tương ứng. Như vậy, mảng 2 chiều của chúng ta sẽ trở thành mảng sẽ được in ra có số hàng là (nrows + 1) và số cột là (ncols + 1). Do vậy, ta phải khai báo biến table là 1 mảng 2 chiều số nguyên có tối đa 11 cột và 11 hàng.

Để dễ theo dõi chương trình, ta thực hiện cấu trúc module khi viết chương trình bằng cách tiến hành làm các thủ tục procedure cho đọc số liệu, tính tổng các phần tử theo hàng, tính tổng các phần tử theo cột và in ra màn hình bảng kết quả. Các thủ tục này sẽ có tên tương ứng là readinput, rowsums, columsums và writeoutput.

Thuật toán logic yêu cầu cho mỗi thủ tục là cách khai báo thẳng trước (straightforward), chú ý rằng trong mỗi thủ tục ta có một vòng lặp đôi (double loop). Ví dụ, để đọc số liệu ở bảng gốc, ta sẽ phải làm một vòng lặp đôi sau:

```
FOR row := 1 TO nrows DO
```

```
 BEGIN
```

```
 FOR col := 1 TO ncols DO readln(table[row, col]);
```

```
 Writeln;
```

```
 END ;
```

Câu lệnh Writeln để báo chương trình nhảy tới dòng kế.

Tương tự, vòng lặp sau được viết để tính tổng các phần tử theo hàng:

```
FOR row := 1 TO nrows DO
```

```
 BEGIN
```

```
 table [row, ncols + 1] := 0 ;
```

```
 FOR col := 1 TO ncols DO
```

```
 table [row, ncols + 1] := table [row, ncols + 1] + table [row, col];
```

```
 END ;
```

Tương tự, cấu trúc vòng lặp đôi cũng được dùng để tính tổng các phần tử cột và in ra bảng kết quả cuối cùng.

Sau đây là chương trình Pascal của bài toán trên:

```
PROGRAM Tongbang ;

 {đọc một bảng số, tính tổng từng cột và hàng của cá bảng}

VAR

 row, col : 1 .. 11 ;

 nrows, ncols : 1 .. 10 ;

 table : ARRAY [1 .. 11, 1 .. 11] OF real ;

PROCEDURE Rowsums ; {cộng các phần tử theo cột bên trong mỗi hàng }

BEGIN

 FOR row := 1 TO nrows DO

 BEGIN

 table [row,ncols+1] := 0 ;

 FOR col := 1 TO ncols DO

 table[row, ncols+1] := table[row, ncols+1] + table[row,col];

 END ;

 END ;

 END ;

PROCEDURE Columnsums ; {cộng các phần tử theo hàng bên trong từng cột }

BEGIN

 FOR col := 1 TO ncols DO

 BEGIN

 table [nrows+1, col] := 0 ;

 FOR row := 1 TO nrows DO

 table[nrows+1,col] := table[nrows+1,col] + table[row,col];

 END ;

 END ;

 END ;

PROCEDURE Readinput ; {đọc các phần tử của bảng }

BEGIN

 Write(' Nhập số hàng (1 .. 10) ? ');Readln(nrows) ;
```

```
Write(' Nhập số cột (1 .. 10) ? ');Readln(ncols) ;

FOR row := 1 TO nrows DO

 BEGIN

 Writeln (' Nhập số liệu hàng số , row :2') ;

 FOR col := 1 TO ncols DO readln(table [row, col]) ;

 END ;

END ;

PROCEDURE Writeoutput ; { In ra bảng số liệu và kết quả tính tổng }

BEGIN

 Writeln('Bảng số liệu và kết quả tính tổng các phần tử theo hàng và cột ');

 Writeln('=====');

 Writeln;

 FOR row := 1 TO nrows + 1 DO

 BEGIN

 FOR col := 1 TO ncols+1 DO Write (table [row,col] : 6 : 1) ;

 Writeln;

 END ;

 END ;

BEGIN { Thân chương trình chính }

 Readinput ;

 Rowsums ;

 Columnsums ;

 Writeoutput;

END. { Chấm dứt chương trình }
```

Giả sử, ta có bảng số liệu sau :

|      |      |      |      |
|------|------|------|------|
| 2.5  | -6.3 | 14.7 | 4.0  |
| 10.8 | 12.4 | -8.2 | 5.5  |
| -7.2 | 3.1  | 17.7 | -9.1 |



Khi chạy chương trình, ta có (số có gạch dưới là số của người thử chương trình):

Nhập số hàng (1 .. 10) ? 3

Nhập số cột (1 .. 10) ? 4

Nhập số liệu hàng số 1

2.5   -6.3   14.7   4.0

Nhập số liệu hàng số 2

10.8   12.4   -8.2   5.5

Nhập số liệu hàng số 3

-7.2   3.1   17.7   -9.1

Chương trình sẽ tính tổng các giá trị ở hàng và cột, xong in ra màn hình kết quả:

Bảng số liệu và kết quả tính tổng các phần tử theo hàng và cột

2.5   -6.3   14.7   4.0   14.9

10.8   12.4   -8.2   5.5   20.5

-7.2   3.1   17.7   -9.1   4.5

6.1   9.2   24.2   0.4   0.0

Ta có thể kiểm tra kết quả ở các hàng và cột.

## 2. Dữ liệu kiểu chuỗi (String Type Data)

Một chuỗi dữ liệu là một loạt các ký tự được định nghĩa bằng từ khoá STRING theo sau là số ký tự cực đại có thể có của chuỗi ký tự. String là một kiểu cấu trúc được thêm vào trong Turbo Pascal.

### a. Khai báo

Chúng ta có thể khai báo kiểu chuỗi ký tự String gián tiếp hoặc trực tiếp. Khai báo gián tiếp là khai kiểu trước rồi sau đó mới khai báo biến. Cách khai báo trực tiếp là khai thẳng biến số. Chiều dài tối đa của chuỗi ký tự phải là một hằng nguyên và được đặt trong dấu ngoặc vuông [ ]. Trường hợp không khai báo thì chương trình sẽ lấy giá trị mặc nhiên là 255 ký tự XXXXXXXXXX

+ *Khai báo gián tiếp*

**TYPE**

<Tên kiểu String> = STRING [hằng nguyên] ;

**VAR**

<Tên biến> : <Tên kiểu String> ;

Ví dụ 8.14:

TYPE

TenSV = STRING [25] ; {định độ dài tối đa là 25}

Diachi = STRING; {mặc nhiên có độ dài tối đa là 255}

VAR

HT : TenSV ;

DC : Diachi ;

+ *Khai báo trực tiếp*

VAR

<Tên biến> : STRING [hằng nguyên] ;

Ví dụ 8.15:

VAR

HT : STRING [25] ;

DC : STRING;

Chuỗi ký tự sẽ chiếm số byte trong bộ nhớ bằng số ký tự lớn nhất đã khai báo trước cộng thêm 1 byte đầu tiên chứa số ký tự hiện có của chuỗi ký tự.

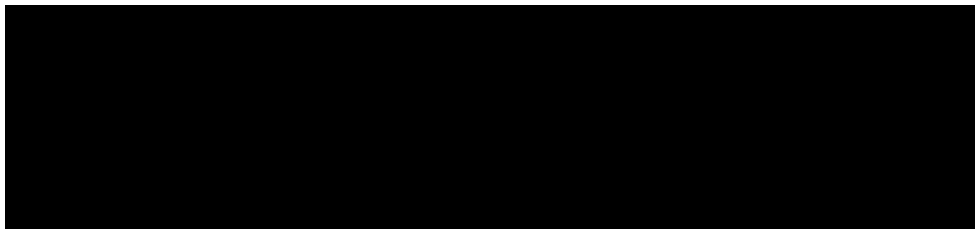
Ví dụ 8.16:

TYPE DH = STRING[10] ;

VAR CT : DH ;

và nếu ta gán CT := CAN THO;

thì CT sẽ được cấp phát 1 + 10 = 11 ô nhớ (byte) liên tục, với hình ảnh sau :



Chú ý:

- Độ dài của chuỗi ký tự CT là 7 ký tự mặc dầu độ dài lớn nhất cho phép là 10.
- Vì ta dùng 1 byte để chứa chiều dài nên string chỉ có tối đa là 255 ký tự.

## b. Các thao tác trên chuỗi

+ **Phép gán**

Giống như phép gán trong các kiểu vô hướng khác, phép gán chuỗi là lệnh gán một biến với một biểu thức ký tự để trong cặp dấu nháy đơn

Cú pháp:

**<Tên biến> := Biểu thức ký tự ;**

Ví dụ 8.17:

HT := Lê Văn Hai ;

DC := Số 12/4 đường Trần Hưng Đạo, TP. Cần thơ ;

### + Phép cộng

Phép cộng là thuật toán nối các chuỗi lại với nhau bằng dấu cộng (+).

Ví dụ trên nếu ghép HT + DC thì ta sẽ được:

Lê Văn Hai Số 12/4 đường Trần Hưng Đạo, TP. Cần thơ

Ghi chú: Không có phép trừ, nhân, chia trong chuỗi ký tự.

### + Các phép so sánh

Các so sánh gồm có bằng nhau =, lớn hơn >, lớn hơn hoặc bằng >=, khác nhau <>, nhỏ hơn <, nhỏ hơn hoặc bằng <=

Khi so sánh 2 chuỗi ký tự thì các ký tự được so sánh từng cặp một từ trái sang **phải theo giá trị của bảng mã ASCII. Có 2 khả năng xảy ra khi so sánh:**

- Nếu 2 chuỗi có độ dài khác nhau nhưng số ký tự giống nhau cho đến độ dài chuỗi ngắn nhất thì chuỗi ngắn nhỏ hơn chuỗi dài.

Ví dụ 8.18: 'Nation' < 'National'      "Lan" < 'Lang'

- Nếu 2 chuỗi có độ dài và nội dung giống nhau thì bằng nhau.

Ví dụ 8.19: 'Hello' = 'Hello'

Ghi chú: Chuỗi rỗng (null string, viết là "") là chuỗi không có chứa gì cả. Nó có giá trị nhỏ hơn mọi string khác rỗng.

Vì vậy: 'A' > ""      và chr(32) > ""

### + Câu lệnh Read và Readln

Hai câu lệnh này đối với chuỗi cũng tương tự như đối với các kiểu vô hướng khác, nhưng cần lưu ý:

- Lệnh Read và Readln chỉ cho phép đọc tối đa 127 ký tự một chuỗi nhập từ bàn phím mặc dù chiều dài tối đa của một chuỗi có thể đến 255 ký tự.

- Nếu ta đọc một lúc nhiều biến theo kiểu Read(biến1, biến2, ..., biếnN) ( hoặc Readln(biến1, biến2, ..., biếnN)) thì có thể bị nhầm lẫn khi ta nhập giá trị có độ dài vượt quá độ dài tối đa của biến1

thì phần vượt sẽ được gán cho biến2. Ngược lại, nếu ta nhập giá trị ít hơn độ dài của biến1 thì chương trình lại lấy các giá trị của biến2 gán thêm cho biến1 kể cả khoảng trống. Do vậy, cách tốt nhất là đối với biến kiểu String chỉ nên nhập mỗi lần 1 biến.

Ví dụ 8.20: Nên tránh viết kiểu Read(TenSV, Diachi); mà nên viết :

Read(TenSV) ;

Read(Diachi) ;

hoặc:

Readln(TenSV) ;

Readln(Diachi) ;

- Độ dài thực tế của chuỗi là độ dài thực tế khi ta đọc vào từ bàn phím mặc dầu trước đó ta có khai báo độ dài chuỗi. Nếu ta gõ Enter mà không gõ ký tự nào trước đó thì mặc nhiên chương trình hiểu đó là một chuỗi rỗng (null string hay st = "").

### + Câu lệnh Write và Writeln

Tương tự như trên nhưng cần một số lưu ý về cách viết:

- Nếu viết Write(st) hoặc Writeln(st) gọi là cách viết không qui cách thì mỗi ký tự sẽ chiếm 1 vị trí trên màn hình.
- Nếu viết Write(st : n) hoặc Writeln(st : n) gọi là cách viết theo qui cách, với n là số nguyên, thì màn hình sẽ dành n vị trí để viết chuỗi st theo lối canh trái nếu  $n > 0$  và ngược lại theo lối canh phải nếu  $n < 0$ .
- Một số chuỗi mà trong đó có dấu như là một chữ viết tắt, ví dụ như câu: Hes an Intal staff (Ông ta là một nhân viên quốc tế) thì nơi có dấu phải viết thành (đây là 2 dấu nháy đơn chứ không phải là 1 dấu nháy kép).

Ta viết:

Writeln ( ' He 's an Int''al staff ' );

### c. Các thủ tục và hàm chuẩn xử lý chuỗi ký tự

Chuỗi ký tự được dùng khá phổ biến trong lập trình nên Turbo Pascal đã đưa sẵn vào một số thủ tục và hàm chuẩn để xử lý chuỗi ký tự.

#### \* Thủ tục xóa DELETE (St, Pos, Num)

Ý nghĩa: Xóa khỏi chuỗi St một số ký tự là Num bắt đầu từ vị trí Pos tính từ trái sang.

Ví dụ 8.21: VAR st : string [20];

BEGIN

St := ' BÀ BA BÁN BÁNH BÒ ' ; Writeln (St) ;

DELETE (St, 10, 4); Writeln(St); Readln ;

END.

Khi chạy chương trình, ta sẽ thấy trên màn hình:

BÀ BA BÁN BÁNH BÒ

BÀ BA BÁN BÒ

\* **Thủ tục INSERT (Obj, St, Pos)**

Ý nghĩa: Chèn chuỗi Obj xen vào chuỗi St kể từ vị trí Pos tính từ bên trái.

```
Ví dụ 8.22: VAR st : string [25];

 BEGIN

 St := 'BÀ BA BÁN BÁNH BÒ' ; Writeln (St) ;

 INSERT ('BỤNG BỤ' , St, 6); Writeln(St); Readln ;

 END.
```

Khi chạy chương trình, ta sẽ thấy trên màn hình:

BÀ BA BÁN BÁNH BÒ

BÀ BA BỤNG BỤ BÁN BÁNH BÒ

\* **Thủ tục STR (S [: n[: m]], St)**

Ý nghĩa: Đổi giá trị số S thành chuỗi rồi gán cho St, Giá trị n:m nếu có sẽ là số vị trí và số chữ số thập phân của S.

```
Ví dụ 8.23: VAR S: real;

 St: string[10];

 BEGIN

 S:= 12345.6718;

 Writeln(S:5:2);

 Str(S:6:2:st);

 Readln;

 END.
```

Kết quả trên màn hình:

12345.67            {Đây là số }

12345.67            {Đây là chuỗi}

\* **Thủ tục VAL(St, S, Code)**

Ý nghĩa: Đổi chuỗi số St (biểu thị một số nguyên hoặc số thực) thành số (số nguyên hoặc số thực) và gán giá trị này cho S. Code là số nguyên dùng để phát hiện lỗi: nếu đổi đúng thì Code có giá trị = 0, nếu sai do St không biểu diễn đúng số nguyên hoặc số thực thì Code sẽ nhận giá trị bằng vị trí của ký tự sai trong chuỗi St.

```
Ví dụ 8.24: VAR St : String[10];

 SoX : real;

 maloi: integer;

 BEGIN

 St:= '123.456' ;

 VAL(St,SoX,maloi) ;

 Writeln('Số X = , SoX :5:2, và mã lỗi = , maloi) ;

 Readln;

 St:='123.XXX ';

 VAL(St,SoX,maloi);

 Writeln('St = 123.XXX không đổi thành số được !');

 Writeln('Sai lỗi ở vị trí thứ ' , maloi); Readln;

 END.
```

Khi chạy, ta sẽ thấy trên màn hình:

123.45 và maloi = 0

St = 123.XXX không đổi thành số được !

Sai lỗi ở vị trí thứ 5

\* **Hàm LENGTH (St)**

Ý nghĩa: Cho kết quả là một số nguyên chỉ độ dài của chuỗi ký tự St.

Để viết 1 chuỗi ký tự ở trung tâm màn hình, ta có thể dùng thủ thuật viết chuỗi là (80 - length(st)) div 2

Ví dụ 8.25:

```
USES CRT;

VAR St : String[80];

BEGIN
```

```
ClrScr ;

Write(' Nhập vào một câu : '); Readln(St) ;

Gotoxy(80 - Lenght(St)) div2, 12);

Writeln(St) ;

Readln ;
```

END.

\* **Hàm COPY (St, Pos, Num)**

Ý nghĩa: Cho kết quả là một chuỗi ký tự mới có được bằng cách chép từ chuỗi St, bắt đầu từ vị trí Pos và chép Num ký tự.

Nếu vị trí Pos lớn hơn chiều dài của chuỗi St thì hàm COPY sẽ cho một chuỗi rỗng. Nếu giá trị của vị trí Pos và số ký tự Num (Pos + Num) lớn hơn chiều dài của chuỗi St thì hàm COPY chỉ nhận các ký tự nằm trong chuỗi St.

Ví dụ 8.26:

```
VAR St1, St2 : string[25] ;

BEGIN

 St1 := 'UNIVERSITY OF CANTHO : 1966 - 1996' ;

 St2 := COPY (St1, 15, 6) ;

END.
```

Như vậy, giá trị của biến St2 bây giờ là CANTHO.

\* **Hàm CONCAT (St1, St2, ..., StN)**

Ý nghĩa: Cho kết quả là một chuỗi mới được ghép theo thứ tự từ các chuỗi St1, St2, ..., StN. Hàm này giống như phép cộng các chuỗi. Chuỗi mới cũng sẽ không được vượt quá 255 ký tự.

\* **Hàm POS (Obj, St) :**

Ý nghĩa: Cho kết quả là vị trí đầu tiên của chuỗi Obj trong chuỗi St. Nếu không tìm thấy thì hàm POS cho giá trị 0.

Ví dụ 8.27:

nếu St := 1234567890, nếu Obj := 456 thì POS (Obj, St) = 4 còn POS(4X, St)=0

**d. Truy xuất từng ký tự trong chuỗi**

Ta có thể truy xuất đến từng ký tự trong chuỗi với tên biến và chỉ số trong dấu ngoặc vuông [ ] như truy xuất các phần tử của mảng. Ví dụ với chuỗi St thì St[i] là ký tự thứ i trong chuỗi St, dĩ nhiên                     . Chỉ số i chạy dài từ 1 đến độ dài lớn nhất của chuỗi ký tự.

Ví dụ 8.28:

```
PROGRAM DoiChu;

VAR St:String;

 i: integer;

BEGIN

 Write('Hãy nhập tên của bạn : ');

 Readln(St);

 FOR i:= 1 TO Length(St) DO

 St[i] := Ucase(St[i]);

 (*Hàm Ucase đổi ký tự thành chữ in hoa*)

 Writeln;

 Writeln(St);

 Readln;

END.
```

### III. KIỂU TẬP HỢP (SET)

#### 1. Định nghĩa và khai báo

Một tập hợp (SET) bao gồm một số các phần tử có cùng bản chất kiểu là kiểu cơ bản. Trong Turbo Pascal và IBM Pascal, số phần tử tối đa trong một tập hợp là 256. Kiểu cơ bản có thể là kiểu vô hướng liệt kê, kiểu miền con hoặc kiểu Char, không được là số thực. Khái niệm tập hợp trong Pascal tương tự như khái niệm tập hợp trong toán học.

+ Khai báo gián tiếp

#### TYPE

<Kiểu cơ bản> = (phần\_tử\_1, phần\_tử\_2, ..., phần\_tử\_n) ;

<Tên kiểu tập hợp> = SET OF <Kiểu cơ bản> ;

#### VAR

<Tên biến > : <Tên kiểu tập hợp> ;

Ví dụ 8.29: TYPE

Sizes = (short, medium, large) ;

Shirtsizes = SET OF sizes ;



VAR

shortleeve, longleeve : shirtsizes ;

+ Khai báo trực tiếp

VAR

<Tên biến> : SET OF <Kiểu cơ bản> ;

Ví dụ 8.30:

VAR

Chu : SET OF Char ;

So : SET OF 0 .. 9 ;

ABC : SET OF 0 .. 256 ;

Date : SET OF (Sun, Mon, Tue, Wed, Fri, Sat) ;

## 2. Mô tả một tập hợp

Một tập hợp được mô tả bằng cách liệt kê các phần tử của tập hợp, chúng cách nhau bằng một dấu phẩy (,) và được đặt giữa hai dấu móc vuông [ ], các phần tử có thể là hằng, biến hoặc biểu thức.

Ví dụ 8.31:

[] {tập hợp rỗng, không có các phần tử }

[5 .. 15] {tập hợp các chữ số nguyên từ 5 đến 15}

[1, 3, 5] {tập hợp 3 số 1, 3 và 5 }

[Hồng, Lan, Cúc, Mai] {tập hợp tên 4 loài hoa}

[i, i + j\*2, 4, 5] {tập hợp các biến nguyên gồm số 4, 5 và các số nhận từ i, i + j\*2 với i, j là 2 biến nguyên}

## 3. Các phép toán trên tập hợp

### a. Phép gán

Ta có thể gán giá trị các tập đã được mô tả vào các biến tập cùng kiểu. Riêng tập hợp rỗng có thể gán cho mọi biến kiểu tập hợp khác nhau.

Với ví dụ trên, ta có thể gán :

Chu := [X,Y,Z] ;

So := [2,3,4] ;

Date := [] ;

Nếu ta viết  $Chu := [1,2]$ ; thì không hợp lệ vì Chu là tập hợp các chữ.

**b. Phép hợp**

Hợp của 2 tập hợp A và B là một tập hợp chứa tất cả các phần tử của tập A hoặc B hoặc cả A và B.

Ký hiệu của phép hợp là dấu cộng (+). Phép hợp có tính giao hoán:

$$A+B = B+A$$

Ta có thể mô tả phép hợp qua hình ảnh sau :



Ví dụ 8.32

$$A := [0,1,3,5,7,9] ;$$

$$B := [0,2,4,6,8,9] ;$$

$$C := A + B ;$$

{tập hợp C sẽ có các phần tử là  $[0,1,2,3,4,5,6,7,8,9]$  }

**c. Phép giao**

Giao của 2 tập hợp A và B là một tập chứa các phần tử của cả A và cả B.

Ký hiệu  $A * B$ . Phép giao cũng có tính giao hoán, nghĩa là  $A * B = B * A$

Minh họa như sau :



Với ví dụ trong phép hợp, nếu:

$$D := A * B ; \quad \{ \text{tập D chứa phần tử } [0,9] \}$$

Nếu A và B không có phần tử nào giống nhau thì phép hợp sẽ cho tập rỗng.

**d. Phép hiệu**

Hiệu của 2 tập hợp A và B, ký hiệu là  $A - B$ , là một tập hợp chứa các phần tử chỉ thuộc A mà không thuộc B. Lưu ý :  $A - B$  thì khác  $B - A$ .

Ví dụ 8.33:  $A := [3 .. 7] ;$

B := [1.. 6, 10, 15] ;

thì A - B là tập hợp [7] còn B - A là tập hợp [1,2, 10,15]

**e. Phép thuộc IN**

Phép thuộc IN cho phép thử xem một giá trị nào đó thuộc về một tập hay không? Phép thuộc IN cho kết quả có kiểu Boolean. Nếu đúng nó sẽ cho kết quả là TRUE, ngược lại là FALSE.

Ví dụ 8.34: Chu là biến kiểu Char, còn A là biến kiểu SET OF Char và

Chu := 'X' ;

A := ['X', 'x', 'Y', 'y', 'Z', 'z'] ;

thì phép toán Chu IN A sẽ cho kết quả là TRUE

**f. Các phép so sánh =, <>, <= và >=**

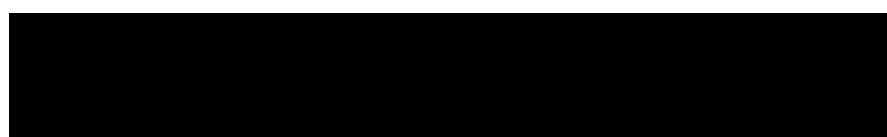
Muốn so sánh 2 tập hợp với nhau thì chúng phải có cùng kiểu cơ bản. Kết quả của các phép so sánh là giá trị kiểu Boolean, tức là TRUE (Đúng) hoặc FALSE (Sai).

Hai tập hợp A và B gọi là bằng nhau (A = B) chỉ khi chúng có các phần tử giống với nhau từng đôi một (không kể thứ tự sắp xếp các phần tử trong 2 tập). Ngược lại của phép so sánh bằng nhau (=) là phép so sánh khác nhau (<>). Nghĩa là, nếu A = B là TRUE thì A <> B sẽ là FALSE và ngược lại.

Phép so sánh nhỏ hơn hoặc bằng (<=) của A <= B sẽ cho kết quả là TRUE nếu mọi phần tử có trong A đều có trong B. Định nghĩa này cũng tương tự như lớn hơn hoặc bằng (>=). Với A >= B thì mọi phần tử của B đều có trong A, kết quả này TRUE, ngược lại là FALSE.

Chú ý: Trong Pascal không có phép so sánh nhỏ hơn (<) và lớn hơn (>). Để kiểm tra xem tập A có thực sự nằm trong tập B hay không (A nhỏ hơn B), ta phải sử dụng thêm các phép logic như sau:

IF (A <> B) AND (A <= B) THEN WRITELN ('A < B')



**4. Viết và đọc dữ liệu kiểu tập hợp**

Với dữ liệu kiểu tập hợp, ta không thể viết ra hoặc đọc vào bằng các thủ tục (Write) Writeln hoặc (Read) Readln. Tuy nhiên, ta có thể thực hiện các thao tác này khi mà kiểu cơ bản của tập hợp là số nguyên, ký tự.

Ví dụ 8.35: Viết chương trình để đọc một câu bất kỳ, sắp xếp các chữ của câu đó theo thứ tự ABC abc từ chữ in đến chữ thường. Chương trình chấm dứt khi nhận được chữ END hoặc end.

```
PROGRAM Letters_used ;

TYPE letters = SET OF char ;

VAR used, unused : letters ;

count, charcount : 0 .. 80 ;
```

```
alpha : char ;

line : string ;

PROCEDURE Readinput ; {đọc một câu bất kỳ}

BEGIN

FOR count := 1 TO 80 DO line[count] := ' ' ;

Writeln (' Nhập vào một dòng câu dưới đây : ') ;

Count := 0;

WHILE NOT eoln DO {hàm eoln trả về giá trị false khi ký tự nhận vào khác}

 BEGIN {ký tự kết thúc dòng CR: carry return}

 count := count + 1 ;

 read(line[count]);

 END ;

readln;

charcount := count;

END ;

PROCEDURE Writeoutput ; {trình bày phân tích của một dòng câu }

BEGIN

writeln;

write(' Các chữ đã sử dụng: ') ;

FOR alpha := 'A' to 'z' DO

 IF [alpha] <= used THEN write(' ', alpha) ;

writeln;

writeln;

END;

BEGIN {Thân chương trình chính}

Readinput;

WHILE NOT (([line[1]] <= ['E', 'e']) AND ([line[2]] <= ['N', 'n'])

AND ([line[3]] <= ['D', 'd'])) DO
```

```
BEGIN

 used := [] ;

 unused := ['A' .. 'Z', 'a' .. 'z'] ;

 FOR count := 1 TO charcount DO

 IF [line[count]] <= unused THEN

 BEGIN

 used := used + [line[count]] ;

 unused := unused - [line[count]] ;

 END ;

 Writeoutput ;

 Readinput ;

 END ;

END.
```

Khi chạy chương trình, ta sẽ thấy (Các dòng chữ gạch dưới là của người dùng):

Nhập vào một dòng câu dưới đây:

Pascal is a structured programming language derived from ALGOL - 60

Các chữ đã sử dụng: A G L O P a c d e f g i l m n o p r s t u v

Nhập vào dòng câu dưới đây:

END

#### IV. KIỂU BẢN GHI (RECORD)

##### 1. Định nghĩa và khai báo

Các cấu trúc dữ liệu kiểu mảng (Array) và tập hợp (Set) có hạn chế ở chỗ các phần tử trong tập hợp của chúng phải cùng kiểu mô tả. Song trong thực tế, có những kiểu cấu trúc dữ liệu khác nhau nhưng lại có một mối liên quan nào đó.

Ví dụ 8.36:

Để mô tả dữ liệu về lý lịch một người nào đó, người ta phải khai báo họ tên người (kiểu String), Phái (Nam :=True, Nữ := False theo kiểu Boolean), ngày sinh (mô tả kiểu date), địa chỉ (kiểu String) và mức lương (kiểu integer), v.v... Với các kiểu cơ bản khác nhau như vậy trong Pascal, ta phải dùng kiểu bản ghi (RECORD).

Kiểu bản ghi trong ngôn ngữ Pascal gắn liền với kiểu tập tin (FILE) - sẽ được trình bày trong phần kế tiếp. Tuy nhiên, ta có thể sử dụng RECORD một cách độc lập với FILE.

RECORD là kiểu dữ liệu bao gồm nhiều thành phần có thể khác nhau về kiểu dữ liệu, mỗi thành phần được gọi là trường (Field).

Cú pháp khai báo kiểu bản ghi (Record) trước rồi khai báo biến như sau:

+ *Khai báo gián tiếp:*

**TYPE**

<Tên kiểu bản ghi> = RECORD

<Tên trường 1a>[,<Tên trường1b>,...] : <Kiểu trường> ;

<Tên trường 2a>[,<Tên trường2b>,...] : <Kiểu trường> ;

..... ;

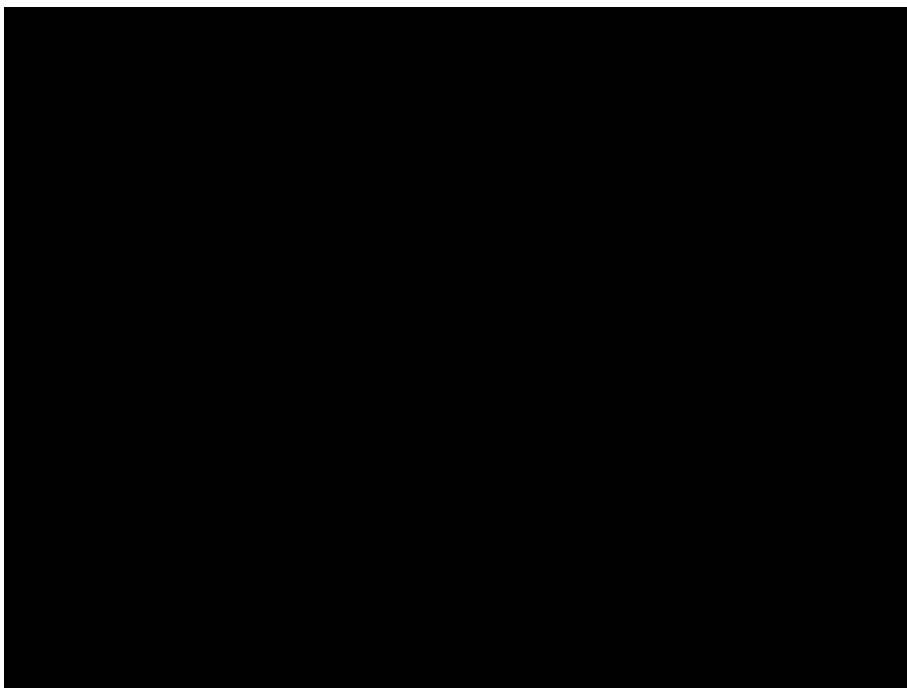
END ;

**VAR**

<Tên biến1>[,<Tên biến2>, ...] : <Tên kiểu bản ghi> ;

Ví dụ 8.37:

Ta đang làm một khai báo về khách hàng của công ty chuyên bán hàng trả góp nào đó. Số liệu cần sử dụng là ngày tháng làm bản ghi và các thông tin cơ bản về khách hàng nợ thanh toán cho công ty, theo minh họa ở hình dưới:



Trong chương trình này, Công ty phân ra 3 tình trạng loại khách nợ (status): đúng kỳ hạn phải trả (current), đã quá hạn phải trả (overdue) và loại khách chệnh mảng, dây dưa việc trả nợ nhiều lần (delinquent). Ở đây:

- Status được khai báo theo kiểu dữ liệu liệt kê (enumerated data type).

- Account (Số kế toán) là một kiểu record, chứa các thông tin về tên và địa chỉ khách nợ (kiểu chuỗi string), số khách nợ (kiểu số nguyên integer-type), loại khách nợ (kiểu liệt kê enumerated type) và số liệu tồn đọng nợ của khách (kiểu số thực real-type).

- Date (Ngày tháng) là một kiểu Record trong chương trình ghi ngày, tháng năm đáo nợ của khách hàng.

Biến của chương trình là khách hàng (customer).

Ta có thể khai báo như sau:

TYPE status = (current, overdue, delinquent);

date = RECORD

day : 1 .. 31 ;

month : 1 ..12 ;

year : 1900 .. 2100 ;

END ;

account = RECORD

Custname : String ;

Custaddress : String ;

Custno : 1 .. 9999 ;

Custtype : status ;

Custbalance : Real ;

Lastpayment : date;

END;

VAR customer : account ;

+ *Khai báo trực tiếp*

**VAR**

<Tên biến1>[,<Tên biến2>, ...] : RECORD

<Tên trường 1a>[,<Tên trường1b>,...] : <Kiểu trường> ;

<Tên trường 2a>[,<Tên trường2b>,...] : <Kiểu trường> ;

..... ;

**END ;**

## 2. Truy xuất một Record

Để truy xuất vào một trường của kiểu Record, ta cần dùng tên biến kiểu Record, sau đó là dấu chấm (.) rồi đến tên trường. Dạng tổng quát sau:

<Biến Record>.<Tên trường>

Ví dụ 8.39: Nhập lý lịch nhân viên của một cơ quan

TYPE

Lylich = RECORD {Lý lịch gồm Họ tên, Tuổi, Giới, Lương}

Hoten : string [25] ;

Tuoi : integer ;

PhaiNam : boolean; {Nam : M (Male), Nữ : F (Female)}

Luong : real;

END;

VAR x, y : Lylich ;

nv : ARRAY [1 .. 200] OF Lylich ; {nv là mảng lý lịch các nhân viên}

.....  
Write('Nhập tổng số nhân viên : '); readln(n) ;

FOR i := 1 TO n DO

BEGIN

Write(' Họ tên : '); readln(nv[i].Hoten);

Writeln(' Tuổi : '); readln(nv[i].Tuoi) ;

Write(' Giới (Nam :M, Nữ : F) ? '); readln (Phai);

IF (Phai = 'M') or (Phai = 'm') THEN nv[i].PhaiNam := TRUE

ELSE nv[i].PhaiNam := FALSE ;

Writeln(' Lương : '); read(nv[i].Luong) ;

END ;

.....

### Lưu ý:

- Các biến Record có thể gán cho nhau. Ví dụ x và y là 2 biến bản ghi có cùng kiểu Lylich, thì ta có thể gán:



x := y;

Như vậy ta không phải lặp lại:

x.Hoten := y.Hoten ;

x.Tuoi := y.Tuoi ;

.....

· **Không được viết ra màn hình hoặc đọc từ bàn phím một biến record như :**

Writeln(x);    hoặc    Readln(x);

· **Không thể so sánh các record bằng các phép toán quan hệ <, >, <=, >=, =, <>**

· **Không được dùng các toán số học và logic với kiểu record.**

Ví dụ 8.40: Nhập vào 2 số phức C1 và C2 và tính C3 là tổng của chúng

Với chương trình loại này ta phải lần lượt nhập từng phần thực và phần ảo riêng rẽ của C1 và C2. Ta không thể dùng dòng lệnh C3 = C1 + C2. Kết quả tính C3 phải là phép cộng riêng rẽ từng phần thực và phần ảo của C1 và C2 rồi ghép lại.

PROGRAM So\_Phuc ;

TYPE

Sophuc = Record

pt, pa : real ;

End;

VAR

c1, c2, c3 : Sophuc ;

BEGIN

Write('Lần lượt nhập phần thực và phần ảo của 2 số phức C1 và C2') ;

Write('Nhập phần thực của số phức C1 : ') ; Readln(c1.pt) ;

Write('Nhập phần ảo của số phức C1 : ') ; Readln(c1.pa) ;

Write('Nhập phần thực của số phức C2 : ') ; Readln(c2.pt) ;

Write('Nhập phần ảo của số phức C2: ') ; Readln(c2.pa) ;

c3.pt := c1.pt + c2.pt ;

c3.pa := c1.pa + c2.pa ;

Writeln('Kết quả của phép cộng 2 số phức :');

```

Write('C3 = C1 + C2 ');

Write(' = (', c1.pt:5:2, '+i ', c1.pa:5:2, ') +(', c2.pt:5:2, '+i ', c2.pa:5:2, ') ');

Write('C3 = ', c3.pt:5:2, '+i', c3.pa:5:2);

Readln;

```

END.

### 3. Các Record lồng nhau

**Record lồng nhau là record mà có trường (field) của nó lại có kiểu là một record khác.** Ta có thể định nghĩa các record lồng nhau theo một cách trực tiếp hay gián tiếp nhau và cách khai báo cũng hoàn toàn tương tự như cách khai báo record ở trên.

Ví dụ 8.41:

```

TYPE
 dd_mm_yy = Record
 dd:1..31;
 mm:1..12;
 yyyy:1900..2100;
 end;

 hoso = Record
 masv:string[7];
 ngsinh:dd_mm_yyyy;
 diem:real;
 end;

VAR
 Lop: Array[1..20] of hoso;

```

### 4. Câu lệnh WITH

Nhiều chương trình đòi hỏi các phần tử khác nhau của cùng một record phải được thao tác tại các vị trí khác nhau bên trong chương trình. Như vậy phải cần có nhiều chỉ thị trường khác nhau đặc trưng. Việc này làm chương trình trở nên phức tạp, tẻ nhạt và khó theo dõi. Để giải quyết tình trạng này, Pascal đã đưa ra cấu trúc câu lệnh WITH ... DO trong record nhằm bớt đi các rắc rối từ các chỉ thị trường (hay nói cách khác, câu lệnh WITH .. DO như là phép toán đặt thừa số chung mà ở đó thừa số chung là các tên biến record).

Dạng tổng quát của câu lệnh WITH là:

**WITH <tên biến record> DO**

**BEGIN**

<Câu lệnh với tên trường 1>;

..... ;

<Câu lệnh với tên trường n>;

**END ;**

Ví dụ 8.42:

Một biến bản ghi DANSO có các trường KHUVUC, HOTEN, NGAYSINH, DIACHI, NGHE đưa dữ liệu từ bàn phím như sau:

**WITH DANSO DO**

**BEGIN**

Write ('Khu vực điều tra :'); Readln (KHUVUC);

Write ('Họ tên công dân :'); Readln (HOTEN);

Write ('Ngày-tháng-năm sinh :'); Readln (NGAYSINH);

Write ('Địa chỉ công dân :'); Readln (DIACHI);

Write ('Nghề nghiệp : '); Readln (NGHE);

**END ;**

Đối với bản ghi có nhiều thứ bậc:

R1 là biến bản ghi có trường R2

R2 là biến bản ghi con có trường R3

R3 là biến bản ghi con có trường R4

.....

thì câu lệnh WITH được tổ chức lồng nhau:

**WITH R1 DO**

**WITH R2 DO**

.....

**WITH RN DO <Câu lệnh ...> ;**

hay viết đơn giản hơn:

**WITH Record1, Record2, ..., RecordN DO <Câu lệnh BEGIN .. END> ;**

Ở ví dụ 8.41, ta có thể viết như sau:

WITH Lop[i] DO

Begin

With Ngsinh Do

Begin dd:= 25; mm:=05; yyyy:=1978; End;

masv:='7962343'; diem:=9.0;

End;

## PHỤ CHƯƠNG

### ĐỒ HỌA TRONG TURBO PASCAL

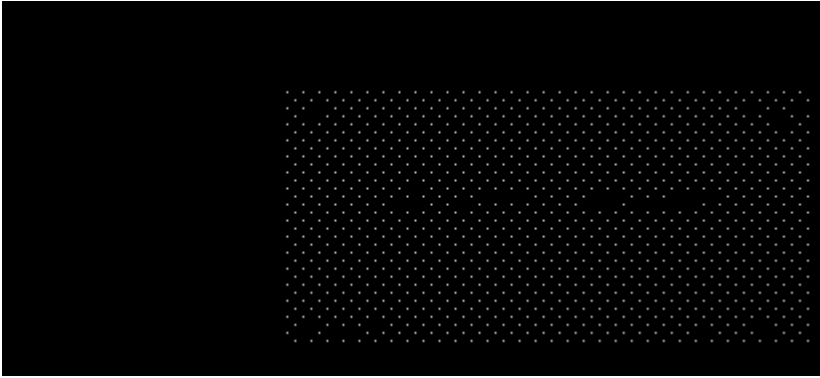
- I. [KHÁI NIỆM VỀ ĐỒ HOA.](#)
- II. [KHỞI ĐỘNG CHẾ ĐỘ ĐỒ HOA.](#)
- III. [LỖI ĐỒ HOA.](#)
- IV. [MỘT SỐ HÀM VÀ THỦ TỤC CHUẨN TRONG UNIT GRAPH.](#)
  1. [Màu và mẫu\(kiểu\).](#)
  2. [Điểm, kiểu đường thẳng, đường thẳng.](#)
  3. [Các hình không tô.](#)
  4. [Các hình có tô.](#)
  5. [Xử lý chuỗi ký tự trên màn hình đồ họa.](#)
  6. [Cửa sổ trong chế độ đồ họa.](#)
  7. [Đóng chế độ đồ họa.](#)
- V. [MỘT VÀI VÍ DỤ MINH HOA.](#)

Phần này chỉ nhằm giới thiệu một số các khái niệm cơ bản về chế độ đồ họa, các thủ tục và hàm để: khởi động chế độ đồ họa, vẽ các hình cơ bản như đường thẳng, đường tròn, cung elip, hình quạt, đa giác, chuỗi ký tự, cửa sổ ViewPort,... đã được khai báo sẵn trong Unit Graph của Turbo Pascal.

#### I. KHÁI NIỆM VỀ ĐỒ HỌA

Trong Turbo Pascal có hai chế độ thường được sử dụng đó là: chế độ văn bản (text mode) và chế độ đồ họa (graphics mode). Chế độ văn bản thì màn hình được thiết lập hiển thị 25 dòng và 80 cột như đã giới thiệu ở phần trước. Còn ở chế độ đồ họa, thì màn hình lại được thiết lập dựa trên từng điểm ảnh (pixel), mỗi màn hình gồm nhiều điểm ảnh được sắp xếp trên các đường thẳng nằm ngang và đường thẳng đứng, sự bài trí số pixel trên mỗi loại màn hình được gọi là độ phân giải (Resolution), độ phân giải càng cao thì số pixel càng nhiều và hình ảnh càng mịn.

Hệ tọa độ cho mỗi loại màn hình có khác nhau (xem bảng 1), chẳng hạn ở loại màn hình VGA là 640x480 như hình sau:



Một chương trình đồ họa thường gồm các phần sau:

- Khởi tạo chế độ đồ họa.
- Xác định màu nền, màu và kiểu chữ, màu đường vẽ, màu tô và kiểu tô.
- Vẽ và tô các hình ta cần thực hiện.
- Các thao tác khác như cho hiện dòng chữ, chú thích.
- Đóng chế độ đồ họa để trở về chế độ văn bản.

## II. KHỞI ĐỘNG CHẾ ĐỘ ĐỒ HỌA

Để có thể thực hiện chương trình trên chế độ đồ họa, trước tiên ta phải khởi động chế độ đồ họa. Việc này được thông qua thủ tục sau:

Procedure InitGraph(var GraphDriver:Integer; var GraphMode: Integer; PathToDriver: string);

Với: GraphDriver, GraphMode là loại màn hình và một màn hình (xem bảng 1).

PathToDriver là đường dẫn của thư mục chứa các tập tin điều khiển đồ họa.

Ví dụ: Giả sử ta có loại màn hình VGA, một là VGAI, các tập tin điều khiển đồ họa ở trong thư mục F:\WINAPPS\BP\BGI, ta có thể viết phần chương trình khởi động đồ họa như sau:

```
Uses Graph;

Var mh,mode: integer; {mh: loại màn hình}

Begin

 mh:=VGA; mode:=VGAI; {có thể dùng hằng DETECT cho mh } (1)

 Initgraph(mh,mode,' F:\WINAPPS\BP\BGI');

End.
```

Việc nhớ các loại màn hình và một màn hình là điều gây dễ nhầm lẫn, do vậy ta có thể để cho máy tự động dò tìm loại và một màn hình. Như vậy ở chương trình trên ta bỏ dòng (1) thì khi thực hiện máy sẽ tự động dò tìm (DETECT), đây là điểm rất hay vì nó sẽ cho khởi động loại màn hình đang sử dụng và một đồ họa có độ phân giải cao nhất.

Bảng 1: Các giá trị của Grapdriver, GraphMode và Resolution của một số loại màn hình thông dụng.

| <b>GraphDriver</b> | <b>GraphMode</b> | <b>Resolution</b>                              |
|--------------------|------------------|------------------------------------------------|
| DETECT (0)         |                  | Độ phân giải cao nhất và loại màn hình sử dụng |
| CGA (1)            | CGAC0 (0)        | 320 x 200                                      |
|                    | CGAC1 (1)        | 320 x 200                                      |
|                    | CGAC2 (2)        | 320 x 200                                      |
|                    | CGAC3 (3)        | 320 x 200                                      |
|                    | CGACHi (4)       | 640 x 200                                      |
| EGA (3)            | EGALo (0)        | 640 x 200                                      |
|                    | EGAHi (1)        | 640 x 3 50                                     |
| VGA (9)            | VGALo (0)        | 640 x 200                                      |
|                    | VGAMed (1)       | 640 x 350                                      |
|                    | VGAHi (2)        | 640 x 480                                      |

Chú ý: Ở bảng này các hằng DETECT có giá trị 0, hằng VGA có giá trị 9, hằng VGAHi có giá trị 2, ...

Ta có thể xác định Grapdriver, GraphMode và Resolution trên một máy đang hoạt động bằng chương trình sau:

```
Uses Graph;

Var mh,mode:integer;

Begin
 mh:=Detect; {Có thể bỏ dòng này}

 initgraph(mh,mode,'f:\winapps\bp\bgi');

 writeln('GraphDriver = ',mh,' Graphmode = ',mode,' Resolution = ',GetmaxX,
 'x',GetMaxY);

 readln;

 closegraph;

End.
```

Với: GetmaxX, và GetMaxY là các hàm trả về giá trị lớn nhất tương ứng cho hàng, cột của màn hình và một hiện hành.

### III. LỖI ĐỒ HỌA

Khi khởi động đồ họa, nếu máy không tìm thấy các chương trình điều khiển đồ họa thì sẽ phát sinh ra lỗi đồ họa và như vậy chương trình không thể thực hiện được hoặc có thể treo máy.

Trong cả 2 trường hợp có hoặc không có lỗi, ta nên sử dụng hàm GraphResult để biết có lỗi hay không? Có thể kết hợp với hàm GraphErrorMsg để nhận được thông báo đúng nhất. Bảng 2 liệt kê một số thông báo lỗi thường gặp:

Bảng 2: Các lỗi đồ họa

| Hằng             | Trị mã lỗi | Thông tin lỗi phát hiện                        |
|------------------|------------|------------------------------------------------|
| GrOk             | 0          | Không có lỗi khởi động đồ họa.                 |
| GrNoInitgraph    | -1         | Chưa khởi động đồ họa.                         |
| GrNotDetected    | -2         | Không có phần cứng đồ họa.                     |
| GrFileNotFound   | -3         | Không tìm thấy trình điều khiển đồ họa.        |
| GrInvalidDriver  | -4         | Trình điều khiển không hợp lệ.                 |
| GrNoLoadMem      | -5         | Không đủ bộ nhớ RAM cho đồ họa.                |
| GrNoScanMem      | -6         | Tràn vùng nhớ trong Scan Fill.                 |
| GRNoFloodMem     | -7         | Tràn vùng nhớ trong Flood Fill.                |
| GrFontNotFound   | -8         | Không tìm thấy tập tin Font.                   |
| GrNoFontMem      | -9         | Không đủ bộ nhớ RAM để nạp Font.               |
| GrInvalidMode    | -10        | Kiểu đồ họa không hợp lệ cho trình điều khiển. |
| GrError          | -11        | Lỗi đồ họa tổng quát.                          |
| GrIOError        | -12        | Lỗi đồ họa vào ra.                             |
| GrInvalidFont    | -13        | Tập tin Font không hợp lệ.                     |
| GrInvalidFontNum | -14        | Số hiệu đại diện cho Font không hợp lệ.        |

Ví dụ: Chương trình kiểm tra quá trình khởi động đồ họa và nếu có lỗi sẽ thông báo lỗi ra màn hình:

```
Uses Graph;
```

```
Var
```

```
maloi: Integer;
```

```
GrDriver, GrMode: Integer;
```

Begin

```
GrDriver := Detect;

InitGraph(GrDriver, GrMode, 'f:\winapps\bp\bgi');

maloi := GraphResult; { Check for errors }

if maloi <> grOk then

 begin

 Writeln('Lỗi đồ họa là : ', GraphErrorMsg(maloi));

 Writeln('Lỗi!, Thoát khỏi chương trình ...');

 Halt(1); {Lệnh ngắt ngang và kết thúc chương trình }

 end

else { Phần chương trình cần thực hiện khi không có lỗi đồ họa }

End.
```

#### IV. MỘT SỐ HÀM VÀ THỦ TỤC CHUẨN TRONG UNIT GRAPH

##### 1. Màu và mẫu (kiểu)

Đối với màn hình trắng đen (Hercules Monochrome) ta có 2 giá trị màu 0 và 1, còn đối với các màn hình màu (VGA, EGA,...) thì có 16 giá trị màu từ 0..15 được liệt kê trong bảng 3, và kiểu tô màu thì có 11 kiểu đã được định sẵn là từ 0..11 và một kiểu do người lập trình định nghĩa (User - defined fill) và các kiểu tô được liệt kê trong bảng 4.

###### a. Thủ tục chọn màu đường vẽ

**SetColor(ColorNum:word);**

###### b. Thủ tục đặt màu nền của màn hình

**SetBkColor(ColorNum :word);**

###### c. Thủ tục chọn kiểu tô và màu tô

**SetFillStyle(Pattern:word; ColorNum:word);**

###### d. Hàm nhận màu trả về do thủ tục SetColor đã đặt màu trước đó

**GetColor: word;**

###### e. Hàm nhận màu nền trả về do thủ tục SetBkColor đã đặt trước đó

**GetBkColor: word;**

###### f. Hàm trả về giá trị màu lớn nhất



**GetMaxColor: word;**

Bảng 3: Các giá trị có thể nhận của biến màu ColorNum

| Tên hằng     | Giá trị màu | Màu hiển thị      |
|--------------|-------------|-------------------|
| Black        | 0           | Đen               |
| Blue         | 1           | Xanh da trời      |
| Green        | 2           | Xanh lá cây       |
| Cyan         | 3           | Xanh lơ           |
| Red          | 4           | Đỏ                |
| Magenta      | 5           | Tím               |
| Brown        | 6           | Nâu               |
| LightGray    | 7           | Xám nhạt          |
| DarkGray     | 8           | Xám sẫm           |
| LightBlue    | 9           | Xanh da trời nhạt |
| LightGreen   | 10          | Xanh lá cây nhạt  |
| LightCyan    | 11          | Xanh lơ nhạt      |
| LightRed     | 12          | Đỏ nhạt           |
| LightMagenta | 13          | Tím nhạt          |
| Yellow       | 14          | Vàng              |
| White        | 15          | Trắng             |

Bảng 4: Các giá trị có thể nhận của biến kiểu tô Pattern

| Tên hằng    | Giá trị kiểu tô | Diễn giải kiểu tô  |
|-------------|-----------------|--------------------|
| EmptyFill   | 0               | Tô bằng màu nền    |
| SolidFill   | 1               | Tô đặc             |
| LineFill    | 2               | Tô bằng gạch ngang |
| LtSlashFill | 3               | Tô bằng ///        |
| SlashFill   | 4               | Tô bằng /// in đậm |
| BkSlashFill | 5               | Tô bằng \\\ in đậm |

|                |    |                                  |
|----------------|----|----------------------------------|
| LtBkSlashFill  | 6  | Tô bằng \\\                      |
| HatchFill      | 7  | Tô bằng đường gạch bóng nhạt     |
| XHatchFill     | 8  | Tô bằng đường gạch bóng chữ thập |
| InterleaveFill | 9  | Tô bằng đường đứt quãng          |
| WideDotFill    | 10 | Tô bằng những dấu chấm thưa      |
| CloseDotFill   | 11 | Tô bằng dấu chấm dày             |
| UserFill       | 12 | Mẫu tô tự tạo                    |

**2. Điểm, kiểu đường thẳng, đường thẳng**

**a. Thủ tục vẽ một điểm tại một tọa độ (x,y)**

```
PutPixel(x,y:integer;ColorNum:word);
```

**b. Thủ tục chọn kiểu đường thẳng**

```
SetLineStyle(linestyle:word;pattern:word;thickness:word);
```

Với tham số LineStyle có các giá trị như bảng 5 sau:

Bảng 5: Tham số LineStyle

| <b>Hằng</b> | <b>Giá trị</b> | <b>Diễn giải</b> |
|-------------|----------------|------------------|
| SolidLn     | 0              | Nét đậm          |
| DottedLn    | 1              | Nét chấm         |
| CenterLn    | 2              | Nét chấm gạch    |
| DashedLn    | 3              | Nét gạch         |
| UserBitLn   | 4              | Mẫu tự tạo       |

Với tham số thickness có các giá trị như bảng 6 sau:

Bảng 6: Tham số Thickness

| <b>Hằng</b> | <b>Giá trị</b> | <b>Diễn giải</b>   |
|-------------|----------------|--------------------|
| NormWidth   | 1              | Bề dày bình thường |
| ThickWidth  | 3              | Bề dày đậm         |

**c. Thủ tục vẽ đường thẳng từ tọa độ (x1 ,y1) đến tọa độ (x2 ,y2)**

**Line(x<sub>1</sub>,y<sub>1</sub>,x<sub>2</sub>,y<sub>2</sub>: integer);**

### 3. Các hình không tô

#### a. Thủ tục vẽ hình chữ nhật

**Rectangle(x<sub>1</sub>,y<sub>1</sub>,x<sub>2</sub>,y<sub>2</sub>:integer);**

#### b. Thủ tục vẽ hình tròn

**Circle(x,y:integer;r:word);**

Với x,y là tọa độ tâm, r là bán kính.

#### c. Thủ tục vẽ một cung tròn

**Arc(x,y:integer; StAngle,EndAngle:word; r :word);**

Với StAngle là góc bắt đầu, EndAngle là góc kết thúc.

#### d. Thủ tục vẽ cung Ellipse hoặc một Ellipse

**Ellipse(x,y:integer;StAngle,EndAngle:word;Xradius,Yradius:word);**

Nếu StAngle = 0 và EndAngle =360 thì là một hình Ellipse, nếu EndAngle < 360 thì là một cung Ellipse.

### 4. Các hình có tô

#### a. Đường gấp khúc (đa giác)

Muốn vẽ một đường gấp khúc đi qua n điểm tọa độ : (x<sub>1</sub> ,y<sub>1</sub>), (x<sub>2</sub> ,y<sub>2</sub>),..., (x<sub>n</sub> ,y<sub>n</sub> ) thì ta phải đưa tọa độ n điểm này vào một mảng poly nào đó mà mỗi phần tử của mảng có kiểu PointType đã được định nghĩa sẵn như sau:

Type

PointType = Record

x,y : integer;

end;

Khi điểm cuối (x<sub>n</sub> ,y<sub>n</sub> ) có tọa độ trùng với điểm đầu (x<sub>1</sub> ,y<sub>1</sub> ) thì n điểm này sẽ tạo thành một đường gấp khúc khép kín.

Dùng thủ tục DrawPoly(n, poly); để vẽ đường gấp khúc đi qua n tọa độ đã định sẵn trong mảng poly.

Dùng thủ tục FillPoly(n,poly); để vẽ và tô đường gấp khúc đi qua n tọa độ đã định sẵn trong mảng poly.

Ví dụ:

Uses Graph;

Const

gk: Array[1..3] of pointtype=((x:5;y:200),(x:190;y:5),(x:100;y:300)); {gấp khúc}

gkkk: Array[1..4] of Pointtype = ((x:405;y:200),(x:590;y:5),(x:500;y:300),(x:405;y:200));

Var Gd, Gm: Integer;

Begin

Gd := Detect;

InitGraph(Gd, Gm, 'F:\WINAPPS\BP\BGI');

if GraphResult <> grOk then Halt(1);

SetBkcolor(CYAN);

SetColor(YELLOW);

SetFillStyle(SolidFill,MAGENTA);

DrawPoly(3,gk);

FillPoly(3,gk); {đường gấp khúc không khép kín}

FillPoly(4,gkkk); {đường gấp khúc khép kín}

Readln;

CloseGraph;

End.

### b. Thủ tục vẽ hình chữ nhật

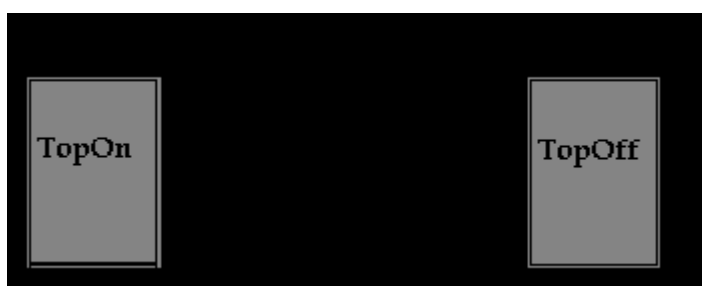
**Bar(x<sub>1</sub>,y<sub>1</sub>,x<sub>2</sub>,y<sub>2</sub>: integer);**

### c. Thủ tục vẽ hình hộp chữ nhật

**Bar3D(x<sub>1</sub>,y<sub>1</sub>,x<sub>2</sub>,y<sub>2</sub>:integer; depth:word;top:boolean);**

Tham số depth: là số điểm trên bề sâu của khối 3 chiều.

Tham số top có 2 giá trị được định nghĩa sẵn là: TopOn (True) tương ứng khối 3 chiều sẽ có nắp và TopOff (False) sẽ ứng với khối 3 chiều không có nắp (xem hình vẽ).



**d. Thủ tục vẽ hình Ellipse**

**FillEllipse(x,y:integer;xradius,yradius:word);**

**e. Thủ tục vẽ hình quạt**

**PieSlice(x,y:integer; StAngle,EndAngle,radius:word);**

**5. Xử lý chuỗi ký tự trên màn hình đồ họa**

**a. Thủ tục nạp Font chữ**

Các font chữ nằm trong các tập tin có phần mở rộng là .CHR . Để nạp các font chữ này ta dùng thủ tục:

**SetTextStyle(font,direction,charsize:word);**

Với: Tham số font có thể nhận một trong các giá trị sau:

|                    |             |   |
|--------------------|-------------|---|
| Hằng DefauFont     | hay giá trị | 0 |
| Hằng TriplexFont   | hay giá trị | 1 |
| Hằng SmallFont     | hay giá trị | 2 |
| Hằng SansSerifFont | hay giá trị | 3 |
| Hằng GothicFont    | hay giá trị | 4 |

Tham số direction có thể nhận một trong các giá trị sau:

|               |             |   |
|---------------|-------------|---|
| Hằng HorizDir | hay giá trị | 0 |
| Hằng VertDir  | hay giá trị | 1 |

Tham số charsize là cỡ ký tự và nó có thể nhận một trong các giá trị từ 1 đến 10.

**b. Thủ tục đặt chế độ căn chỉnh chuỗi văn bản**

**SetTextJustify(horiz, vert :word);**

Trong đó:

Tham số horiz có thể là một trong các hằng: LeftText, CenterText, RightText.

Tham số vert có thể là một trong các hằng: BottomText, CenterText, TopText.

**c. Thủ tục hiển thị chuỗi văn bản tại vị trí con nháy**

**OutText (text:string);**

**d. Thủ tục hiển thị chuỗi văn bản tại tọa độ (x,y)**

**OutTextXY (x,y:integer;text:string);**

## 6. Cửa sổ trong chế độ đồ họa (ViewPort)

Để thiết lập một cửa sổ trên màn hình đồ họa, ta sử dụng đến chức năng của ViewPort. Cửa sổ ViewPort được xem như một vùng chữ nhật trên màn hình đồ họa, nó giống như thủ tục Window trong chế độ văn bản (Text mode), nghĩa là ta có thể hiện một dòng văn bản, vẽ hình hoặc xóa chỉ nằm gọn trong ViewPort đã định, ta có thể minh họa một cửa sổ ViewPort như hình sau:



Để hiểu rõ cách thiết lập một ViewPort, ta hãy xét đến cách khai báo kiểu của ViewPort như sau:

```
ViewPortType = Record
 x1, y1, x2, y2 : Integer;
 Clip : Boolean;
End;
```

Trong đó: (x1, y1) , (x2, y2) lần lượt là góc tọa độ trên bên trái và tọa độ góc dưới bên phải, mà chúng phải thỏa tính chất sau: XXXXXXXXXX

Clip là một biến trường có kiểu Boolean mà nó có ý nghĩa như sau:

- Nếu có giá trị bằng True (hay bằng hằng ClipOn) thì không cho phép vẽ bên ngoài ViewPort.
- Nếu có giá trị bằng False (hay bằng hằng ClipOff) thì cho phép vẽ bên ngoài ViewPort.

### a. Thủ tục thiết lập một ViewPort

**SetViewPort(x1,y1,x2,y2:integer; Clip:Boolean);**

Sau khi thiết lập ViewPort ta sẽ có một hệ tọa độ mới mà góc trên bên trái của ViewPort sẽ có tọa độ (0,0).

Ví dụ: Như hình vẽ ở trên (giả sử ta chọn Clip bằng hằng ClipOn) thì ta phải thiết lập ViewPort như sau: SetViewPort(300,250,500,350,ClipOn);

\* Tọa độ âm dương


Với một số đồ thị của toán học phải có tọa độ âm dương, để vẽ nó ta phải chia màn hình ra làm 4 phần tương ứng với 4 vùng (I, II, III, IV) âm dương của một hệ trục tọa độ xy. Để thực hiện việc này, ta phải dùng đến cửa sổ ViewPort, với cách thiết lập sao cho tọa độ (0,0) của trục tọa độ xy là tâm tuyệt đối của màn hình góc trên bên trái của ViewPort như sau:

- Đặt:  $x1 = \text{GetmaxX};$        $y1 = \text{GetmaxX};$   
           $x2 = \text{GetmaxX};$        $y2 = \text{GetmaxX};$

- Dùng thủ tục `SetViewport(x1,y1,x2,y2,ClipOff)` , với `Clip = ClipOff` để có thể vẽ ra ngoài giới hạn của ViewPort.



Ví dụ:

Vẽ đồ thị hàm  $\sin(x)$  trong hệ trục tọa độ âm dương, với hoành độ 

```
Program Dothi;

Uses Crt,Graph;

Const

 ScaleX=20;

 ScaleY=80;

Var mh,mode,x,y,i:integer;

Begin

 InitGraph(mh,mode,'F:\WINAPPS\BP\BGI');

 SetViewport(GetmaxX DIV 2,GetmaxY DIV 2,GetmaxX,GetmaxY,ClipOff);

 SetColor(blue);

 Line(-(GetmaxX DIV 2),0,GetmaxX DIV 2,0);

 Line(0,-(GetmaxY DIV 2),0,GetmaxY DIV 2);

 SetTextJustify(CenterText,CenterText);
```

```
SetColor(White);

OutTextXY(-GetmaxX DIV 4,-GetmaxX DIV 4,'DO THI HINH SIN ');

SetColor(Red);

OutTextXY(GetmaxX DIV 2- 32,2,"Truc x >");

OutTextXY(27,-(GetmaxY DIV 2-5),'^ Truc y');

OutTextXY(0,0,'0,0');

for i:= -400 to 400 do

begin

 x:=Round(2*Pi*i* ScaleX /200);

 y:=Round(Sin(2*Pi*i/200)* ScaleY);

 PutPixel(x,y,Yellow);

end;

Repeat Until KeyPressed;

CloseGraph;

End.
```

#### **b. Thủ tục nhận ViewPort hiện hành**

```
GetViewSettings(Var ViewPort: ViewPortType);
```

#### **c. Thủ tục xóa bên trong màn hình ViewPort hiện hành**

```
ClearViewPort;
```

Thủ tục xóa sạch tất cả các phần (hình vẽ, chuỗi ký tự, ...) bên trong ViewPort và đưa con trỏ về tọa độ (0,0) của cửa sổ ViewPort hiện hành.

#### **d. Thủ tục xóa sạch màn hình đồ họa**

```
ClearDevice;
```

#### **7. Đóng chế độ đồ họa**

Để trở về chế độ văn bản, ta dùng thủ tục:

```
CloseGraph;
```

### **V. MỘT VÀI VÍ DỤ MINH HỌA**

Ví dụ 1: Vẽ Bầu trời đầy sao



```
PROGRAM Vebautroi;

Uses Graph,Crt;

Var gd,gm,x,y:integer; maxcolor:word;

Begin

 Gd:=Detect;

 Initgraph(gd,gm,'C:\TP70\BGI');

 If Graphresult<> Grok Then Halt(1);

 x:=GetmaxX; y:=GetmaxY;

 Maxcolor:=Getmaxcolor;

 Randomize;

 While (not keypressed) do

 Begin delay(100);

 Putpixel(random(x),random(y),

 Random (maxcolor-1)+1);

 end;

 Closegraph;

 End.
```

Ví dụ 2: Vẽ đồ thị các hàm số: Sin(x),Cos(x) và Arctan(x)

```
PROGRAM VeDothi;

Uses

 Crt,Graph;

Var

 mh,mode:integer;chon:char;chugiai:string;

Procedure Chonham;

begin

 writeln('Các đồ thị có thể:');

 writeln('1---->Đồ thị hình Sin(x)');

 writeln('2---->Đồ thị hình Cos(x)');
```

```
writeln('3---->Đồ thị hình ArcTan(x)');
```

```
write('Chọn đồ thị nào ?'); readln(chon);
```

```
Case chon of
```

```
'1': chugiai:=ĐỒ THỊ HÀM SIN;
```

```
'2': chugiai:=ĐỒ THỊ HÀM COS;
```

```
'3': chugiai:=ĐỒ THỊ HÀM ARCTAN;
```

```
end;
```

```
end;
```

```
Function F(chon:char;x:real):real;
```

```
begin
```

```
Case chon of
```

```
'1': F:=Sin(x);
```

```
'2': F:=Cos(x);
```

```
'3': F:=Arctan(x);
```

```
end;
```

```
end;
```

```
Procedure Dothi(a,b:real;x1,y1,x2,y2:integer;mn,md:integer);
```

```
var fx,k,h,r,c,d:real;
```

```
 x,y,n,m,i:integer;
```

```
begin
```

```
 c:=f(chon,a); d:=f(chon,a);
```

```
 r:=a; h:=(b-a)/1000;
```

```
 while r <= b do
```

```
 begin
```

```
 fx:=f(chon,r);
```

```
 if c>fx then c:=fx;
```

```
 if d<fx then d:=fx;
```

```
 r:=r+h;
```

```
end;

Setcolor(md);Setbkcolor(mn);

n:=x2-x1;

h:=(b-a)/n;

m:=y2-y1;

k:=(d-c)/m;

for i:=0 to n do

begin

x:=x1+i;

fx:=f(chon,a+i*h);

y:=round((fx-c)/k)+y1;

y:=y2-y+y1;

if i=0 then moveto(x,y)

else lineto(x,y);

end;
```

end;

Begin (\* Chương trình chính \*)

```
Clrscr;

Chonham;

mh:=detect;

Initgraph(mh,mode,'u:\bgi');

Setviewport(GetmaxX DIV 2,GetmaxY DIV 2,GetmaxX,GetmaxY,ClipOff);

Line(-(GetmaxX DIV 2),0,GetmaxX DIV 2,0);

Line(0,-(GetmaxY DIV 2),0,GetmaxY DIV 2);

SetTextJustify(CenterText,CenterText);

OutTextXY(-(GetmaxX DIV 4),-(GetmaxX DIV 4),chugiai);

SetColor(Red);

OutTextXY(GetmaxX DIV 2- 32,2,'Truc x >');
```

```
OutTextXY(27,-(GetmaxY DIV 2-5),'^ Truc y');
```

```
OutTextXY(0,0,'0,0');
```

```
Dothi(-4*pi,4*pi,-(getmaxx div 2)+100,-(getmaxy div 2)+100,getmaxx div 2 -100,
Getmaxy div 2 - 100,magenta,yellow);
```

```
Readln;
```

```
Closegraph;
```

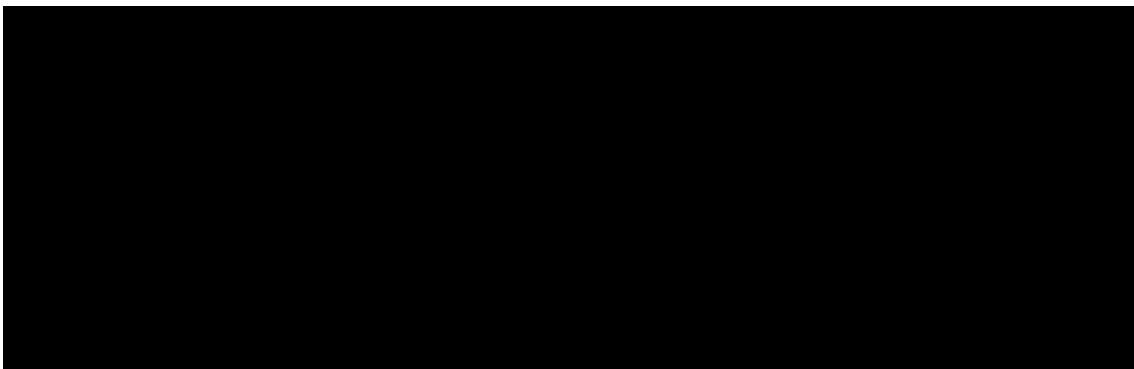
End.

## BÀI ĐỌC THÊM

### TÓM TẮT

### CÁC CÚ PHÁP TRONG NGÔN NGỮ PASCAL

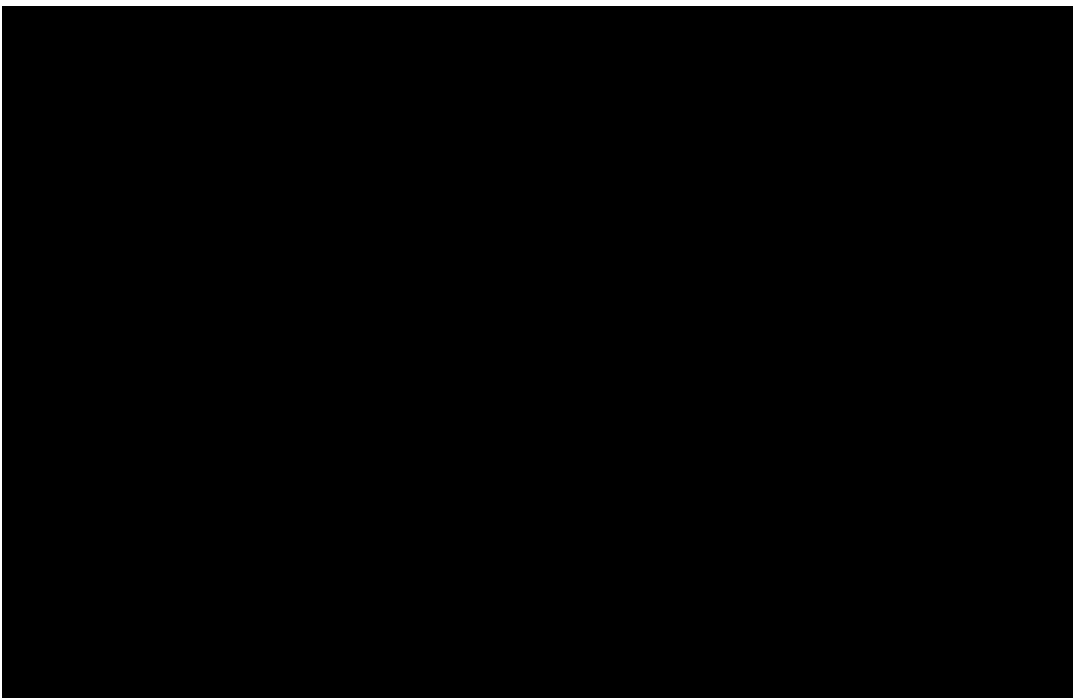
Niklaus Writh, cha đẻ của ngôn ngữ Pascal, đã đề xuất ra việc xây dựng các sơ đồ mô tả tóm tắt các cú pháp của ngôn ngữ lập trình Pascal. Sơ đồ cú pháp (syntax diagram) sẽ giúp ta nắm được các dạng thức của những yếu tố cấu trúc chương trình. Một số các phần tử trong sơ đồ có ý nghĩa sau:

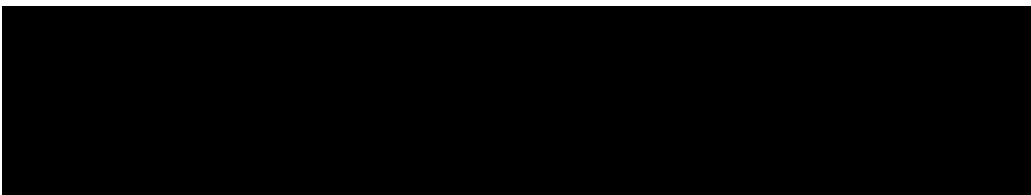
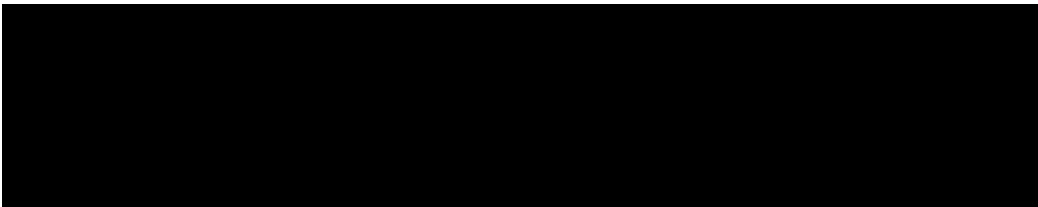
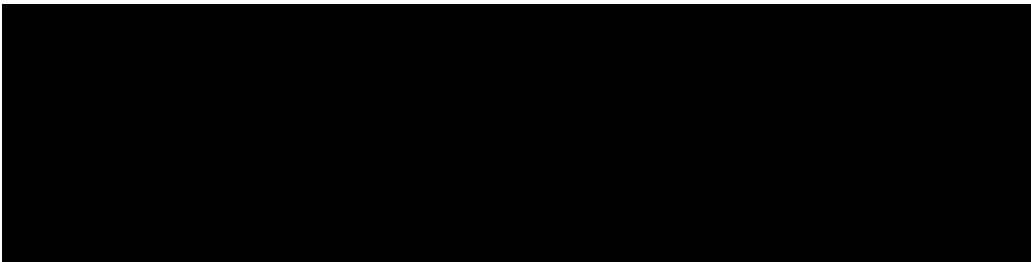
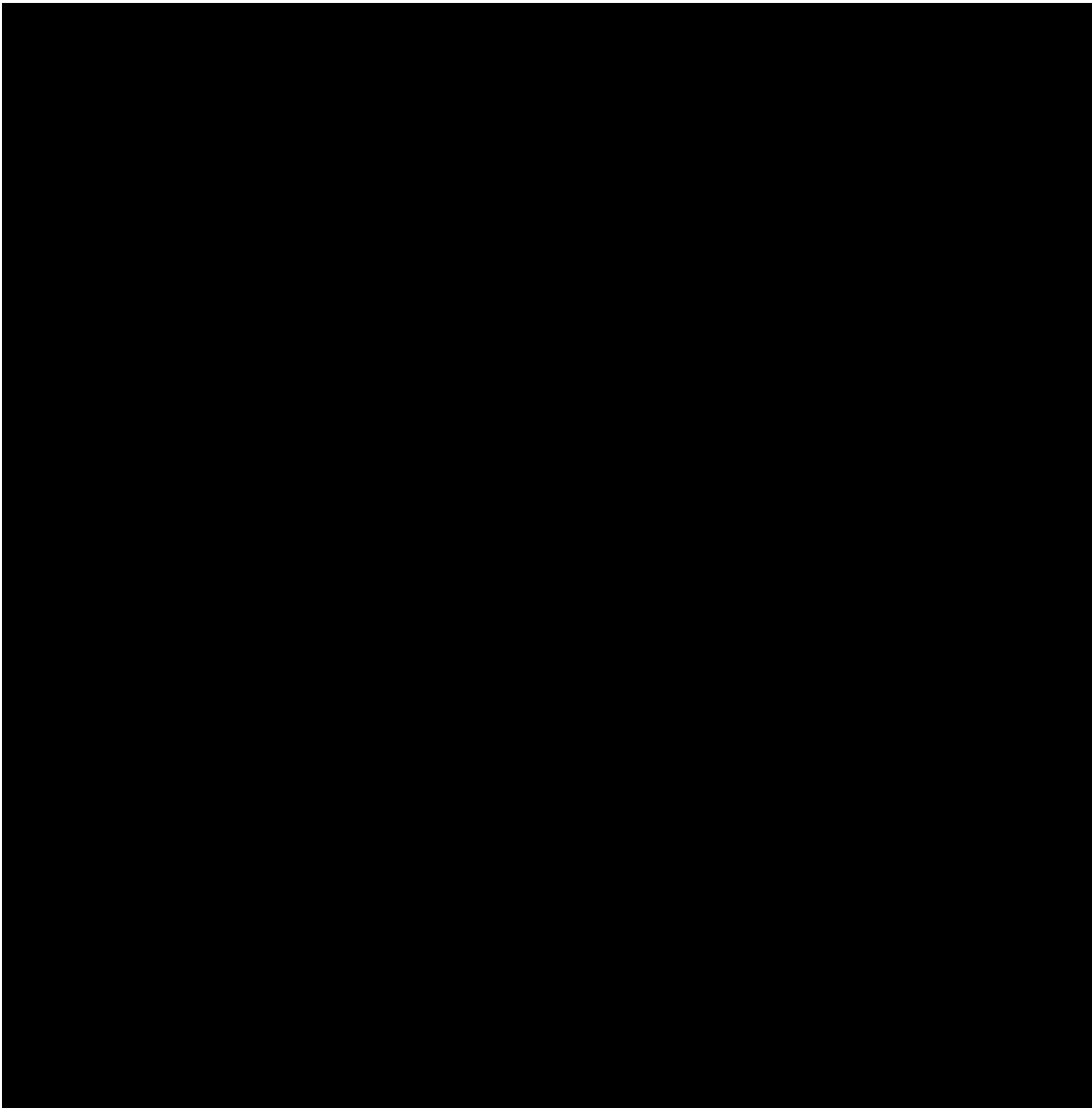


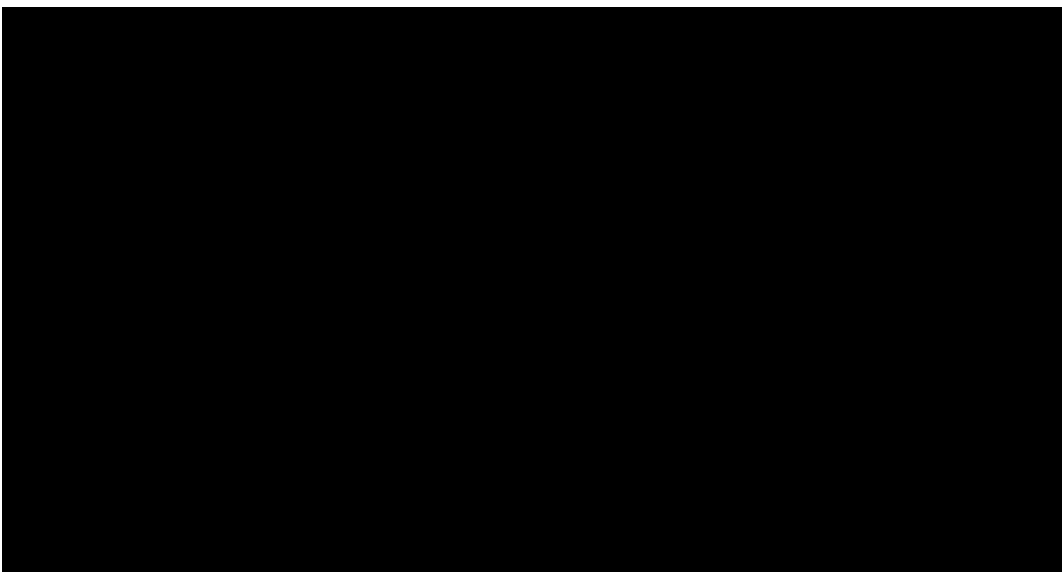
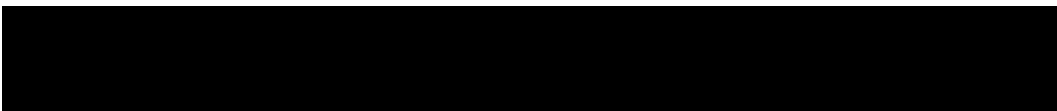
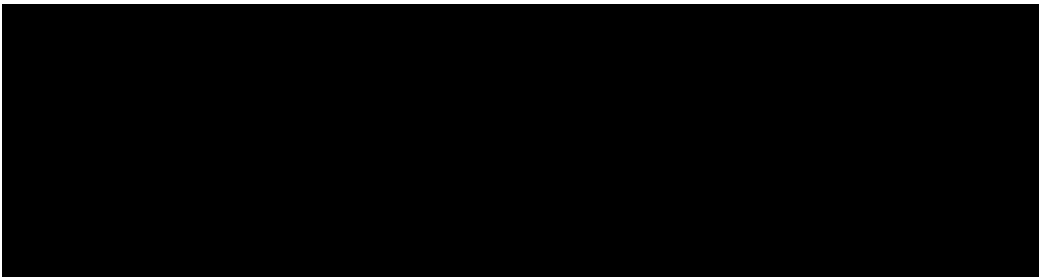
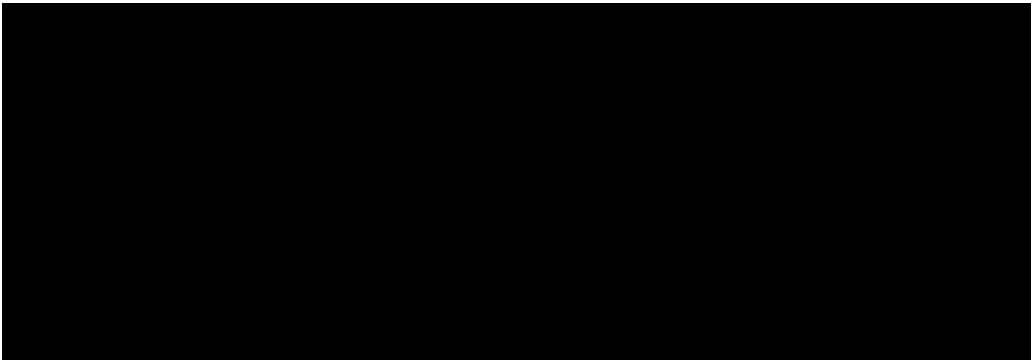
Ở sơ đồ tên (danh hiệu) phải bắt đầu bằng một chữ cái, sau đó có thể có nhiều chữ, nhiều số và gạch dưới.

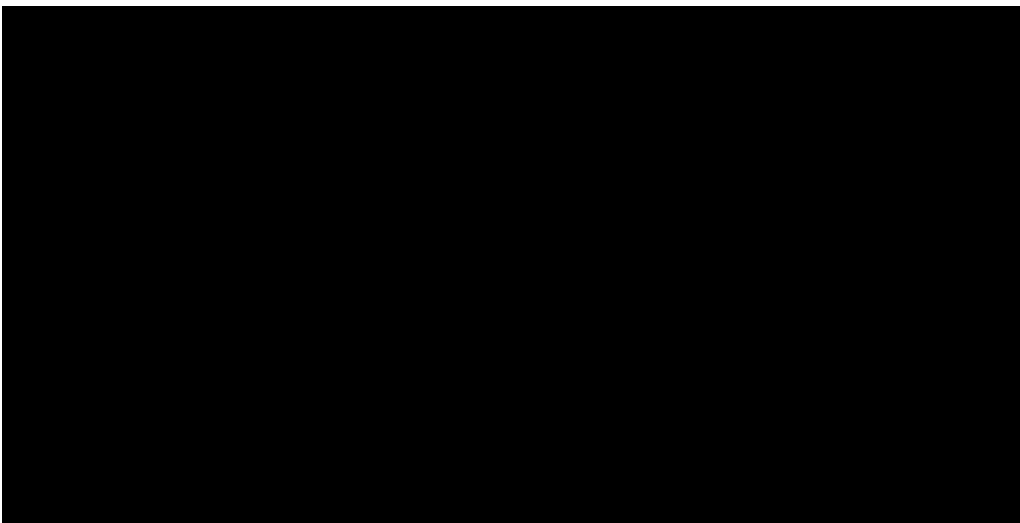
Ở sơ đồ biến, một biến bất kỳ có thể là biến có chỉ số hay không có chỉ số. Mọi biến đều có thể nhận diện qua một tên được đặt theo qui tắc đặt tên, nếu có chỉ số thì mỗi chỉ số có dạng một biểu thức, cách nhau bằng dấu phẩy ',' và tất cả chỉ số được đặt trong dấu ngoặc vuông '[' và ']'

### I. SƠ ĐỒ CẤU TRÚC CĂN BẢN









### **TẠM KẾT**

Đến đây, các bạn sinh viên đã đi qua một chặng đầu cơ bản nhưng rất quan trọng trong việc trang bị cho mình một kiến thức cần thiết trong khoa học máy tính. Bài giảng này, dĩ nhiên do giới hạn về thời lượng học tập, không thể trình bày hết các chi tiết của từng phần và cần phải tiếp tục bổ sung nhất là phần thực hành. Nếu các bạn nắm vững các điều đã học thì các bạn có nhiều cơ may tự học hỏi, tìm tòi và khám phá nhiều điều thú vị khác.

Hiện nay, trên thị trường sách và thư viện ở Việt nam, các bạn có thể kiếm nhiều tài liệu bổ ích viết bằng tiếng Việt hoặc tiếng nước ngoài để bổ xung cho kiến thức của các bạn. Lĩnh vực tin học tuy mới mẻ ở Việt nam nhưng rất có nhiều hứa hẹn phát triển trong tương lai.

Chúc các bạn nhiều sức khoẻ và thành công.



SÁCH HƯỚNG DẪN HỌC TẬP

# TIN HỌC ĐẠI CƯƠNG

**Biên soạn:**

**PHAN THỊ HÀ**

**NGUYỄN TIẾN HÙNG**



TRƯỜNG ĐẠI HỌC CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG  
Kính gửi: Trường Nguyễn Trãi, Hà Đông-Hà Tây  
Website: <http://www.cpb.vn>; E-mail: [dhkc@pbv.vn](mailto:dhkc@pbv.vn)  
Số điện thoại: (04) 5541221; Fax: (04) 5540587

## GIỚI THIỆU MÔN HỌC

### 1. GIỚI THIỆU CHUNG:

Bài giảng “Tin học đại cương” được xây dựng theo chương trình đào tạo từ xa nhằm cung cấp cho sinh viên các lớp Đại học từ xa của Học viện công nghệ Bưu chính viễn thông; đồng thời cũng là tài liệu tham khảo bổ ích cho sinh viên Đại học và Cao đẳng đào tạo tại chỗ thuộc các chuyên ngành Kỹ thuật, chuyên ngành Quản trị kinh doanh.

Bài giảng “Tin học đại cương” đã bám sát nội dung cơ sở của đề cương chi tiết Bộ giáo dục và Đào tạo ban hành. Tuy nhiên, nhóm tác giả có hiệu chỉnh và cập nhật các phần nội dung ứng dụng mới của Công nghệ thông tin hiện nay. Bài giảng gồm 5 chương và phụ lục. Trong mỗi chương, nhóm tác giả có đưa ra những ví dụ minh họa, các câu hỏi và bài tập một cách hệ thống để giúp cho sinh viên nắm bắt được phần học lý thuyết và tiếp cận được với kiến thức thực tiễn.

Chương 1: Các khái niệm cơ bản. Chương này nhằm cung cấp cho sinh viên các kiến thức, các khái niệm cơ sở liên quan đến tin học, hệ thống máy tính trước khi bước vào các chương tiếp theo.

Chương 2: Hệ điều hành (HĐH). Chương này cung cấp cho sinh viên các khái niệm cơ bản, tổng quan về hệ điều hành, bên cạnh đó còn cung cấp cho sinh viên phương thức hoạt động, chức năng, nhiệm vụ của HĐH, cách sử dụng của một số hệ điều hành thông dụng như HĐH MS-DOS và HĐH WINDOWS.

Chương 3: Các phần mềm ứng dụng thông dụng. Phần này nhằm cung cấp cho sinh viên các kỹ năng cơ bản trong việc sử dụng một số ứng dụng thông dụng như Microsoft Word, Excel, Power Point và Virus tin học. Qua đó, sinh viên có khả năng sử dụng phần tin học văn phòng để làm công cụ phục vụ cho các công việc văn phòng và đặc biệt là các môn học tiếp theo có sử dụng tin học văn phòng. Bên cạnh đó sinh viên hiểu được tác hại của Virus cũng như cách phòng chống Virus.

Chương 4 (Chủ yếu dành cho các ngành ĐTVT và CNTT): Ngôn ngữ lập trình C. Chương này cung cấp cho sinh viên các kiến thức tổng quan và cơ bản về ngôn ngữ lập trình C. Qua đó sinh viên có thể nắm được các khái niệm cơ bản về lập trình và thiết lập được một số chương trình đơn giản phục vụ cho

khoa học kỹ thuật và đặc biệt là làm công cụ để phục vụ cho các môn học về tin học và viễn thông mà các em sắp học.

Chương 5 (Chủ yếu dành cho ngành QTKD): Hệ quản trị cơ sở dữ liệu (CSDL) Microsoft Access. Chương này cấp cho sinh viên kiến thức tổng quan về hệ quản trị CSDL và cụ thể là hệ quản trị CSDL Microsoft Access, cách thức tổ chức dữ liệu trên hệ quản trị CSDL cũng như các tính năng và các công cụ mạnh của Microsoft Access. Qua đó sinh viên nắm được tất cả các kỹ năng cơ bản cần có để xây dựng và sử dụng CSDL trên phần mềm Microsoft ACCESS.

## 2. MỤC ĐÍCH MÔN HỌC

Học phần này cung cấp cho sinh viên các kiến thức cơ bản về tin học, cấu trúc máy tính, hệ điều hành, các chương trình ứng dụng thông dụng, ngôn ngữ lập trình C.

## 3. PHƯƠNG PHÁP NGHIÊN CỨU MÔN HỌC

Để học tốt môn học này, sinh viên cần lưu ý những vấn đề sau :

### 1- Thu thập đầy đủ các tài liệu :

◇ Bài giảng: *Tin học đại cương*, Phan Thị Hà, Nguyễn Tiến Hùng, Học viện Công nghệ BCVT, 2005.

◇ Sách hướng dẫn học tập và bài tập: *Tin học đại cương*, Phan Thị Hà, Nguyễn Tiến Hùng, Học viện Công nghệ BCVT, 2005.

◇ Bài giảng điện tử: *Tin học đại cương*, Học viện Công nghệ BCVT, 2005.

*Nếu có điều kiện, sinh viên nên tham khảo thêm:* Các tài liệu tham khảo trong mục Tài liệu tham khảo ở cuối cuốn sách này.

### 2- Đặt ra mục tiêu, thời hạn cho bản thân:

✓ *Đặt ra mục các mục tiêu tạm thời và thời hạn cho bản thân, và cố gắng thực hiện chúng*

Cùng với lịch học, lịch hướng dẫn của Học viện của môn học cũng như các môn học khác, sinh viên nên tự đặt ra cho mình một kế hoạch học tập cho riêng mình. Lịch học này mô tả về các tuần học (tự học) trong một kỳ học và đánh dấu số lượng công việc cần làm. Đánh dấu các ngày khi sinh viên phải thi sát hạch, nộp các bài luận, bài kiểm tra, liên hệ với giảng viên.

✓ *Xây dựng các mục tiêu trong chương trình nghiên cứu*

Biết rõ thời gian nghiên cứu khi mới bắt đầu nghiên cứu và thử thực hiện, cố định những thời gian đó hàng tuần. Suy nghĩ về thời lượng thời gian nghiên

cứu để “*Tiết kiệm thời gian*”. “*Nếu bạn mất quá nhiều thì giờ nghiên cứu*”, bạn nên xem lại kế hoạch thời gian của mình.

### **3- Nghiên cứu và nắm những kiến thức đề cốt lõi:**

Sinh viên nên đọc qua sách hướng dẫn học tập trước khi nghiên cứu bài giảng môn học và các tài liệu tham khảo khác. Nên nhớ rằng việc học thông qua đọc tài liệu là một việc đơn giản nhất so với việc truy cập mạng Internet hay sử dụng các hình thức học tập khác.

Hãy sử dụng thói quen sử dụng bút đánh dấu dòng (highline maker) để đánh dấu các đề mục và những nội dung, công thức quan trọng trong tài liệu.

### **4- Tham gia đầy đủ các buổi hướng dẫn học tập:**

Thông qua các buổi hướng dẫn học tập này, giảng viên sẽ giúp sinh viên nắm được những nội dung tổng thể của môn học và giải đáp thắc mắc; đồng thời sinh viên cũng có thể trao đổi, thảo luận của những sinh viên khác cùng lớp. Thời gian bố trí cho các buổi hướng dẫn không nhiều, do đó đừng bỏ qua những buổi hướng dẫn đã được lên kế hoạch.

### **5- Chủ động liên hệ với bạn học và giảng viên:**

Cách đơn giản nhất là tham dự các diễn đàn học tập trên mạng Internet. Hệ thống quản lý học tập (LMS) cung cấp môi trường học tập trong suốt 24 giờ/ngày và 7 ngày/tuần. Nếu không có điều kiện truy nhập Internet, sinh viên cần chủ động sử dụng hãy sử dụng dịch vụ bưu chính và các phương thức truyền thông khác (điện thoại, fax,...) để trao đổi thông tin học tập.

### **6- Tự ghi chép lại những ý chính:**

Nếu chỉ đọc không thì rất khó cho việc ghi nhớ. Việc ghi chép lại chính là một hoạt động tái hiện kiến thức, kinh nghiệm cho thấy nó giúp ích rất nhiều cho việc hình thành thói quen tự học và tư duy nghiên cứu.

### **7 -Trả lời các câu hỏi ôn tập sau mỗi chương, bài.**

Cuối mỗi chương, sinh viên cần tự trả lời tất cả các câu hỏi. Hãy cố gắng vạch ra những ý trả lời chính, từng bước phát triển thành câu trả lời hoàn thiện.

Đối với các bài tập, sinh viên nên tự giải trước khi tham khảo hướng dẫn, đáp án. Đừng ngại ngần trong việc liên hệ với các bạn học và giảng viên để nhận được sự trợ giúp.

***Nên nhớ thói quen đọc và ghi chép là chìa khoá cho sự thành công của việc tự học!***

## Chương 1: CÁC KHÁI NIỆM CƠ BẢN

### 1.1. GIỚI THIỆU

Chương này cung cấp cho sinh viên các kiến thức sau:

- Các khái niệm cơ bản về thông tin, các hệ đếm và mệnh đề logic trong lĩnh vực tin học.
- Tổng quan về quá trình xử lý thông tin, nhận diện kỹ thuật phần cứng và phần mềm được nghiên cứu trong tin học.
- Các khái niệm cơ bản về thuật toán và sơ đồ khối để giải quyết một bài toán cụ thể trên máy tính điện tử.
- Cấu trúc của một hệ thống máy tính, phần cứng, phần mềm, các thiết bị ngoại vi.
- Các khái niệm về mạng, kết nối mạng, các cấu trúc liên kết mạng, các thành phần thiết bị đầu nối mạng và các dịch vụ cơ bản của mạng máy tính.

### 1.2. TÓM TẮT CHƯƠNG I

#### 1.2.1. Khái niệm thông tin

Thông tin là một khái niệm trừu tượng, nó được thể hiện dưới nhiều dạng thức khác nhau. Thông tin có thể được phát sinh, được lưu trữ, được biến đổi trong những vật mang tin (gọi là giá). Thông tin được biến đổi bởi các dữ liệu và các dữ liệu này có thể được truyền đi, được sao chép, được xử lý hoặc bị phá hủy. Thông tin được thể hiện bằng các dạng tín hiệu vật lý.

#### 1.2.2. Xử lý thông tin

*Sơ đồ tổng quát của một quá trình xử lý thông tin*



### 1.2.3. Tin học

Tin học là một ngành khoa học công nghệ nghiên cứu các phương pháp xử lý thông tin một cách tự động dựa trên các phương tiện kỹ thuật mà chủ yếu hiện tại là máy tính điện tử. Tin học hiện đang được ứng dụng rộng rãi trong tất cả các ngành nghề khác nhau của xã hội từ khoa học kỹ thuật, y học, kinh tế, công nghệ sản xuất đến khoa học xã hội, nghệ thuật

### 1.2.4. Hệ đếm

Hệ đếm là tập hợp các ký hiệu và qui tắc sử dụng tập ký hiệu đó để biểu diễn và xác định các giá trị các số. Mỗi hệ đếm có một số ký số (digits) hữu hạn và tổng số ký số của mỗi hệ đếm được gọi là cơ số (base hay radix), ký hiệu là  $b$ .

### 1.2.5. Cấu trúc tổng quát của máy tính điện tử

Mỗi loại máy tính có thể có các hình dạng hoặc cấu trúc khác nhau tùy theo mục đích sử dụng. Tuy nhiên, một máy tính muốn hoạt động được phải hội tụ đủ các yếu tố sau :

- Phần cứng: bao gồm các thiết bị vật lý mà người dùng có thể quan sát được. Đó là các bảng mạch điện tử được lắp ghép lại với nhau và được cung cấp điện năng để hoạt động. Phần cứng máy tính thường được chia ra làm ba phân cơ bản - đó là: Thiết bị nhập, thiết bị xử lý và thiết bị xuất

- Phần mềm: bao gồm các chương trình được viết bởi các nhà lập trình nhằm mục đích điều khiển các mạch điện tử cũng như thực hiện các phép tính toán. Phần mềm thường chia làm ba loại cơ bản - đó là: Hệ điều hành, phần mềm ứng dụng và phần mềm tiện ích.

### 1.2.6. Khái niệm về mạng máy tính

Ở mức độ đơn giản nhất, mạng (network) bao gồm hai máy tính nối với nhau bằng cáp (cable) theo một chuẩn nào đó sao cho chúng có thể dùng chung dữ liệu. Mọi mạng máy tính dù phức tạp đến đâu đi nữa cũng đều bắt nguồn từ hệ thống đơn giản đó.

## 1.3. CÂU HỎI VÀ BÀI TẬP

**Câu 1:** Hãy xác định mã nhị phân nào sau đây là kết quả được chuyển đổi từ mã số thập phân  $29.75_{(10)}$  :

a/  $01101.01_{(2)}$

b/  $11011.11_{(2)}$

c/  $11101.11_{(2)}$

d/ Tất cả đều sai

**Câu 2:** Hãy xác định mã nhị phân nào sau đây là kết quả được chuyển đổi từ mã số hexa-deximal  $3C4_{(16)}$  :

- a/  $001011000110_{(2)}$
- b/  $001111000100_{(2)}$
- c/  $010010100010_{(2)}$
- d/ Tất cả đều sai

**Câu 3:** Hãy xác định kết quả đúng của phép tính cộng 2 số nhị phân  $0101 + 1100$  :

- a/ 10011
- b/ 10001
- c/ 11001
- d/ 10101

**Câu 4:** Hãy xác định kết quả đúng của phép tính nhân 2 số nhị phân  $0110 * 1011$  :

- a/ 1001001
- b/ 1100101
- c/ 1000110
- d/ 1000010

**Câu 5:** Trong phần cứng máy tính, những thiết bị nào sau đây thuộc nhóm thiết bị đầu vào (Input) :

- a/ Keyboard, Mouse, Light Pen
- b/ Keyboard, Scanner, Digitizer
- c/ Mouse, Light Pen, Touch screen
- d/ Tất cả đều đúng

**Câu 6:** Trong phần cứng máy tính, những thiết bị nào sau đây thuộc nhóm thiết bị đầu ra (Output) :

- a/ Printer, RAM Disk, Plotter
- b/ Printer, Scanner, Plotter
- c/ Monitor, Plotter, Printer
- d/ Tất cả đều sai

**Câu 7:** Phần mềm AntiVirus được cài đặt trong máy tính thuộc nhóm phần mềm nào sau đây

- a/ Phần mềm ứng dụng
- b/ Phần mềm hệ thống
- c/ Cả hai nhóm phần mềm trên.
- d/ Tất cả đều sai.

**Câu 8:** Hệ điều hành mạng nào sau đây cho phép thực hiện kiểu mạng ngang hàng :

- a/ Windows 2000/NT
- b/ UNIX.
- c/ LINUX
- d/ Novell Netware

**Câu 9:** Hệ điều hành mạng nào sau đây cho phép thực hiện kiểu mạng Client/Server :

- a/ Windows NT Server
- b/ UNIX
- c/ Novell Netware
- d/ Tất cả đều đúng

**Câu 10:** Máy tính PC (Personal Computer) mà chúng ta đang sử dụng hiện nay thuộc phân loại máy tính nào sau đây.

- a/ Microcomputers
- b/ Minicomputers
- c/ Mainframe computers
- d/ Supercomputers

**Câu 11:** Trong cấu trúc liên kết (Topology) mạng máy tính cục bộ (LAN) có những kiểu cơ bản nào sau đây :

- a/ Ring Topology và Bus Topology
- b/ Ring Topology và Star Topology
- c/ Bus Topology, Star Topology và Ring Topology
- d/ Bus Topology, Star Topology, Ring Topology và Mesh Topology



**Câu 12:** Hãy xác định thiết bị liên kết mạng HUB hoạt động ở tầng nào trong mô hình tham chiếu OSI

- a/ Tầng vật lý
- b/ Tầng liên kết dữ liệu
- c/ Tầng mạng
- d/ Tầng vận chuyển

**Câu 13:** Hãy xác định thiết bị liên kết mạng Bridge hoạt động ở tầng nào trong mô hình tham chiếu OSI

- a/ Tầng vật lý
- b/ Tầng liên kết dữ liệu
- c/ Tầng mạng
- d/ Tầng vận chuyển

**Câu 14:** Hãy xác định thiết bị liên kết mạng Router hoạt động ở tầng nào trong mô hình tham chiếu OSI

- a/ Tầng vật lý
- b/ Tầng liên kết dữ liệu
- c/ Tầng mạng
- d/ Tầng vận chuyển

**Câu 15:** Giao thức nào sau đây được sử dụng chung trong tất cả các hệ điều hành mạng hiện nay :

- a/ UUCP
- b/ TCP/IP
- c/ NetBEUI
- d/ Tất cả đều sai.

**Câu 16:** Giao thức nào sau đây được sử dụng dành riêng trong các hệ điều hành mạng của Microsoft :

- a/ IPX/SPX
- b/ TCP/IP
- c/ NetBEUI
- d/ Tất cả đều đúng.

**Câu 17:** Theo giá trị mặc định của IP, địa chỉ cổng 80 được gán sử dụng cho giao thức Internet nào sau đây

- a/ HTTP
- b/ FTP
- c/ SMTP
- d/ TELNET

**Câu 18:** Theo giá trị mặc định của IP, địa chỉ cổng 23 được gán sử dụng cho giao thức Internet nào sau đây

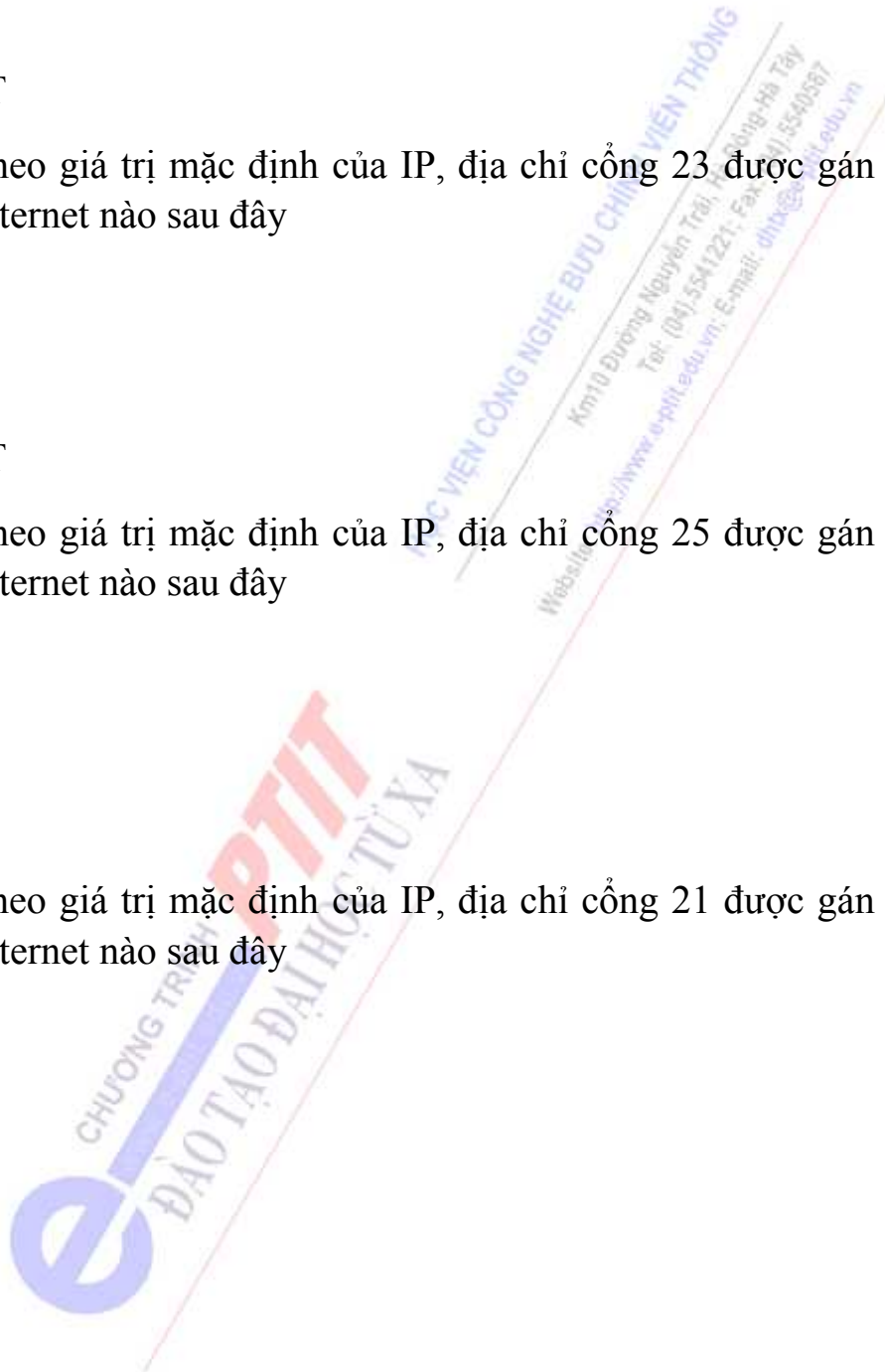
- a/ HTTP
- b/ FTP
- c/ SMTP
- d/ TELNET

**Câu 19:** Theo giá trị mặc định của IP, địa chỉ cổng 25 được gán sử dụng cho giao thức Internet nào sau đây

- a/ HTTP
- b/ FTP
- c/ SMTP
- d/ POP

**Câu 20:** Theo giá trị mặc định của IP, địa chỉ cổng 21 được gán sử dụng cho giao thức Internet nào sau đây

- a/ HTTP
- b/ FTP
- c/ SMTP
- d/ TELNE



## Chương 2: HỆ ĐIỀU HÀNH

### 2.1. GIỚI THIỆU

Chương này cung cấp cho sinh viên các kiến thức sau:

- Khái niệm và phân loại hệ điều hành.
- Giới thiệu về hệ điều hành (HĐH) MS-DOS, các lệnh cơ bản của HĐH MS-DOS.
- HĐH Windows, các khái niệm, môi trường làm việc, chi tiết về hướng dẫn sử dụng Windows.
- Giới thiệu các hệ điều hành khác.

### 2.2. TÓM TẮT CHƯƠNG 2

#### 2.2.1. Khái niệm và phân loại hệ điều hành

- Khái niệm: hệ điều hành là một tập hợp các chương trình lo việc điều khiển hoạt động của máy tính và tạo môi trường để các phần mềm khác chạy được.
- Phân loại hệ điều hành: Người ta phân loại hệ điều hành theo khả năng thực hiện cùng lúc một hay nhiều chương trình hoặc khả năng quản lý một hay nhiều máy tính.

Theo tiêu chuẩn thứ nhất ta có 2 loại hệ điều hành: HĐH đơn nhiệm, HĐH đa nhiệm.

Theo tiêu chuẩn thứ 2 ta có: Hệ điều hành cho máy đơn lẻ ,hệ điều hành mạng

#### 2.2.2. Hệ điều hành MS-Dos

##### 2.2.2.1. Các thành phần cơ bản của MS-DOS

- a) Boot record
- b) Tập hệ thống IO.sys
- c) Tập hệ thống MSDOS.SYS:
- d) Chương trình COMMAND.COM và các lệnh nội trú
- e) Lệnh ngoại trú

### 2.2.2.2. Các lệnh cơ bản của MS-DOS

#### a) Lệnh nội trú (internal command)

Tập lệnh nội trú gồm các lệnh chính thường dùng như:

- Các lệnh liên quan đến thư mục: DIR, CD, MD, RD, PATH, TREE, ...
- Các lệnh liên quan đến file: TYPE, DEL, COPY, REN, ...
- Các lệnh thời gian: TIME, DATE
- Các lệnh khác: PROMPT, CLS, VER, VOL, ...

#### b. Lệnh ngoại trú (external command)

Các lệnh ngoại trú như:

- Các lệnh liên quan đến ổ đĩa: FORMAT, LABEL, CHKDSK, DISKCOPY, ...
- Các lệnh liên quan đến tổ chức hệ thống: SYS, TREE, DELTREE, ...
- Các lệnh liên quan đến tập tin: UNDELETE, ATTRIB, FIND, ...
- Các lệnh khác: PRINT, GRAPHICS, SORT, ...

### 2.2.3. Hệ điều hành WINDOWS

#### 2.2.3.1. Giới thiệu về hệ điều hành WINDOWS:

Là hệ điều hành đa nhiệm, có tính năng giao diện người-máy bằng âm thanh, đồ họa, trang bị nhiều chức năng cửa sổ, sử dụng các trình đơn kéo xuống và con chuột, có các kiểu chữ đa dạng, các dụng cụ văn phòng (đồng hồ, máy tính, lịch, sổ ghi chép...).

#### 2.2.3.2. Cơ bản về cách cài đặt HĐH WINDOWS

Có thể cài đặt Windows 2000 theo hai cách: Nâng cấp từ Windows 95 hay Windows 98 lên Windows 2000 (chương trình Setup tự động chuyển các thiết lập hệ thống trước đây và các chương trình ứng dụng cũ đã cài đặt vào Windows 2000), cài Windows 2000 hoàn toàn mới (máy tính trước đây và hệ điều hành không phải là Windows, đĩa cứng hoàn toàn mới, các chương trình ứng dụng cũng phải cài lại từ đầu).

#### 2.2.3.3. Sử dụng MENU START và thanh TASKBAR

##### - Menu Start

Khi nháy nút Start xuất hiện menu Start:

Trong menu này có các mục:

- **Programs**: khi trỏ chuột vào mục này sẽ xuất hiện menu dọc cấp hai Programs chứa tất cả các chương trình có thể chạy.
- **Documents**: mở các tài liệu đã được mở gần đây nhất.
- **Settings**: sửa đổi màn hình nền hoặc các thiết lập hệ thống.
- **Search**: tìm tệp hoặc thư mục.
- **Help**: khi chọn mục này cửa sổ trợ giúp xuất hiện. Cửa sổ trợ giúp có 3 nút: Contents (hướng dẫn phân theo chủ đề), Index (các chủ đề hướng dẫn liệt kê theo thứ tự từ vựng), Search (tìm mọi chủ đề có chứa một từ hay câu, sau đó lựa chọn chủ đề cần xem hướng dẫn).
- **Run**: mở các mục như trang Web, chương trình và các tài nguyên máy tính khác.
- **Shut Down**: tắt hoặc khởi động lại máy.

#### 2.2.3.4. Windows Explorer

Windows Explorer là công cụ trợ giúp của Windows 2000 giúp ta biết được toàn bộ máy tính có những gì, cho phép xem nội dung máy tính như một hệ thống thứ bậc hay cấu trúc cây. Ta có thể xem dễ dàng nội dung mỗi ổ đĩa, mỗi thư mục, cũng như bất kỳ ổ đĩa nào trong mạng.

#### 2.2.3.5. Desktop và các biểu tượng

Các biểu tượng trên Desktop chia thành hai loại: biểu tượng mặc nhiên là biểu tượng có sẵn của Windows 2000 như My Computer và Recycle Bin, biểu tượng đường tắt (Shortcut) có dấu mũi tên ở góc dưới bên trái hình vẽ biểu tượng. Biểu tượng đường tắt cung cấp cách truy nhập vào các tài liệu và chương trình thường hay sử dụng một cách dễ dàng bằng cách nháy đúp lên biểu tượng.

##### - Biểu tượng My Computer

My Computer là rất hữu dụng nếu ta muốn xem nội dung riêng một thư mục hay một ổ đĩa. Khi ta nháy đúp My Computer trên màn hình nền, các ổ đĩa đang dùng hiện lên trong cửa sổ My Computer. Nháy đúp lên biểu tượng ổ đĩa, cửa sổ sẽ hiện lên các thư mục chứa trong ổ đĩa này. Sau đó ta có thể nháy đúp vào thư mục để xem các tệp có trong thư mục này.

## 2.2.4. Hệ điều hành LINUX

### 2.2.4.1. Các đặc điểm cơ bản

Hệ điều hành UNIX có một số đặc điểm sau:

- Đa chương
- Nhiều người sử dụng
- Bảo mật
- Độc lập phần cứng
- Kết nối mở
- Dùng chung thiết bị
- Tổ chức tập tin phân cấp

### 2.2.4.2. Các thành phần chính của hệ điều hành UNIX:

- Windows & Graphic User Interface
- Shell
- Lệnh và tiện ích
- Các bộ điều khiển thiết bị
- Kernel

## 2.3. CÂU HỎI VÀ BÀI TẬP

### Câu hỏi

1. Nêu khái niệm về hệ điều hành
2. Phân loại hệ điều hành
3. Hệ điều hành nào là HĐH đơn nhiệm?
  - DOS
  - WINDOWS 2000
  - WINDOWS XP
4. Hệ điều hành nào là HĐH đa nhiệm?
  - UNIX
  - WINDOWS 2000
  - DOS

5. Trong HĐH MS\_DOS đĩa hệ thống là đĩa chứa ít nhất những tệp nào?

- COMMAND.com
- IO.sys
- MSDOS.sys
- Cả 3 tệp trên

6. Định nghĩa nào là định nghĩa tệp tin:

□ là nơi lưu trữ thông tin bao gồm chương trình, dữ liệu, văn bản,...  
Mỗi tệp tin có một tên riêng phân biệt. Tên tệp tin thường có 2 phần: phần tên (name) và phần mở rộng (extension). Phần tên là bắt buộc phải có của một tệp tin, còn phần mở rộng thì có thể có hoặc không.

□ là nơi cất giữ các tệp tin theo một chủ đề nào đó theo ý người sử dụng. Đây là biện pháp giúp ta quản lý được tệp tin, dễ dàng tìm kiếm chúng khi cần truy xuất. Các tệp tin có liên quan với nhau có thể được xếp trong cùng một tệp tin.

□ là lệnh chỉ dẫn lộ trình cho phép ta từ thư mục bất kỳ có thể đến trực tiếp thư mục cần truy xuất.

7. Định nghĩa nào là định nghĩa đường dẫn:

□ là nơi lưu trữ thông tin bao gồm chương trình, dữ liệu, văn bản,...  
Mỗi tệp tin có một tên riêng phân biệt. Tên tệp tin thường có 2 phần: phần tên (name) và phần mở rộng (extension). Phần tên là bắt buộc phải có của một tệp tin, còn phần mở rộng thì có thể có hoặc không.

□ là nơi cất giữ các tệp tin theo một chủ đề nào đó theo ý người sử dụng. Đây là biện pháp giúp ta quản lý được tệp tin, dễ dàng tìm kiếm chúng khi cần truy xuất. Các tệp tin có liên quan với nhau có thể được xếp trong cùng một tệp tin.

□ là lệnh chỉ dẫn lộ trình cho phép ta (con trỏ) từ thư mục bất kỳ có thể đến trực tiếp thư mục cần truy xuất.

8. Muốn tạo một thư mục bằng lệnh của MSDOS thì ta dùng lệnh nào trong các lệnh sau?

- Copy con
- Rd

- Md
- DIR

9. Muốn tạo một tệp bằng lệnh của MSDOS thì ta dùng lệnh nào trong các lệnh sau :

- Copy con
- del
- Md
- DIR

10. Muốn tạo xóa một tệp bằng lệnh của MSDOS thì ta dùng lệnh nào trong các lệnh sau :

- Copy con
- del
- Md
- Rd

11. Muốn khởi tạo đĩa mới bằng lệnh của MSDOS thì ta dùng lệnh nào trong các lệnh sau :

- Format
- del
- Md
- Date

12. Muốn tạo đĩa khởi động tại ổ đĩa A: bằng lệnh của MSDOS thì ta dùng lệnh nào trong các lệnh sau :

- Format A:
- Format A:/s
- Format A:/a
- Sys A:/q

13. Muốn xóa một thư mục bằng lệnh của MSDOS thì ta dùng lệnh nào trong các lệnh sau :

- Copy con
- del



- Md
- Rd

14. Muốn đổi tên một tệp bằng lệnh của MSDOS thì ta dùng lệnh nào trong các lệnh sau

- rename
- del
- Md
- không lệnh nào đúng

15. Trong HĐH WINDOWS chương trình ứng dụng Control Panel dùng để:

- Cài đặt hay thay đổi cấu hình của hệ thống
- Xem nội dung của thư mục
- Chống phân mảnh đĩa cứng

16. Windows Explorer là công cụ trợ giúp, giúp ta biết được:

- Toàn bộ máy tính có những gì, cho phép xem nội dung của máy tính như một hệ thống thứ bậc hay cấu trúc hình cây. Cho phép xem nội dung mỗi ổ đĩa, mỗi thư mục, bất kỳ một ổ đĩa nào trong mạng

- Phép tạo và hiệu chỉnh các hình ảnh đồ họa: một bức vẽ trong Paint, thông tin đồ họa lưu trong Clipboard, một bức ảnh ghi ở dạng BMP, GIF, JPEG

- Các đối tượng bị xóa bỏ khi ta thao tác trong Windows 2000. Các tệp không thực sự bị loại khỏi đĩa cứng cho đến khi ta làm rỗng thùng rác.

17. Khi muốn di chuyển dữ liệu từ vùng nguồn đến vùng đích, ta phải sử dụng lệnh nào trong cặp lệnh nào dưới đây?

- Cut, Paste
- Copy, Paste
- File, Open

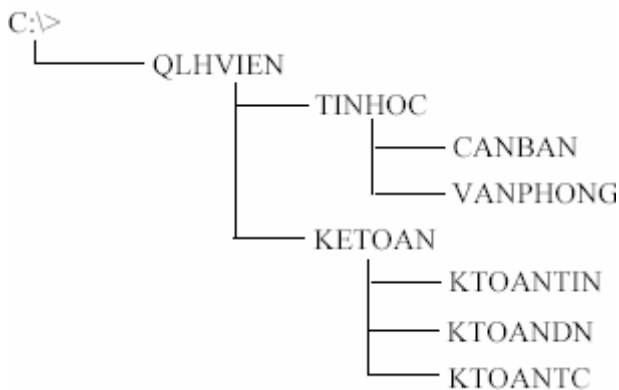
18. Khi muốn copy dữ liệu từ vùng nguồn đến vùng đích, ta phải sử dụng lệnh nào trong cặp lệnh nào dưới đây?

- Cut, Paste
- Copy, Paste
- Del, Paste

## Bài tập thực hành

### Bài 1. Dùng lệnh của MSDOS để thực hiện các yêu cầu sau:

1. Tạo cây thư mục sau:



2. Trong thư mục tin học, tạo 2 tập tin danh sách1.txt và file danh sách2.txt với nội dung tùy chọn

3. Nối 2 tập tin danh sách1.txt và danh sách2.txt thành danh sách.txt

4. Di chuyển tất các tệp vừa tạo ở câu 2,3 vào thư mục CANBAN

5. Đổi tên tệp danh sách.txt trong thư mục CANBAN thành ds.txt

6. Xóa màn hình và xem nội dung văn bản theo các cách sau:

Đứng tại thư mục gốc ( Ổ đĩa C)

Đứng tại thư mục KETOAN

Đứng tại thư mục CANBAN

7. Tạo thêm thư mục HA trong ổ đĩa C.Sao chép tất cả các tập tin trong thư mục CANBAN vào thư mục HA

8. Xem lại ngày, giờ hệ thống nếu sai thì sửa lại

9. Xóa tập tin danh sách1.txt, danh sách2.txt trong thư mục HA

10. Xem nội dung tập tin ds.txt trong thư mục HA bằng cách đứng tại thư mục QLHVIEN

11. Xóa thư mục KTOANTC

12. Copy tất cả các tệp có phần mở rộng là TXT từ thư mục gốc vào trong thư mục HA .

13. Xóa toàn bộ cây thư mục QLHVIEN

## **Bài 2: Thực hành các yêu cầu sau trên HĐH WINDOW2000**

Khởi động máy để vào Windows. Nháy vào nút Start để hiện menu Start, hãy duyệt qua tất cả các menu con của menu Start có trên máy.

Dùng lệnh Start / Help để mở cửa sổ trợ giúp, duyệt qua các hướng dẫn phân theo chủ đề (nháy vào các mục có cuốn sách để mở các mục con, nháy vào các mục có dấu chấm hỏi để xem chi tiết), xem các hướng dẫn xét theo thứ tự từ vựng. Đóng cửa sổ hướng dẫn.

Tạo một biểu tượng đường tắt cho Notepad trên màn hình nền với tên là Soan Thao Van Ban, nơi chứa chương trình Notepad là %SystemRoot%\System32\notepad.exe. Tạo biểu tượng đường tắt thứ hai cho Paint trên màn hình nền với tên là Ve Tranh, nơi chứa chương trình Paint là %System%\System32\mspaint.exe.

Nháy vào biểu tượng đường tắt Soan Thao Van Ban trên Desktop để mở cửa sổ Notepad, cực đại cửa sổ, soạn thảo một văn bản bất kỳ và ghi vào đĩa với tên tệp là Vidu1.txt. Nháy vào biểu tượng Ve Tranh trên Desktop để mở Paint, cực đại cửa sổ, vẽ một bức tranh tùy ý và ghi vào đĩa với tên tệp là Vidu2.bmp.

Chuyển đổi làm việc giữa hai cửa sổ bằng các nút trên thanh Taskbar. Cực tiểu cả hai cửa sổ về các nút trên thanh Taskbar.

Thiết lập màn hình nền: nháy nút phải chuột trên Desktop, hiện menu tắt, chọn Properties, hiện hộp thoại, chọn lớp Background, duyệt xem các ảnh nền trong khung Select a background picture, chọn một ảnh, OK.

Thiết lập màn hình chờ (màn hình hiện khi không dùng máy tính): nháy nút phải chuột trên Desktop hiện menu tắt, chọn Properties, hiện hộp thoại, chọn lớp Screen Saver, duyệt xem qua các màn hình chờ trong hộp điều khiển kéo xuống Screen Saver, chọn một màn hình chờ, trong mục Wait đặt là 5 phút, chọn OK.

Xem các thông tin hệ thống: Dùng lệnh Start / Programs / Accessories / System Tools / System Information.

Xem các phông chữ đã cài đặt trên máy: dùng lệnh Start / Settings / Control Panel / Fonts.

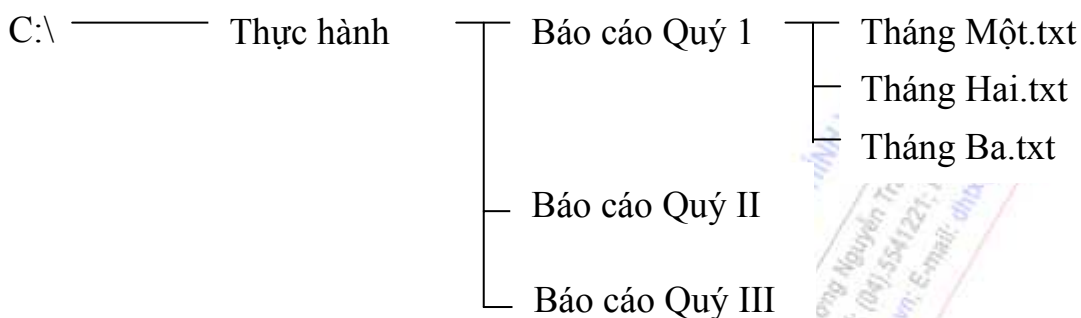
Đóng tất cả các cửa sổ đã mở. Thoát khỏi Windows 2000 và tắt máy tính.

## **Bài 3: Thực hành các yêu cầu sau trên HĐH WINDOW**

Khởi động Windows Explorer, trong khung bên trái của cửa sổ hãy nháy vào các thư mục có dấu cộng (+) để xem toàn bộ cây thư mục của ổ đĩa cứng C.

Trong khung bên trái chọn ổ đĩa C, duyệt qua 5 cách xem các tệp trong khung bên phải bằng lệnh View / Details, List, Large Icons, Small Icons, Thumbnails.

Tạo cây thư mục sau từ thư mục gốc C:



Sao chép hai tệp *Thang Mot.txt* và *Thang hai.txt* sang thư mục *Bao cao Quy II*. Di chuyển tệp *Thang ba.txt* sang thư mục *Bao cao Quy III*. Kiểm tra việc thực hiện.

Xóa thư mục thực hành cùng với các thư mục con. Đóng cửa sổ Windows Explorer.

#### Bài 4.

Khởi động Paint vẽ bức tranh bất kỳ lưu vào tệp tranh.bmp

Vẽ chữ ký của bạn và ghi tệp vào đĩa với tên là Bai3.bmp.

## Chương 3: CÁC PHẦN MỀM ỨNG DỤNG THÔNG DỤNG

### 3.1. GIỚI THIỆU

Chương này cung cấp cho sinh viên các kiến thức sau:

Các kỹ năng cơ bản trong việc sử dụng phần mềm ứng dụng soạn thảo văn bản Word.

Các thao tác cơ bản trong việc sử dụng và thực hiện các hàm của bảng tính Excel cũng như cách chèn hình ảnh, biểu đồ bảng tính.

Các cách cơ bản để tạo ra một trình diễn báo cáo bằng POWERPOINT, thiết lập cách hiển thị của các Slide cùng một số công cụ phụ trợ.

Khái niệm về VIRUS tin học và cách phòng chống VIRUS.

### 3.2. TÓM TẮT CHƯƠNG III

#### 3.2.1. Chương trình soạn thảo văn bản WORD

##### 3.2.1.1. Các thao tác soạn thảo cơ bản

*Soạn thảo văn bản bằng tiếng Việt:*

Trong Windows 2000 dùng bộ mã Unicode version 2 (mã hai byte) và có nhiều font cho phép hiển thị tiếng Việt có dấu như Times New Roman, Arial, Tahoma, Courier New. Muốn tạo văn bản tiếng Việt ta cần một chương trình bàn phím tiếng Việt để gõ tiếng Việt, có thể dùng VietKey.

##### 3.2.1.2. Các thao tác định dạng văn bản

a. Định dạng ký tự

- Định dạng bằng hộp thoại Font

Đối với một đoạn văn bản, ta có thể định dạng lại nó theo font mới, cỡ mới, kiểu mới. Trước tiên hãy chọn khối văn bản cần định dạng lại, dùng lệnh Format / Font, xuất hiện hộp thoại Font.

b. Định dạng Paragraph

Chức năng này cho phép thay đổi cách hiển thị của một đoạn văn bản. Chọn đoạn văn bản, dùng lệnh Format / Paragraph, xuất hiện hộp thoại sau gồm hai lớp: Indents and Spacing, Line and Page Breaks....

### 3.2.1.3. Định dạng trang và in ấn

#### a. Chọn cỡ giấy và đặt lề

Trước khi bắt tay vào soạn thảo một văn bản ta phải chọn cỡ giấy định in và đặt lề cho trang in bằng lệnh File / Page Setup, xuất hiện hộp thoại Page Setup

#### b. In văn bản

Muốn in văn bản dùng lệnh File / Print (hoặc nháy chuột vào nút Print trên thanh Standard)

### 3.2.1.4. Chèn hình ảnh và công thức toán

#### - Chèn thêm các ký tự đặc biệt

Để chèn thêm một ký tự đặc biệt (ví dụ &R), thực hiện theo trình tự như sau: đưa con trỏ văn bản đến nơi cần chèn, dùng lệnh Insert / Symbol

#### - Đánh công thức toán - Equation

Trước tiên ta đặt con trỏ nhập vào chỗ định chèn công thức, dùng lệnh Insert / Object, hiện hộp thoại Object, chọn lớp Create New, nháy đúp chuột vào mục Microsoft Equation 3.0. Khi đó tại điểm chèn xuất hiện khung để nhập công thức và cửa sổ nhỏ Equation

#### - Chèn tệp ảnh vào văn bản đang soạn

Dùng lệnh Insert / Picture / From File Tạo các dòng chữ nghệ thuật nhờ WordArt

## 3.2.2. Bảng tính EXCEL 2000

### 3.2.2.1. Định dạng dữ liệu số:

Dùng để trình bày lại vùng bảng tính như chọn kiểu thể hiện số liệu, chọn đơn vị tính thích hợp, thay đổi kiểu chữ, cỡ chữ, nhấn mạnh nội dung số liệu quan trọng

### 3.2.2.2. Định dạng dữ liệu chữ:

Dữ liệu chữ đã nhập trong các ô có thể định dạng lại theo các thành phần: Font (nét chữ), Font Style (ngghiêng, đậm,...), Size (kích thước chữ), Color (màu của chữ). Các bước thực hiện định dạng ký tự: chọn vùng dữ liệu để định dạng, thực hiện lệnh Format / Cells, xuất hiện hộp thoại Format Cells, chọn lớp Font (các mục cũng giống như trong Word 2000), chọn Font, Font Style, Size, Underline, Color thích hợp cho chữ.

### 3.2.2.3. Sử dụng công thức, các hàm:

Excel có một số hàm mẫu (Function Wizard) dùng tiện lợi và đơn giản, ví dụ công thức =A3+A4+A5+A6+A7 có thể thay bằng hàm SUM (A3:A7). Dạng thức tổng quát của hàm:

=<Tên hàm>(Danh sách đối số)

#### Các hàm số học

**ABS** (N), **SQRT** (N), **INT** (N), **PI** (), **MOD** (N,M), **ROUND** (biểu\_thức\_số, n), **SUM** (danh sách các trị), **AVERAGE** (danh sách các trị), **MAX** (danh sách các trị), **MIN** (danh sách các trị), **COUNTA** (danh sách các trị), **COUNT** (danh sách các trị), **RANK** (x, danh sách)

#### Các hàm ngày tháng

**DAY** (dữ liệu kiểu ngày), **MONTH** (dữ liệu kiểu ngày), **YEAR** (dữ liệu kiểu ngày)

#### Các hàm logic

**IF** (biểu thức logic, biểu thức 1, biểu thức 2): nếu biểu thức logic là đúng thì hàm cho giá trị là <biểu thức 1>, trái lại cho giá trị là <biểu thức 2>. Các **biểu thức 1** và **biểu thức 2** cũng có thể là những hằng trị, biểu thức (chuỗi, số, logic) và cũng có thể là một hàm IF khác.

**AND** (điều kiện 1, điều kiện 2, ...), **OR** (điều kiện 1, điều kiện 2, ...), **NOT** (điều kiện)

#### Các hàm xử lý chuỗi

**LEFT**(TEXT,N), **RIGHT**(TEXT,N), **LEN**(TEXT), **LOWER**(TEXT), **UPPER** (TEXT), **PROPER**(TEXT)

#### Các hàm tìm kiếm

**VLOOKUP** (x, Bảng, Cột tham chiếu, Cách dò), **HLOOKUP**(x, Bảng, Hàng tham chiếu, Cách dò).

### 3.2.3. Phần mềm trình diễn POWERPOINT 2000

PowerPoint dùng để tạo một trình diễn (Presentation) gây ấn tượng cho một bài nói trước đông đảo người nghe trong một hội nghị, một buổi bảo vệ luận án. Cách dùng hệ thống menu, dùng các thanh công cụ Standard và Formatting, dùng Office Assistant và nhận trợ giúp, chèn các đối tượng đồ họa

và hình ảnh, chèn Table trong PowerPoint được thực hiện giống như trong Word. Ta có thể dễ dàng sao chép văn bản trong Word và các biểu đồ trong Excel vào trong PowerPoint.

### **3.2.3.1. Tạo trình diễn dựa trên Slide trắng**

Các bước để tạo một trình diễn từ đầu:

**Bước 1:** Dùng lệnh File / New, hiện hộp thoại New Presentation:

**Bước 2:** Gõ tiêu đề của Slide và nội dung của nó. Chèn một slide trắng mới: nhấn nút Common Tasks trên thanh Formatting, hiện menu dọc, chọn mục New Slide (hoặc dùng lệnh Insert / New Slide, hay Ctrl+M), xuất hiện lại hộp thoại New Slide, chọn một mẫu slide, bổ xung nội dung cho nó.

**Bước 3:** Xem cách hiển thị slide đang thiết kế: nhấn nút Slide Show ở góc trái bên dưới cửa sổ PowerPoint. Trong lúc xem khi nhấn nút phải chuột vào màn hình, xuất hiện menu tắt gồm các mục: End Show (kết thúc xem), Next (xem trang tiếp), Previous (lùi về trang trước).

**Bước 4:** Ghi tệp vào đĩa dùng lệnh File / Save.

### **3.2.3.2. In các slide và các chú giải ra giấy**

Dùng lệnh File / Print, xuất hiện hộp thoại Print, trong mục Print What hãy lựa chọn đối tượng cần in.

### **3.2.4. Virus tin học**

Virus tin học hay còn gọi là virus máy tính (computer virus) là một đoạn chương trình ẩn, có kích thước nhỏ, có khả năng tự sao chép, sinh sôi nảy nở như một virus sinh học, lây lan nhanh và có thể gây nguy cơ phá hoại các phần mềm hệ của hệ điều hành, xóa một phần hay toàn bộ các tập tin dữ liệu, tạo ra các tập tin vô nghĩa chiếm lấy không gian bộ nhớ của máy, hoặc chỉ đơn thuần trêu chọc hay đe dọa suông người sử dụng.

## **3.3. CÂU HỎI VÀ BÀI TẬP**

### **Phần Word**

#### **I. Câu hỏi**

1. Khi đang soạn thảo trên phần mềm Word, nếu muốn ghi dữ liệu vào 1 tệp thì ta chọn File trên thanh Menu sau đó chọn:

- New
- Save



- Open

- Edit

2. Khi đang soạn thảo trên phần mềm Word, nếu muốn mở 1 tệp thì ta chọn File trên thanh menu sau đó chọn:

- New

- Save

- Open

- Edit

3. Khi đang soạn thảo trên phần mềm Word, muốn sao chép một khối ta phải chọn khối muốn sao chép sau đó chọn: Edit rồi tiếp tục chọn :

- Copy

- Cut

- Insert

- Select all

4. Khi đang soạn thảo trên phần mềm Word, muốn di chuyển một khối ta phải chọn khối muốn sao chép sau đó chọn: Edit rồi tiếp tục chọn :

- Copy

- Cut

- Insert

- Save

5. Sau khi đã thực hiện câu 7 hoặc câu 8 muốn dán văn bản vào vị trí con trỏ đang đứng ta chọn Edit rồi chọn:

- Paste

- Cut

- Insert

- Select all

6. Khi đang soạn thảo, muốn định dạng nhanh bằng thanh công cụ ta dùng thanh công cụ:

- Formatting

- Standard

Autotext

Clipboard

7. Khi muốn định dạng kí tự cho một đoạn văn bản đã được chọn ta chọn Format sau đó chọn :

Font

Paragraph

Tab

Drop Tab

8. Muốn thay đổi màu nền hoặc tạo khung cho một đoạn văn bản đã được chọn ta chọn Format sau đó chọn:

Borders and Shading

Tab Stop

Font

Clipboard

9. Trước khi in ta phải chọn cỡ giấy định in bằng cách chọn **file**, chọn **Page Setup**, rồi tiếp tục chọn:

Paper size

Margins

Page break

Normal view

10. Trước khi in nếu ta muốn đặt lề cho trang in trước hết ta làm bằng cách chọn **file**, chọn **Page Setup**, rồi tiếp tục chọn:

Layout

Margins

Page break

Normal view

11. Muốn chèn vào công thức toán học thì ta chọn Insert, chọn Object, chọn Creat new rồi tiếp tục chọn:

Microsoft Equation...

Media Clip

- Bitmap Image
- Symbole

12. Muốn in tất cả các trang trong văn bản trước hết ta làm bằng cách chọn **File**, chọn **Print** trong vùng **Page range** chọn

- All
- Curent page
- Pages
- Selection

## II. Bài thực hành

### Bài thực hành 1

1. Vào Word, mở tệp ThucHanh1.doc, soạn thảo Bảng lương sau:

#### BẢNG LƯƠNG TRƯỜNG ĐHDL CỬU LONG

| STT       | Họ và tên      | Đơn vị  | Lương   | Phụ cấp | Tổng số |
|-----------|----------------|---------|---------|---------|---------|
| 1         | Lê Mỹ Linh     | Du Lịch | 2910000 | 350000  |         |
| 2         | Nguyễn Anh Sơn | CNTT    | 2850000 | 240000  |         |
| 3         | Lê Thị Diệp    | Kinh Tế | 4800000 | 220000  |         |
| 4         | Bùi Thế Phùng  | CNTT    | 4870000 | 260000  |         |
| 5         | Trần Bình Minh | Du Lịch | 3000000 | 270000  |         |
| 6         | Đình Văn Quang | CNTT    | 2950000 | 250000  |         |
| 7         | Lý Phương Nam  | Kinh Tế | 3820000 | 350000  |         |
| 8         | Lưu Huyền Đức  | Kinh Tế | 2200000 | 300000  |         |
| 9         | Nông Đức Bình  | Du Lịch | 3700000 | 400000  |         |
| Tổng cộng |                |         |         |         |         |

2. Tính cột Tổng số phía bên phải Bảng, tính hàng Tổng cộng phía dưới Bảng cho các cột số. Đánh dấu khối toàn bộ bảng và sao chép thêm 3 bảng nữa, gọi là Bảng 2, Bảng 3 và Bảng 4. Sắp xếp Bảng 2 theo cột Tổng số giảm dần. Sắp xếp bảng 3 theo cột Họ và tên tăng dần. Sắp xếp Bảng 4 theo cột Đơn vị tăng dần.

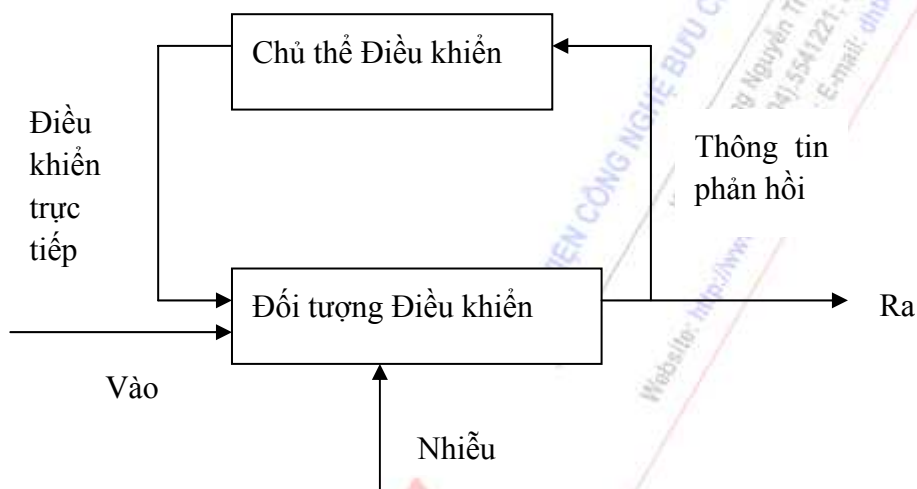
## Bài thực hành 2

Vào Word, mở một tệp mới có tên là ThucHanh2.doc và soạn thảo đoạn văn bản bất kỳ cùng với ảnh lấy từ Clip Art:

## Bài thực hành 3

Vào Word, tạo một tệp mới có tên là ThucHanh3.doc và soạn thảo, vẽ sơ đồ sau:

Vẽ sơ đồ sau của một quá trình điều khiển:



## Bài thực hành 4.

Vào Word và mở tệp mới có tên là ThucHanh4.doc. Gõ đoạn văn bản sau:

$$\|x^{(k)} - x^*\| \leq \frac{\mu}{1 - \mu} \|x^{(k)} - x^{(k-1)}\| \leq \frac{\mu^k}{1 - \mu} \|x^{(1)} - x^{(0)}\|$$

$$\underline{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \cdot & \cdot & \dots & \cdot \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}$$

## Phần Excel

### I. Câu hỏi

18. Khi đang trên bảng tính Excel, nếu muốn ghi dữ liệu vào 1 tệp thì ta chọn File trên thanh Menu sau đó chọn:

- New

- Save
- Open
- Edit

19. Khi đang làm việc trên bảng tính Excel, nếu muốn mở 1 tệp khác thì ta chọn File trên thanh menu sau đó chọn:

- New
- Save
- Open
- Edit

20. Một ô trong excel chỉ chứa một kiểu dữ liệu . Hãy liệt kê các kiểu dữ liệu cơ bản trong excel:

.....  
.....  
.....  
.....

21. Muốn xóa dữ liệu trong một vùng nào đó thì trước hết ta phải lựa chọn vùng cần xóa, sau đó gõ phím

- Insert
- Enter
- Del
- End

22. Muốn sao chép liệu từ vùng nguồn sang một vùng đích thì trước hết ta phải lựa chọn vùng nguồn, sau đó chọn Edit, rồi tiếp tục chọn

- Copy
- Cut
- Cancel
- Select

23. Sau khi đã làm câu 22, muốn dán dữ liệu vào vùng đích, ta phải di chuyển con trỏ vào đến vị trí của ô góc trên bên trái của vùng đích rồi chọn Edit, chọn:

- Copy
- Cut
- Select
- Paste

24. Muốn chèn dãy số tự động vào một hàng hay một cột thì trước hết gõ giá trị bắt đầu vào ô đầu tiên, chọn vùng cần điền số, thực hiện lệnh Edit chọn, chọn fill, chọn:

- Series
- Left
- Up
- Open

25. Khi muốn điền dữ liệu kiểu công thức hoặc một hàm vào một ô thì ta phải gõ dấu gì trước công thức, hàm:

- ?
- =
- 
- \*

26. Hàm nào trong các hàm dưới đây là hàm giá trị tuyệt đối của biểu thức số N

- ABS(N)
- INT (N)
- PI()
- MIN(N,M)

27. Hàm nào trong các hàm dưới đây là hàm căn bậc hai của số N

- SQRT(N)
- INT (N)

- ABS(N)
- MIN(N,M)

28. Hàm nào trong các hàm dưới đây là hàm tính tổng của các giá trị có trong danh sách

- MIN(Danh sách các trị)
- SUM(Danh sách các trị)
- ABS(N)
- MAX(Danh sách các trị)

29. Hàm nào trong các hàm dưới đây là hàm tìm giá trị nhỏ nhất của các giá trị có trong danh sách

- MIN(Danh sách các trị)
- SUM(Danh sách các trị)
- ABS(N)
- MAX(Danh sách các trị)

30. Hàm nào trong các hàm dưới đây là hàm tìm giá trị lớn nhất của các giá trị có trong danh sách

- MIN(Danh sách các trị)
- SUM(Danh sách các trị)
- AVERAGE(Danh sách các trị)
- MAX(Danh sách các trị)

31. Hàm nào trong các hàm dưới đây là hàm cho phần dư của phép chia nguyên N cho M

- MOD(N,M)
- SUM(Danh sách các trị)
- ABS(N)
- PI()

32. Hàm nào trong các hàm dưới đây là hàm cho giá trị tháng của dữ liệu kiểu ngày

- DAY(dữ liệu kiểu ngày)

- YEAR(dữ liệu kiểu ngày)
- MONTH (dữ liệu kiểu ngày)
- AND()

33. Hàm nào trong các hàm dưới đây là hàm cho giá trị năm của dữ liệu kiểu ngày

- DAY(dữ liệu kiểu ngày)
- YEAR(dữ liệu kiểu ngày)
- MONTH (dữ liệu kiểu ngày)
- AND()

34. Hàm nào trong các hàm sau đây là hàm cho độ dài của chuỗi TEXT

- LOWER(TEXT)
- UPPER(TEXT)
- PROPER(TEXT)
- LEN(TEXT)

## II. Bài tập thực hành

### Bài thực hành 6 (Excel):

Hãy nhập dữ liệu cho một hóa đơn bán hàng như sau:

|   | A          | B                | C               | D              | E                 |
|---|------------|------------------|-----------------|----------------|-------------------|
| 1 |            | HÓA ĐƠN          |                 |                |                   |
| 2 | <b>STT</b> | <b>Tên sách</b>  | <b>Số lượng</b> | <b>Đơn giá</b> | <b>Thành tiền</b> |
| 3 | 1          | Tin học          | 12              | 30000          | .....             |
| 4 | 2          | Chính trị        | 10              | 20000          | .....             |
| 5 | ...        |                  |                 |                |                   |
| 6 | ...        | <b>Tổng cộng</b> | .....           |                | .....             |

- Cột thành tiền được tính bằng công thức: Thành tiền= Đơn giá \* Số lượng
- Tính tổng cộng số lượng sách trong hóa đơn
- Tính tổng cộng Số tiền có trong hóa đơn



- Sắp xếp dữ liệu theo chiều tăng dần của **Tên sách**

**Bài thực hành 7:**

- Dùng Excel để tạo ra một tệp bangdiem.xls với nội dung như sau

**BẢNG ĐIỂM NĂM HỌC 2004**

| STT | HỌ TÊN        | TIẾNG ANH | TRIẾT HỌC | TIN HỌC | ĐIỂM TB | XẾP LOẠI |
|-----|---------------|-----------|-----------|---------|---------|----------|
|     | Hệ số môn học | 3         | 4         | 4       |         |          |
| 1   |               |           |           |         |         |          |
| 2   |               |           |           |         |         |          |
| 3   |               |           |           |         |         |          |
| ... |               |           |           |         |         |          |

- Nhập số liệu cho bảng tính trên, với điểm trung bình được tính theo hệ số môn học (ở phía dưới tên môn) và điền vào cột xếp loại theo quy định:

Điểm trung bình  $\geq 8.0$  đạt loại giỏi

$6.5 \leq$  Điểm trung bình  $< 8.0$ , Đạt loại Khá

$5.0 \leq$  Điểm trung bình  $< 6.5$ , Đạt loại Trung bình

Điểm trung bình  $< 5.0$ , Đạt loại Kém

**Bài thực hành 8:**

Nhập dữ liệu cho bảng tính thu nhập của một cửa hàng kinh doanh về tin học từ năm 2000-2003 như sau:

|          | A           | B                | C                |
|----------|-------------|------------------|------------------|
| <b>1</b> | <b>Năm</b>  | <b>Phần cứng</b> | <b>Phần mềm</b>  |
| <b>2</b> | <b>2000</b> | <b>500000000</b> | <b>97000000</b>  |
| <b>3</b> | <b>2001</b> | <b>570000000</b> | <b>80000000</b>  |
| <b>4</b> | <b>2002</b> | <b>600000000</b> | <b>100000000</b> |
| <b>5</b> | <b>2003</b> | <b>650000000</b> | <b>121000000</b> |

- Hãy tạo biểu đồ về thu nhập của phần cứng theo từng năm
- Hãy tạo biểu đồ về thu nhập của phần mềm theo từng năm

## Phần POWERPOINT

### I. Câu hỏi

35. In ra các slide gồm cả phần dòng ghi chú ra giấy bằng cách nào?

- Trong hộp thoại Print, chọn Handouts và chọn số slide trong một trang (Slides per page) là 3.
- Trong hộp thoại Print, chọn Handouts và số slide trong một trang, sau đó chọn Include comment pages.
- Trong hộp thoại Print, chọn Notes Pages thay vì chọn Handouts.

36. Bằng cách nào để hiển thị đối tượng của slide sao cho sinh động, hấp dẫn người xem?

### II. Bài thực hành

#### Bài thực hành 9 (Power Point).

Lập các Silde giới thiệu về bản thân. Slide 1: tiêu đề “Giới thiệu bản thân”, họ tên, ngày sinh, quê quán, dân tộc, địa chỉ nơi ở, số điện thoại. Slide 2: tiêu đề “Quá trình học tập”, phổ thông cơ sở, phổ thông trung học, đại học, cao học. Slide 3: tiêu đề “Quá trình công tác”, các nơi đã công tác. Slide 4: tiêu đề “Hoàn cảnh gia đình”, họ tên bố mẹ, anh chị em ruột.

## Chương 4: NGÔN NGỮ LẬP TRÌNH C

### 4.1. GIỚI THIỆU

Nội dung

Chương này cung cấp cho sinh viên các kiến thức sau:

- Một số kiến thức cơ sở về ngôn ngữ lập trình C
- Câu lệnh, các cấu trúc lệnh điều khiển
- Hàm và phạm vi hoạt động của biến
- Kiểu dữ liệu có cấu trúc: Kiểu mảng, kiểu chuỗi ký tự

Mục đích, yêu cầu:

Nhằm cung cấp cho sinh viên các kiến thức tổng quan và cơ bản về ngôn ngữ lập trình C. Qua đó học viên có thể nắm được các khái niệm cơ bản về lập trình và thiết lập được một số chương trình đơn giản phục vụ cho khoa học kỹ thuật và đặc biệt là làm công cụ để phục vụ cho các môn học về tin học và viễn thông mà các bạn sắp học.

### 4.2. TÓM TẮT CHƯƠNG IV

#### 4.2.1. Các bước cơ bản khi viết chương trình

Bước 1: Soạn thảo chương trình (dùng Turbo C)

Bước 2: Dịch và hiệu chỉnh chương trình (dùng turbo c)

Bước 3: Thực hiện chương trình

#### 4.2.2. Quá trình thực hiện các chương trình trong C

Thực hiện trình soạn thảo của Turbo C đó là TC.EXE, thông thường được đặt trong thư mục C:\TC\BIN.

Dịch chương trình bằng cách ấn phím F9, sau đó sửa lỗi nếu có thông báo

Dịch và thực hiện chương trình chỉ cần bấm tổ hợp phím CTRL + F9, sau đó sửa lỗi nếu có thông báo

Có thể xem kết quả bằng cách ấn tổ hợp ALT+F5

### 4.2.3. Các kiểu cơ sở dữ liệu

Một kiểu dữ liệu (Data Type) được hiểu là tập hợp các giá trị mà một biến thuộc kiểu đó có thể nhận được làm giá trị của biến cùng với các phép toán trên nó. Các kiểu dữ liệu cơ sở trong C bao gồm kiểu các số nguyên (int, long), kiểu số thực (float, double), kiểu kí tự (char).

Sau đây là bảng các giá trị có thể của các kiểu dữ liệu cơ bản của C:

| Kiểu          | Miền xác định                  | Kích thước |
|---------------|--------------------------------|------------|
| char          | 0.. 255                        | 1 byte     |
| int           | -32767 .. 32767                | 2 byte     |
| long          | -2147483648..2147483647        | 4 byte     |
| unsigned int  | 0 .. 65535                     | 2 byte     |
| unsigned long | 0..<br>2147483647*2=4294967295 | 4 byte     |
| float         | 3. 4e-38 .. 3.4e + 38          | 4 byte     |
| double        | 1.7e-308 .. 1.7e + 308         | 8 byte     |

### 4.2.4. Thủ tục Vào/ra chuẩn

Thủ tục vào ra chuẩn là các hàm đã được thiết lập sẵn trong thư viện vào ra chuẩn (stdio.h) dùng để đưa ra hoặc nhập vào giá trị của các biến... Một số hàm vào ra chuẩn hay sử dụng như:

Vào ra bằng getchar(), putchar()

In ra theo khuôn dạng - printf

Nhập vào có khuôn dạng - scanf

### 4.2.5. Tham nhập vào thư viện chuẩn

Mỗi tệp gốc có tham trở tới hàm thư viện chuẩn đều phải chứa dòng khai báo #include < tên\_tệp\_thư\_viện >

### 4.2.6. Biến, hằng, câu lệnh

- **Biến:** Biến là một đại lượng có giá trị thay đổi trong khi thực hiện chương trình. Mỗi biến có một tên và một địa chỉ của vùng nhớ dành riêng cho biến. Mọi biến đều phải khai báo trước khi sử dụng nó. Quy tắc khai báo một biến được thực hiện như sau:

Tên\_kiểu\_dữ\_liệu tên\_biến; trong trường hợp có nhiều biến có cùng kiểu, chúng ta có thể khai báo chung trên một dòng trong đó mỗi biến được phân biệt với nhau bởi một dấu phẩy và có thể gán giá trị ban đầu biến trong khi khai báo.

- **Hằng:** Hằng là đại lượng mà giá trị của nó không thay đổi trong thời gian thực hiện chương trình. C sử dụng chỉ thị #define để định nghĩa các hằng.

- **Câu lệnh:** Là phần xác định công việc mà chương trình phải thực hiện để xử lý các dữ liệu đã được mô tả và khai báo. Trong C các câu lệnh cách nhau bởi dấu chấm phẩy. câu lệnh được chia ra làm hai loại: Là câu lệnh đơn giản và câu lệnh có cấu trúc

Câu lệnh đơn giản là lệnh không chứa các lệnh khác, đó là phép gán, lệnh gọi hàm void

Câu lệnh có cấu trúc: Bao gồm nhiều lệnh đơn giản và có khi có cả lệnh cấu trúc khác bên trong ghép lại với nhau. Các lệnh loại này như :

- + Cấu trúc lệnh khối (lệnh ghép)
- + Lệnh if
- + Lệnh switch
- + Các lệnh lặp: for, while, do... while

#### 4.2.7. Hàm

Hàm (function) hay nói đúng hơn là chương trình con (sub\_program) chia cắt các nhiệm vụ tính toán lớn thành các công việc nhỏ hơn và có thể sử dụng nó ở mọi lúc trong chương trình, đồng thời hàm cũng có thể được cung cấp cho nhiều người khác sử dụng dưới dạng thư viện mà không cần phải bắt đầu xây dựng lại từ đầu. Các hàm thích hợp còn có thể che dấu những chi tiết thực hiện đối với các phần khác trong chương trình, vì những phần này không cần biết hàm đó thực hiện như thế nào.

##### - Khai báo, thiết kế hàm

Mọi hàm trong C dù là nhỏ nhất cũng phải được thiết kế theo nguyên tắc sau:

Kiểu\_hàm Tên\_hàm ( Kiểu\_1 biến\_1, Kiểu\_2 biến\_2, . . . )

{Khai báo biến cục bộ trong hàm;

Câu\_lệnh\_hoặc\_dãy\_câu\_lệnh;

return(giá\_trị);

}

Trước khi sử dụng hàm cần phải khai báo nguyên mẫu cho hàm (function prototype) và hàm phải phù hợp với nguyên mẫu của chính nó. Nguyên mẫu của hàm thường được khai báo ở phần đầu chương trình theo cú pháp như sau:

Kiểu\_hàm Tên\_hàm ( Kiểu\_1, Kiểu\_2 , . . . );

**- Phương pháp truyền tham biến cho hàm:**

Tên\_hàm ( tham biến 1 ,tham biến 2, . . . );

Cơ chế truyền cho hàm theo địa chỉ của biến được gọi là phương pháp truyền tham biến cho hàm. Nếu hàm được truyền theo tham biến thì nội dung của biến sẽ bị thay đổi sau khi thực hiện hàm.

Cơ chế truyền giá trị của biến cho hàm được gọi là phương pháp truyền theo tham trị. Nếu hàm được truyền theo tham trị thì nội dung của biến sẽ không bị thay đổi sau khi thực hiện hàm.

#### 4.2.8. Mảng

Mảng là một tập cố định các phần tử cùng có chung một kiểu dữ liệu với các thao tác tạo lập mảng (create), tìm kiếm một phần tử của mảng (retrieve), lưu trữ mảng (store). Ngoài giá trị, mỗi phần tử của mảng còn được đặc trưng bởi chỉ số của nó (index) thể hiện thứ tự của phần tử đó trong mảng. Không có các thao tác bổ sung thêm phần tử hoặc loại bỏ phần tử của mảng vì số phần tử trong mảng là cố định.

Một mảng gồm n phần tử mà mỗi phần tử của nó lại là một mảng k - 1 chiều thì nó được gọi là mảng k chiều. Số phần tử của mảng k chiều là tích số giữa số các phần tử của mỗi mảng một chiều.

Khai báo mảng một chiều được thực hiện theo qui tắc như sau:

Tên\_kiểu Tên\_biến[Số\_phần\_tử];

Cấu trúc lưu trữ của mảng: Mảng được tổ chức trong bộ nhớ như một vector, mỗi thành phần của vector được tương ứng với một ô nhớ có kích cỡ đúng bằng kích cỡ của kiểu phần tử và được lưu trữ kế tiếp nhau. Nếu chúng ta có khai báo mảng gồm n phần tử thì phần tử đầu tiên là phần tử thứ 0 và phần tử cuối cùng là phần tử thứ n - 1, đồng thời mảng được cấp phát một vùng không gian nhớ liên tục có số byte được tính theo công thức:

Kích\_cỡ\_mảng = ( Số\_phần\_tử \* sizeof(kiểu\_phần\_tử).

Truy nhập vào từng phần tử của mảng: Tên\_biến[i], với i là chỉ số phần tử đó trong mảng

- Xâu kí tự là một mảng trong đó mỗi phần tử của nó là một kí tự, kí tự cuối cùng của xâu được dùng làm kí tự kết thúc xâu. Kí tự kết thúc xâu được ngôn ngữ C qui định là kí tự '\0', kí tự này có mã là 0 (NULL) trong bảng mã ASCII.

### 4.3. CÂU HỎI VÀ BÀI TẬP

#### 1. Giả sử có khai báo như sau:

```
int n=10;p=4;
long q=2;
float x=1.75;
```

Hãy cho biết giá trị của mỗi biểu thức sau:

- n+q
- n+x
- n%p+q

#### 2. Cho đoạn chương trình

```
int x=5;
float y=9.0
float z;
z=y/x
```

Hãy chọn kết quả đúng của biết giá trị của z:

- 1
- 1.8
- 2
- Không câu nào ở trên là đúng

#### 3. Hãy chọn kết quả của phép tính: 23%3:

- 1
- 2
- 3
- 4

#### 4. Hãy cho biết kết quả của đoạn chương trình sau:

```
#include<stdio.h>
main()
```

```
{
 int n=20,p=10,q=5,t;
 t=n+p;
 printf("n=%d p=%d t=%d",n,p,t);
 n+=p; t-=n;
 printf("n=%d t=%d",n,t);
}
```

**Bài tập: Hãy viết các chương trình để**

1. Hiện câu chào
2. Hiện câu chào và chờ nhấn phím mới kết thúc
3. Nhập 2 số nguyên, tính tổng, hiệu, tích, thương của 2 số nguyên đó
4. Nhập 2 số thực, tính tổng, hiệu, tích, thương của 2 số thực đó
5. Nhập 3 số thực, tìm max của chúng
6. Liệt kê các số nguyên tố không lớn hơn số  $n$  cho trước
7. Liệt kê các số nguyên tố từ  $m$  đến  $n$
8. Tìm ước số chung lớn nhất của 2 số bất kỳ nhập vào từ bàn phím
9. Chuyển đổi 1 số nguyên thập phân sang dạng nhị phân
10. Đảo một chuỗi kí tự
11. Tìm số lớn nhất trong dãy số thực
12. Tìm xem 1 số thực  $x$  có xuất hiện trong dãy số thực hay không
13. Tính giá trị của đa thức bậc  $n$  theo phương pháp Horner
14. Loại trừ các dấu cách thừa trong chuỗi kí tự (chỉ để lại một dấu cách)
15. Đếm số chữ trong xâu kí tự
16. Tính số  $\pi$  thức công thức
17. Nhập ma trận A, ma trận B cấp  $n \times n$ , sau đó hãy hiển thị ra màn hình ma trận C là ma trận tổng của hai ma trận trên, ma trận D là tích của hai ma trận trên.



## Chương 5: HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU ACCESS

### 5.1. GIỚI THIỆU

Chương này cung cấp cho sinh viên các kiến thức sau:

Tổng quan về hệ quản trị cơ sở dữ liệu

Giới thiệu về hệ quản trị cơ sở dữ liệu Access

Các kiến thức cơ bản trong việc tạo lập cơ sở dữ liệu của Access, thiết kế xây dựng bảng (Table), mẫu biểu (Form), truy vấn (Query), Báo biểu (Report)

### 5.2. TÓM TẮT CHƯƠNG 5

#### 5.2.1. Tổng quan về hệ quản trị CSDL

Hệ quản trị CSDL là hệ thống phần mềm điều khiển toàn bộ các chiến lược truy nhập CSDL

Các chức năng cơ bản của hệ quản trị CSDL bao gồm

- Tạo ra và duy trì cấu trúc dữ liệu
- Cập nhật dữ liệu
- Lưu trữ dữ liệu
- Tìm kiếm và xử lý các dữ liệu lưu trữ
- Cho phép nhiều người dùng truy xuất đồng thời
- Hỗ trợ tính bảo mật và riêng tư
- Cung cấp một cơ chế chỉ mục (index) hiệu quả để lấy nhanh các dữ liệu lựa chọn.
- Bảo vệ dữ liệu khỏi mất mát bằng các quá trình sao lưu (backup) và phục hồi (recovery).

Hiện nay có một số hệ quản trị CSDL quan hệ được sử dụng rộng rãi trên thị trường như Oracle, Informix, Sybase, Foxpro, Access,.....

### 5.2.2. Hệ quản trị CSDL Access

Access là hệ quản trị CSDL, vì vậy bạn có thể dùng M.ACCESS để quản lý các thông tin trong CSDL. Trong một file Access, các dữ liệu được chia vào các bảng riêng biệt, gọi là bảng; Việc xem, thêm hay cập nhật dữ liệu thực hiện bằng các cửa sổ nhập liệu, gọi là biểu mẫu - form; Việc tìm kiếm và xử lý các thông tin có chọn lọc thực hiện bằng các truy vấn - query; Việc phân tích và in dữ liệu theo một khuôn mẫu thiết kế trước sử dụng đến các báo biểu - report; để cho phép người dùng xem, sửa, phân tích các thông tin của cơ sở dữ liệu từ Internet hoặc Intranet bằng các trang truy cập dữ liệu, data access pages.

### 5.2.3. Quy trình thiết kế một CSDL

Trước khi bắt tay vào xây dựng các bảng, biểu mẫu, các báo biểu và các thành phần khác của một CSDL, ta cần phải bỏ thời gian thiết kế chi tiết.

#### **Các bước cơ bản để thiết kế một CSDL là:**

1. Xác định mục đích của CSDL: CSDL dùng để lưu trữ thông tin gì, cần rút ra những thông tin gì từ CSDL.
2. Xác định các bảng dữ liệu - bảng - cần có trong CSDL: một Bảng không được chứa dữ liệu trùng lặp, và thông tin cũng không nên để bị trùng lặp giữa các bảng. Mỗi một bảng chỉ nên chứa thông tin liên quan đến một chủ đề.
3. Xác định các trường dữ liệu - field - cần có trong mỗi bảng dữ liệu: Mỗi bảng chứa thông tin về cùng một chủ thể, mỗi field của một bảng lại chứa các thông tin riêng biệt về chủ thể đó. Ta chỉ nên ghi nhận các thông tin cần thiết từ thực tế, không nên có các trường tính toán - calculated field. Lưu trữ thông tin dưới dạng chi tiết nhất có thể (ví dụ: Họ, Tên chứ không nên Họ và Tên).
4. Xác định các trường chứa giá trị duy nhất ở mỗi bản ghi: Để Access kết nối được các thông tin lưu trữ trong các bảng riêng biệt. ví dụ: kết nối các khách hàng với đơn đặt hàng của người đó. Mỗi bảng cần phi có một trường hoặc một nhóm các trường chứa giá trị duy nhất dùng để phân biệt giữa các bản ghi trong một bảng. Một trường như vậy gọi là trường khoá - Primary key.
5. Xác định các mối quan hệ giữa các bảng: Ta chỉ ra cho Access cách để tìm kiếm và đưa các thông tin liên quan lại tập hợp với nhau.
6. Tinh chỉnh thiết kế: Sau khi đã thiết kế, ta cần vận hành thử nghiệm CSDL để đảm bảo các chức năng cần có đều thực hiện suôn sẻ đúng như mong đợi. Nếu phát hiện những thông tin gì không cần thiết thì ta nên loại bỏ để tránh sự công kênh.

7. Nhập dữ liệu và tạo các đối tượng CSDL: Khi đã hài lòng với cấu trúc các bảng biểu và các mối liên kết là lúc phi nhập liệu. Dựa trên dữ liệu đã nhập ta có thể dễ dàng xây dựng các biểu mẫu, báo biểu, truy vấn để phục vụ công tác xử lý đầu vào và đầu ra thông tin.

8. Sử dụng công cụ phân tích của Access: Access cung cấp công cụ Bảng Analyzer Wizard giúp người dùng phân tích thiết kế đã có và chỉnh sửa chúng.

### 5.3. CÂU HỎI VÀ BÀI TẬP CHƯƠNG 5

#### Câu hỏi

1. Tập CSDL của ACCESS có đuôi (phần mở rộng) là:

- dbf
- mdb
- xls

2. Khi bắt đầu vào làm việc với M.Access, nếu muốn thiết lập một CSDL mới. Ta chọn:

- Blank Database
- Open an Existing Database

3. Khi bắt đầu khởi động M.Access, nếu muốn mở 1 tập CSDL cũ ra để xem, sửa.... Ta chọn:

- Blank Wizard
- Open an Existing Database

4. Đối tượng nào là đối tượng nằm trong tập CSDL của Acces

- Table, Query
- Form, Macro
- Report, Module
- Cả 3 nhóm trên

5. Cửa sổ Database bao gồm các nút nào:

- Các nút đối tượng: Bảng, Query, Form, Report, Macro. Moduls...
- Các nút hành động (Open, Design, New)
- Cả hai nhóm nút trên

6. Trong cửa sổ Database nút hành động **Open** dùng để:

- Mở 1 đối tượng
- Đóng 1 đối tượng
- Tạo 1 đối tượng mới

7. Trong cửa sổ Database nút hành động **New** dùng để:

- Mở 1 đối tượng
- Đóng 1 đối tượng
- Tạo 1 đối tượng mới

8. Trong cửa sổ Database nút hành động **Design** dùng để:

- Chỉnh sửa thiết kế 1 đối tượng
- Đóng 1 đối tượng
- Tạo 1 đối tượng mới

9. Khi đang làm việc với M.Access muốn tạo ra một tệp CSDL mới khác ta chọn **File** trên thanh **menu** sau đó chọn:

- New
- Open
- Close

10. Khi đang làm việc với M.Access muốn mở 1 tệp CSDL khác (đã có) ta chọn **File** trên thanh **menu** sau đó chọn:

- New
- Open
- Save as

11. Khi đang làm việc với M.Access muốn đóng tệp CSDL đang được mở ta chọn **File** trên thanh **menu** sau đó chọn:

- Save
- Open
- Close

12. Khi đang làm việc với M.Access muốn thoát ra khỏi M.Access để trở về hệ điều hành Windows ta chọn **File** trên thanh **menu** sau đó chọn:

- Close
- Open
- Exit

13. Một tệp CSDL của M.Access gồm có:

- 1 bảng
- Nhiều bảng
- 2 bảng

14. Một tệp CSDL của M.Access gồm có:

- 1 Query
- 2 Query
- Nhiều Query

15. Trong cửa sổ database muốn tạo bảng mới ta chọn Table (Bảng) sau đó chọn:

- New
- Design
- Open
- Creat

16. Trong cửa sổ database muốn mở 1 bảng ở chế độ nhập liệu ta chọn Table (Bảng) sau đó chọn tên bảng cần mở rồi chọn tiếp:

- New
- Design
- Open
- Use

17. Trong cửa sổ database muốn chỉnh sửa lại cấu trúc của bảng (ví dụ thêm 1 trường mới vào bảng) ta chọn Table (Bảng) sau đó chọn tên bảng cần chỉnh rồi chọn tiếp:

- New

- Design
- Open
- print

18. Khi tạo bảng, muốn nhập tên trường ta chọn cột:

- Field name
- Data type
- Properties
- Edit

19. Khi tạo bảng, muốn chỉ định kiểu dữ liệu cho trường thì trên hàng của trường đó ta chọn cột:

- Field name
- Data type
- Insert

20. Khi tạo khoá chính cho trường thì trước hết ta chọn trường đó, sau đó chọn **insert** trên thanh menu rồi tiếp tục chọn:

- select all
- primary key
- copy
- cut

21. Khi tạo bảng dữ liệu, ta muốn khai báo một trường có **kiểu số** thì trên cột **Data Type** tại dòng tương ứng của trường ta chọn:

- Number
- Text
- Date/time
- Yes/no

22. Khi tạo bảng dữ liệu, ta muốn khai báo một trường có **kiểu đối tượng hình ảnh** thì trên cột **Data Type** tại dòng tương ứng của trường ta chọn:

- OLE Object
- Text

- Hyperlink
- Currency

23. Muốn xây dựng liên kết giữa các bảng có liên quan thì ta phải: thêm các bảng có liên quan vào cửa sổ?

- Relationships window
- Database
- file
- open

24. Trong cửa sổ **Database** muốn tạo 1 query mới ta chọn **Query** sau đó chọn:

- New
- Design
- Open
- file

25. Trong cửa sổ **Database** muốn chỉnh sửa 1 query ta chọn **Query** sau đó chọn tên query cần sửa rồi chọn tiếp:

- save
- New
- Design
- Open

26. Trong cửa sổ **Database** muốn thực hiện 1 query ta chọn **Query** sau đó chọn tên query thực hiện rồi chọn tiếp:

- save as
- New
- Design
- Open

27. Truy vấn nào trong các truy vấn sau là truy vấn xóa:

- Delete query
- Open query

Append query

Make-table query

**28.** Truy vấn nào trong các truy vấn sau là truy vấn bổ sung

Insert query

Update query

Append query

Make-table query

**29.** Truy vấn nào trong các truy vấn sau là truy vấn tạo Bảng:

New query

Bảng query

Append query

Make-table query

**30.** Muốn tạo mới form bằng wizard trước hết trong cửa sổ Database, chọn Forms, nhấn New. Trong hộp thoại New Form chọn:

Form wizard

Design view

Datasheet

Bảng

**31.** Muốn thiết kế form theo nhu cầu trước hết trong cửa sổ Database, chọn Forms, chọn New. Trong hộp thoại New Form sau đó ta tiếp tục chọn:

Form wizard

Design view

Datasheet

Bảng

**32.** Muốn tạo mới một báo biểu trước hết tại cửa sổ Database ta chọn:

Bảng

Form

Report

Module



33. Sau khi đã thực hiện xong câu 32 muốn tạo Report ở chế độ Wizard ta chọn:

- Design View
- Report Wizard
- AutoReport: tabular

### Bài tập thực hành

Xây dựng chương trình quản lý sinh viên

Yêu cầu về giao diện:

Sau khi thực hiện xong bài tập này bạn sẽ có một chương trình quản lý tuy rất nhỏ, gọn nhưng có đầy đủ các chức năng: Nhập, xem dữ liệu, thực hiện các chức năng tìm kiếm, xóa bản ghi, in các báo cáo. Khi chạy chương trình ta sẽ thấy màn hình như sau:



- Các nút lệnh trên bảng chọn có các chức năng như sau:
- Khi nhấn vào nút "Thoát khỏi Access" thì sẽ thoát khỏi Access và trở về Windows
- Khi nhấn vào nút "Về cửa sổ Database" thì thoát khỏi bảng chọn, trở về cửa sổ Database của Access. Từ cửa sổ này ta có thể tạo mới, sửa đổi các đối tượng.
- Khi nhấn vào nút "Nhập danh sách khoa" ta thấy cửa sổ sau xuất hiện:

| MÃ KHOA | TÊN KHOA            |
|---------|---------------------|
| DT      | Điện tu             |
| HH      | Hóa học             |
| QT      | Quản trị kinh doanh |
| TH      | Tin học             |
| VL      | Vật lý              |
|         |                     |

- Trên cửa sổ này ta có thể nhập số liệu cho danh sách khoa: thêm mới, xóa, tìm kiếm...
- Khi nhấn vào nút "Nhập danh sách sinh viên" ta thấy xuất hiện cửa sổ:

|              |            |
|--------------|------------|
| Mã sinh viên | 000        |
| Họ           | Nguyen     |
| Tên          | Tan        |
| Ngày sinh    | 01/01/1982 |
| Dân tộc      | KINH       |
| Địa chỉ      |            |
| Mã khoa      | DT         |
| Điện tu      |            |
| Điểm thi     | 22         |
| Điểm xét     | 22         |

- Khi nhấn vào nút "Danh sách khoa" trong phần "Xem và in báo cáo", ta thấy xuất hiện danh sách khoa trên màn hình và có thể in danh sách này khi nhấn vào biểu tượng máy in.
- Nhấn vào nút "Danh sách sinh viên" trong phần báo cáo ta thấy xuất hiện bảng danh sách sinh viên và có thể in tương tự như trên.

## GỢI Ý TRẢ LỜI CÂU HỎI VÀ BÀI TẬP

### CHƯƠNG 1

Câu 1: Hãy xác định mã nhị phân nào sau đây là kết quả được chuyển đổi từ mã số thập phân  $29.75_{(10)}$  :

c/  $11101.11_{(2)}$

Câu 2: Hãy xác định mã nhị phân nào sau đây là kết quả được chuyển đổi từ mã số hexa-deximal  $3C4_{(16)}$  :

b/  $001111000100_{(2)}$

Câu 3: Hãy xác định kết quả đúng của phép tính cộng 2 số nhị phân  $0101 + 1100$  :

b/ 10001

Câu 4: Hãy xác định kết quả đúng của phép tính nhân 2 số nhị phân  $0110 * 1011$  :

d/ 1000010

Câu 5: Trong phần cứng máy tính, những thiết bị nào sau đây thuộc nhóm thiết bị đầu vào (Input) :

d/ Tất cả đều đúng

Câu 6: Trong phần cứng máy tính, những thiết bị nào sau đây thuộc nhóm thiết bị đầu ra (Output) :

c/ Monitor, Plotter, Printer

Câu 7: Phần mềm AntiVirus được cài đặt trong máy tính thuộc nhóm phần mềm nào sau đây

a/ Phần mềm ứng dụng

Câu 8: Hệ điều hành mạng nào sau đây cho phép thực hiện kiểu mạng ngang hàng :

a/ Windows 2000/NT

Câu 9: Hệ điều hành mạng nào sau đây cho phép thực hiện kiểu mạng Client/Server :

d/ Tất cả đều đúng

Câu 10: Máy tính PC (Personal Computer) mà chúng ta đang sử dụng hiện nay thuộc phân loại máy tính nào sau đây.

a/ Microcomputers

Câu 11: Trong cấu trúc liên kết (Topology) mạng máy tính cục bộ (LAN) có những kiểu cơ bản nào sau đây :

c/ Bus Topology, Star Topology và Ring Topology

Câu 12: Hãy xác định thiết bị liên kết mạng HUB hoạt động ở tầng nào trong mô hình tham chiếu OSI

a/ Tầng vật lý

Câu 13: Hãy xác định thiết bị liên kết mạng Bridge hoạt động ở tầng nào trong mô hình tham chiếu OSI

b/ Tầng liên kết dữ liệu

Câu 14: Hãy xác định thiết bị liên kết mạng Router hoạt động ở tầng nào trong mô hình tham chiếu OSI

c/ Tầng mạng

Câu 15: Giao thức nào sau đây được sử dụng chung trong tất cả các hệ điều hành mạng hiện nay :

b/ TCP/IP

Câu 16: Giao thức nào sau đây được sử dụng dành riêng trong các hệ điều hành mạng của Microsoft :

c/ NetBEUI

Câu 17: Theo giá trị mặc định của IP, địa chỉ cổng 80 được gán sử dụng cho giao thức Internet nào sau đây

a/ HTTP

Câu 18: Theo giá trị mặc định của IP, địa chỉ cổng 23 được gán sử dụng cho giao thức Internet nào sau đây

d/ TELNET

Câu 19: Theo giá trị mặc định của IP, địa chỉ cổng 25 được gán sử dụng cho giao thức Internet nào sau đây

c/ SMTP

Câu 20: Theo giá trị mặc định của IP, địa chỉ cổng 21 được gán sử dụng cho giao thức Internet nào sau đây

b/ FTP

## CHƯƠNG 2

### Câu hỏi

1. Nêu khái niệm về hệ điều hành:

- Xem phần khái niệm hệ điều hành (HĐH)

2. Phân loại hệ điều hành

- Xem phần phân loại HĐH

3. Hệ điều hành nào là HĐH đơn nhiệm?

DOS

4. Hệ điều hành nào là HĐH đa nhiệm?

WINDOWS 2000

5. Trong HĐH MS\_DOS đĩa hệ thống là đĩa chứa ít nhất những tệp nào?

Cả 3 tệp trên

6. Định nghĩa nào là định nghĩa tệp tin:

Tệp tin là nơi lưu trữ thông tin bao gồm chương trình, dữ liệu, văn bản,... Mỗi tệp tin có một tên riêng phân biệt. Tên tệp tin thường có 2 phần: phần tên (name) và phần mở rộng (extension). Phần tên là bắt buộc phải có của một tệp tin, còn phần mở rộng thì có thể có hoặc không.

7. Định nghĩa nào là định nghĩa đường dẫn:

Là lệnh chỉ dẫn lộ trình cho phép ta ( con trỏ) từ thư mục bất kỳ có thể đến trực tiếp thư mục cần truy xuất.

8. Muốn tạo một thư mục bằng lệnh của MSDOS thì ta dùng lệnh nào trong các lệnh sau?

Md

9. Muốn tạo một tệp bằng lệnh của MSDOS thì ta dùng lệnh nào trong các lệnh sau :

Copy con

10. Muốn tạo xóa một tệp bằng lệnh của MSDOS thì ta dùng lệnh nào trong các lệnh sau :

del

11. Muốn khởi tạo đĩa mới bằng lệnh của MSDOS thì ta dùng lệnh nào trong các lệnh sau :

Format

12. Muốn tạo đĩa khởi động tại ổ đĩa A: bằng lệnh của MSDOS thì ta dùng lệnh nào trong các lệnh sau :

Format A:/s

13. Muốn xóa một thư mục bằng lệnh của MSDOS thì ta dùng lệnh nào trong các lệnh sau :

Rd

14. Muốn đổi tên một tệp bằng lệnh của MSDOS thì ta dùng lệnh nào trong các lệnh sau

rename

15. Trong HĐH WINDOWS chương trình ứng dụng Control Panel dùng để:

Cài đặt hay thay đổi cấu hình của hệ thống

16. Windows Explorer là công cụ trợ giúp, giúp ta biết được:

Toàn bộ máy tính có những gì, cho phép xem nội dung của máy tính như một hệ thống thứ bậc hay cấu trúc hình cây. Cho phép xem nội dung mỗi ổ đĩa, mỗi thư mục, bất kỳ một ổ đĩa nào trong mạng

17. Khi muốn di chuyển dữ liệu từ vùng nguồn đến vùng đích, ta phải sử dụng cặp lệnh nào trong các cặp lệnh nào dưới đây?

□ Cut,Paste

18. Khi muốn copy dữ liệu từ vùng nguồn đến vùng đích, ta phải sử dụng lệnh nào trong cặp lệnh nào dưới đây?

□ Copy,Paste

### **Bài tập thực hành**

**Bài 1.** Dùng các lệnh cơ bản của MSDOS để thực hiện như MD,COPY CON, COPY, RENAME,CLS,TYPE,DATE, TIME, DEL,RD,...

Bài 2: Thực hành các yêu cầu trên HĐH WINDOW2000

1,2.

- Để thực hành các thao tác theo yêu cầu của đề, chú ý cách sử dụng chuột, đóng mở cửa sổ màn hình và các đối tượng, các menu

3.

- Chú ý cách tạo biểu tượng đường tắt shortcut để gọi tới 1 tệp bất kỳ (xem phần màn hình Desktop và các biểu tượng)

4.

- Chú ý phần cửa sổ và cách điều khiển cửa sổ như cực đại, phóng to, thu nhỏ... cửa sổ.

5. 6. Xem kỹ lại phần màn hình giao diện, cách sử dụng chuột

7. Xem lại phần khởi động 1 chương trình ứng dụng

8. Xem lại phần Control Panel

9. Đề ý kỹ lại phần điều khiển cửa sổ

Bài 3: Để thực hành các yêu cầu của bài này chúng ta xem xét lại nội dung của phần Window Explorer

1. Xóa thư mục thực hành cùng với các thư mục con. Đóng cửa sổ Windows Explorer.

Bài 4. Để thực hiện bài này hãy xem lại phần Công cụ Paint trong Window

## CHƯƠNG 3

### Phần Word

#### I. Câu hỏi

1. Khi đang soạn thảo trên phần mềm Word, nếu muốn ghi dữ liệu vào 1 tệp thì ta chọn File trên thanh Menu sau đó chọn:

Save

2. Khi đang soạn thảo trên phần mềm Word, nếu muốn mở 1 tệp thì ta chọn File trên thanh menu sau đó chọn:

Open

3. Khi đang soạn thảo trên phần mềm Word, muốn sao chép một khối ta phải chọn khối muốn sao chép sau đó chọn: Edit rồi tiếp tục chọn :

Copy

7. Khi đang soạn thảo trên phần mềm Word, muốn di chuyển một khối ta phải chọn khối muốn sao chép sau đó chọn: Edit rồi tiếp tục chọn :

Cut

8. Sau khi đã thực hiện câu 7 hoặc câu 8 muốn dán văn bản vào vị trí con trỏ đang đứng ta chọn Edit rồi chọn:

Paste

9. Khi đang soạn thảo, muốn định dạng nhanh bằng thanh công cụ ta dùng thanh công cụ:

Formatting

10. Khi muốn định dạng kí tự cho một đoạn văn bản đã được chọn ta chọn Format sau đó chọn :

Font

12. Muốn thay đổi màu nền hoặc tạo khung cho một đoạn văn bản đã được chọn ta chọn Format sau đó chọn:

- Borders and Shading

13. Trước khi in ta phải chọn cỡ giấy định in bằng cách chọn **file**, chọn **Page Setup**, rồi tiếp tục chọn:

- Paper size

- 

14. Trước khi in nếu ta muốn đặt lề cho trang in trước hết ta làm bằng cách chọn **file**, chọn **Page Setup**, rồi tiếp tục chọn:

- Margins

15. Muốn chèn vào công thức toán học thì ta chọn Insert, chọn Object, chọn Creat new rồi tiếp tục chọn:

- Microsoft Equation...

16. Muốn in tất cả các trang trong văn bản trước hết ta làm bằng cách chọn **File**, chọn **Print** trong vùng **Page range** chọn

- All

## II. Bài thực hành

**Bài thực hành 1:** Xem xét kỹ phần Lập bảng biểu. Đặc biệt là cách tạo bảng, sắp xếp dữ liệu trên một bảng, tính toán bằng các hàm đã có sẵn hoặc các công thức toán học trong phần Formula của bảng (trên menu Table chọn Formul)

Bài thực hành 2:

Vào Word, tạo một tệp mới có tên là ThucHanh2.doc và soạn thảo đoạn văn bản bất kỳ sau đó Click chuột trái vào Insert clip Art trên thanh drawing

### Bài thực hành 3

Vào Word, tạo một tệp mới có tên là ThucHanh3.doc và soạn thảo có sử dụng các nút chức năng vẽ của thanh Drawing

### Bài thực hành 4.

Vào Word và mở tệp mới có tên là ThucHanh4.doc. Để gõ đoạn văn bản thì ngoài việc soạn thảo bình thường ta phải biết cách chèn

- Công thức toán học trong Word (chọn Insert, chọn Object, chọn Creat new, Microsoft Equation...)
- Các kí tự đặc biệt ( Insert/symbol)
- Sử dụng các mẫu hình trong Autoshapes như basic shapes
- Chỉ số dưới, chỉ số trên



## Phần Excel

### I. Câu hỏi

18. Khi đang trên bảng tính Excel, nếu muốn ghi dữ liệu vào 1 tệp thì ta chọn File trên thanh Menu sau đó chọn:

- Save

19. Khi đang làm việc trên bảng tính Excel, nếu muốn mở 1 tệp thì ta chọn File trên thanh menu sau đó chọn:

- Open

20. Các kiểu dữ liệu cơ bản trong Excel là: Kiểu số, kiểu chuỗi, kiểu công thức

21. Muốn xóa dữ liệu trong một vùng nào đó thì trước hết ta phải lựa chọn vùng cần xóa, sau đó gõ phím

- Del

22. Muốn sao chép dữ liệu từ vùng nguồn sang một vùng đích thì trước hết ta phải lựa chọn vùng nguồn, sau đó chọn Edit, rồi tiếp tục chọn

- Copy

23. Sau khi đã làm câu 22, muốn dán dữ liệu vào vùng đích, ta phải di chuyển con trỏ vào đến vị trí của ô góc trên bên trái của vùng đích rồi chọn Edit, chọn:

- Paste

24. Muốn chèn dãy số tự động vào một hàng hay một cột thì trước hết gõ giá trị bắt đầu vào ô đầu tiên, chọn vùng cần điền số, thực hiện lệnh Edit chọn, chọn fill, chọn:

- Series

25. Khi muốn điền dữ liệu kiểu công thức hoặc một hàm vào một ô thì ta phải gõ dấu gì trước công thức, hàm:

- =

26. Hàm nào trong các hàm dưới đây là hàm giá trị tuyệt đối của biểu thức số N

- ABS(N)

27. Hàm nào trong các hàm dưới đây là hàm căn bậc hai của số N

- SQRT(N)

28. Hàm nào trong các hàm dưới đây là hàm tính tổng của các giá trị có trong danh sách

- SUM(Danh sách các trị)

29. Hàm nào trong các hàm dưới đây là hàm tìm giá trị nhỏ nhất của các giá trị có trong danh sách

- MIN(Danh sách các trị)

30. Hàm nào trong các hàm dưới đây là hàm tìm giá trị lớn nhất của các giá trị có trong danh sách

- MAX(Danh sách các trị)

31. Hàm nào trong các hàm dưới đây là hàm cho phần dư của phép chia nguyên N cho M

- MOD(N,M)

32. Hàm nào trong các hàm dưới đây là hàm cho giá trị tháng của dữ liệu kiểu ngày

- MONTH (dữ liệu kiểu ngày)

33. Hàm nào trong các hàm dưới đây là hàm cho giá trị năm của dữ liệu kiểu ngày

- YEAR (dữ liệu kiểu ngày)

34. Hàm nào trong các hàm sau đây là hàm cho độ dài của chuỗi TEXT

- LEN (TEXT)

## II. Bài tập thực hành

**Bài thực hành 6 và 7 (Excel):** Để làm 2 bài này cần chú ý:

- Tạo một tệp mới (File\new)
- Cách thực hiện các công thức toán học, các hàm cơ bản (Như hàm Sum,...) , sắp xếp dữ liệu trong Excel
- Tạo các đường kẻ theo vùng ô đã chọn.

....

**Bài thực hành 8:**

### Phần POWERPOINT

#### I. Câu hỏi

35. In ra các slide gồm cả phần dòng ghi chú ra giấy cách nào?

Trong hộp thoại Print, chọn Notes Pages thay vì chọn Handouts.

36. Bằng cách nào để hiển thị đối tượng của slide sao cho sinh động, hấp dẫn người xem?

Xem phần 3.3.3.2.(Làm cho văn bản và đối tượng của slide trở nên sinh động)

## II. Bài thực hành

**Bài thực hành 9 (Power Point):** Xem kỹ lại phần Cách tạo ra các Slide

## CHƯƠNG 4

### Câu hỏi

1. Hãy cho biết giá trị của mỗi biểu thức sau:

- $n+q$  : kết quả là 12
- $n+x$ : kết quả là 11.75
- $n\%p+q$  : kết quả là 4

2. Kết quả đúng của  $z$ : 1.8

3. Kết quả của phép tính:  $23\%3$  là 2

4. Kết quả của đoạn chương trình là:

$n=20$     $p=10$     $t=30$   
 $n=30$     $t=10$

### Bài tập

1. Hiện câu chào

//002CHAO1.cpp Hiện câu chào trên màn hình

```
#include <stdio.h>
void main()
{printf("\nTurbo C xin chào bạn");
}
```

2. Hiện câu chào và chờ nhận phím mới kết thúc

//002CHAO2.CPP Xóa màn hình, hiện câu chào, chờ nhận phím

```
#include <stdio.h>
#include <conio.h>
void main()
{clrscr();
```

```
printf("Turbo C xin chao ban!");
getch();
}
```

3. Nhập 2 số nguyên, tính tổng, hiệu, tích, thương của 2 số nguyên đó  
//003AB1.CPP Nhập 2 số nguyên. Tính tổng, hiệu, tích, thương.

```
#include <stdio.h>
#include <conio.h>
void main()
{clrscr();
int a,b,tong,tich,hieu,thuong;
printf("\nNhập 2 số a,b = ");
scanf("%d%d",&a,&b);
tong=a+b;tich=a*b;hieu=a-b;thuong=a/b;
printf("Tổng của 2 số %d và %d là %d",a,b,tong);
//Hay in ra tích, hiệu và thương
getch();
}
```

4. Nhập 2 số thực, tính tổng, hiệu, tích, thương của 2 số thực đó  
//003AB2.CPP Nhập 2 số thực. Tính tổng, hiệu, tích, thương.

```
#include <stdio.h>
#include <conio.h>
void main()
{clrscr();
float a,b,tong,tich,hieu,thuong;
printf("\nNhập 2 số a,b = ");
scanf("%f%f",&a,&b);
tong=a+b;tich=a*b;hieu=a-b;thuong=a/b;
printf("Tổng của 2 số %4.1f và %4.1f là %4.1f",a,b,tong);
//Hay in ra tích, hiệu và thương
getch();
}
```

5. Nhập 3 số thực, tìm max của chúng

//003ABC1.CPP Nhập 3 số thực. Tìm max của chúng.

```
#include <stdio.h>
#include <conio.h>
void main()
{clrscr();
float a,b,c,max;
printf("\nNhập 3 số a,b,c = ");
scanf("%f%f%f",&a,&b,&c);
max=a;
if(b>max) max=b;
if(c>max) max=c;
printf("Max của 3 số %4.1f, %4.1f và %4.1f là %4.1f",a,b,c,max);
//Hay tìm min
getch();
}
```

6. Liệt kê các số nguyên tố không lớn hơn số n cho trước

//006NT1.CPP Liệt kê các số nguyên tố từ 2 đến n

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
void main()
{clrscr();
int n,i,j, a[1000];
printf("\nNhập n = ");scanf("%d",&n);
for(i=2;i<=n;i++) a[i]=1;//coi nhu i la nguyen to
for(i=2;i<=n;i++)
for(j=2;j<=sqrt(i);j++)
if(i%j==0) a[i]=0;
printf("\n");
for(i=2;i<=n;i++)
if(a[i]) printf("%5d",i);
```

```
gotoxy(2,24);
printf("Xem xong nhan phim bat ky de ket thuc");
getch();
}
```

7. Liệt kê các số nguyên tố từ m đến n

//006NT2.CPP Liet ke cac so nguyen to tu m den n

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
#include <math.h>
```

//Ham nt tra ve 1 neu la nguyen to, 0 neu khong phai nguyen to

```
int nt(int n)
```

```
{int i;
```

```
for(i=2;i<=sqrt(n);i++)
```

```
if(n%i==0) return(0);
```

```
return(1);
```

```
}
```

```
void main()
```

```
{clrscr();
```

```
int m,n,i;
```

```
printf("\nNhap m, n = ");scanf("%d%d",&m,&n);
```

```
while(m>=n)
```

```
{printf("\nNhap m, n (m<n) = ");scanf("%d%d",&m,&n);}
```

```
for(i=m;i<=n;i++)
```

```
if(nt(i)) printf("%5d",i);
```

```
gotoxy(2,24);
```

```
printf("Xem xong nhan phim bat ky de ket thuc");
```

```
getch();
```

```
}
```

8. Tìm ước số chung lớn nhất của 2 số bất kỳ nhập vào từ bàn phím

//01EUCLID.CPP, Tim uoc so chung lon nhat cua hai so

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
void main()
{clrscr();
 long m,n,t,m1,n1;
 printf("\nm, n =");scanf("%ld%ld",&m,&n);
 //Neu m<n thi doi vi tri m va n
 if(m<n) {t=m;m=n;n=t;}
 m1=m;n1=n;
 while(n>0)
 {t=n;
 n=m%0n;
 m=t;
 }
 printf("\nUSCLN cua %ld va %ld la: %ld ",m1,n1,m);
 getch();
 }
```

### 9. Chuyển đổi 1 số nguyên thập phân sang dạng nhị phân

//02NPHAN.CPP Chuyển doi mot so thap phan sang nhi phan

```
#include <stdio.h>
#include <conio.h>
#include <dos.h>
#include <stdlib.h>
#define nmax 100
void main()
{clrscr();
 int a[nmax];long n,i,j,k;
 printf("\nNhap so can doi sang so nhi phan: ");scanf("%ld",&n);
 i=0;
 while(n>0)
```

```
{a[i]=n%2;
n=n/2;
i++;
}
i--;
printf("\n");
for(j=i;j>=0;j--) printf("%d",a[j]);
getch();
}
```

10. Đảo một chuỗi kí tự

//06DCHUOI.CPP, Dao chuoi ky tu

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
#include <stdlib.h>
```

```
#include <string.h>
```

```
#define nmax 40
```

```
void daochuoi(char *a)
```

```
{int n = strlen(a);int i,j;char ch;
```

```
i=0;j=n-1;
```

```
while(i<j)
```

```
{ch=a[i];a[i]=a[j];a[j]=ch;i++;j--;
```

```
}
```

```
};
```

```
//=====
```

```
void main()
```

```
{clrscr();
```

```
char a[20];char *b;
```

```
strcpy(a,"12345678");
```

```
printf("\nChuoi ban dau la: %s",a);
```

```
daochuoi(a);
```

```
printf("\nChuoi dao lai la: %s",a);
```

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Km10 Đường Nguyễn Trãi, Hà Đông-Hà Tây  
Tel: (04) 5541221; Fax: (04) 5540587

Website: <http://www.c-ptit.edu.vn>; E-mail: [dhk@ptit.edu.vn](mailto:dhk@ptit.edu.vn)

CHƯƠNG TRÌNH  
ĐÀO TẠO ĐẠI HỌC TỪ XA



```

 getch();
}
11. Tìm số lớn nhất trong dãy số thực
//09TimMax.CPP Tim Max trong day phan tu
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#define nmax 100
//=====
void nhap(float a[],int &n)
{int i,j,k;
printf("\nNhap co cua mang: ");scanf("%d",&n);
printf("\n");
for(i=0;i<n;i++)
 {printf("Nhap a[%d] = ",i); scanf("%f",&a[i]);
 }
clrscr();
}
//=====
int timmax(float a[],int n) //Tra ve vi tri k ma a[k] la max
{int i,k;float max;
max=a[0];k=0;
for(i=1;i<n;i++)
 if(a[i]>max) {max=a[i];k=i;}
return(k);
}
//=====
void view(float a[],int n)
{int i;
printf("\n");
for(i=0;i<n;i++) printf("%3.1f ",a[i]);

```

```
}
//=====
void main()
{clrscr();
float a[nmax];int n,k;
nhap(a,n);
printf("\nDanh sach da nhap la:");
view(a,n);
k=timmax(a,n);
printf("\nPhan tu cuc dai la a[%d] = %.1f",k,a[k]);
getch();
}
```

12. Tìm xem 1 số thực x có xuất hiện trong dãy số thực hay không

```
//TimX.CPP Tim X trong day phan tu
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#define nmax 100
//=====
void nhap(float a[],int &n)
{int i,j,k;
printf("\nNhap co cua mang: ");scanf("%d",&n);
printf("\n");
for(i=0;i<n;i++)
{printf("Nhap a[%d] = ",i); scanf("%f",&a[i]);
}
clrscr();
}
//=====
int timkiem(float a[],int n,float x) //Tra ve vi tri k ma a[k] la a[k]=x
{int i;
```

```

for(i=0;i<n;i++)
 if(a[i]==x) return(i);
return(-1);
}
//=====
void view(float a[],int n)
{int i;
 printf("\n");
 for(i=0;i<n;i++) printf("%3.1f ",a[i]);
}
//=====
void main()
{clrscr();
 float a[nmax],x;int n,k;
 nhap(a,n);
 printf("\nDanh sach da nhap la:");
 view(a,n);
 printf("\nHay nhap gia tri can tim x = ");scanf("%f",&x);
 k=timkiem(a,n,x);
 if(k>=0)
 printf("\nPhan tu tim thay la la a[%d] = %.1f",k,a[k]);
 else
 printf("\nKhong tim thay phan tu %f trong day tren",x);
 getch();
}

```

13. Tính giá trị của đa thức bậc n theo phương pháp Horner

//13HORNER.CPP Tìm gia tri da thuc bac n tai x.

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
#include <stdlib.h>
```

```
#define nmax 100
```

```
//=====
```

```
void nhap(float a[],int &n)
{int i,j,k;
 printf("\nNhap bac cua da thuc: ");scanf("%d",&n);
 printf("\nNhap cac he so tu bac cao den thap:\n");
 for(i=n;i>=0;i--)
 {printf("Nhap a[%d] = ",i); scanf("%f",&a[i]);
 }
 clrscr();
}
//=====
//Tinh p(x) = a[n]*(x mu n) + ... + a[1]*x +a[0]
float horner(float a[],int n,float x) //Tra ve gia tri da thuc tai x
{int i;float s;
 s=a[n];
 for(i=n-1;i>=0;i--) s=s*x+a[i];
 return(s);
}
//=====
void view(float a[],int n)
{int i;
 printf("\nCac he so da thuc tu mu bac cao den thap:\n");
 for(i=n;i>=0;i--) printf("%3.1f ",a[i]);
}
//=====
void main()
{clrscr();
 float a[nmax],x;int n,k;
 nhap(a,n);
 printf("\nCac he so cua da thuc tu bac cao den thap la:");
 view(a,n);
 printf("\nHay nhap gia tri can tinh gia tri da thuc, x = ");scanf("%f",&x);
```

```
printf("\nGia tri da thuc tai x = %.1f la %.1f",x,horner(a,n,x));
getch();
}
```

14. Loại trừ các dấu cách thừa trong chuỗi kí tự ( chỉ để lại một dấu cách)  
//14CHTEN.CPP, Loại trừ dấu cách thừa trong tên

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <conio.h>
//=====
void chuanhoa(char *s)
{char *s1; int n, i,k,j;
n=strlen(s);
s1=new char[n+1];
i=k=0;
while(s[i]!=' ') i++;//Cắt dấu cách bên trái chuỗi
while(i<n)
{if(s[i]!=' ') s1[k++]=s[i++];
else
{s1[k++]=s[i++];
while(s[i]!=' ') i++;
}
}
k=k-1;
while(s1[k]!=' ') k--;//Cắt dấu cách bên phải chuỗi
s1[k+1]='\0';
strcpy(s,s1);
}
//=====
void main()
{clrscr();
```

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG  
Km10 Đường Nguyễn Trãi, Hà Đông-Hà Tây  
Tel: (04) 5541221; Fax: (04) 5540587  
Website: <http://www.c-phl.edu.vn>; E-mail: [dhk@phl.edu.vn](mailto:dhk@phl.edu.vn)

CHƯƠNG TRÌNH  
ĐÀO TẠO ĐẠI HỌC TỪ XA

```
char *s1=" Nguyen Anh Hung ";
puts(s1);
printf("\n");
chuanhoa(s1);
printf("\n%s",s1);
getch();
}
```

15. Đếm số chữ trong xâu kí tự

//15DEMCHU.CPP, dem chu trong chuoai

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <string.h>
```

```
#include <conio.h>
```

```
//=====
```

```
void chuanhoa(char *s)
```

```
{char *s1; int n, i,k,j;
```

```
n=strlen(s);
```

```
s1=new char[n+1];
```

```
i=k=0;
```

```
while(s[i]!=' ') i++;//Cat dau cach ben trai chuoai
```

```
while(i<n)
```

```
{if(s[i]!=' ') s1[k++]=s[i++];
```

```
else
```

```
{s1[k++]=s[i++];
```

```
while(s[i]!=' ') i++;
```

```
}
```

```
}
```

```
k=k-1;
```

```
while(s1[k]!=' ') k--;//Cat dau cach ben phai chuoai
```

```
s1[k+1]='\0';
```

```
strcpy(s,s1);
```

```

}
//=====
//=====
int demchu(char *s) //Tra ve so chu trong chuoai
{char *s1; int n,i,dem;
 n=strlen(s);
 s1=new char[n+1];
 strcpy(s1,s);
 chuanhoa(s1);
 n=strlen(s1);
 dem=0;
 for(i=0;i<n;i++)
 if(s1[i]==' ') dem++;
 return(dem+1);
}
//=====
void main()
{clrscr();
 char *s1=" Nguyen Anh Hung ";
 puts(s1);
 printf("\nSo tu trong chuoai tren la: %d",demchu(s1));
 getch();
}

```

16. Tính số  $\pi$  thức công thức

$$\pi/4 = 1 - 1/3 + 1/5 - 1/7 + \dots + (-1)^N / (2N+1)$$

với độ chính xác Epsilon thỏa mãn:

$$1/(2N+1) < \text{Epsilon} = 1.0\text{E-}04$$

//16TINHPI.CPP, Tinh so pi

#include <stdio.h>

```
#include <conio.h>
#include <stdlib.h>
#include <math.h>
//=====
double tinhpi()
{double tong,sh,epsi,dau;
int n;
epsi=0.0001;
tong=0;
sh=1;
dau=1;
n=0;
while(fabs(sh)>epsi)
{tong=tong+sh;
n++;
dau=-dau;
sh=dau/(2*n+1);
}
return(tong*4);
}
//=====
void main()
{clrscr();
printf("\n Pi = %f",tinhpi());
getch();
}
```

### 17. Cần lưu ý

- Phần mảng 2 chiều
- Cấu trúc điều khiển
- Cách xây dựng hàm (Hàm con)



## CHƯƠNG 5

### Câu hỏi

1. Tệp CSDL của ACCESS có đuôi (phần mở rộng) là:
  - mdb
2. Khi bắt đầu vào làm việc với M.Access, nếu muốn thiết lập một CSDL mới. Ta chọn:
  - Blank Database
3. Khi bắt đầu khởi động M.Access, nếu muốn mở 1 tệp CSDL cũ ra để xem, sửa.... Ta chọn:
  - Open an Existing Database
4. Đối tượng nào là đối tượng nằm trong tệp CSDL của Access
  - Cả 3 nhóm trên
5. Cửa sổ Database bao gồm các nút nào:
  - Cả hai nhóm nút trên
6. Trong cửa sổ Database nút hành động **Open** dùng để:
  - Mở 1 đối tượng
7. Trong cửa sổ Database nút hành động **New** dùng để:
  - Tạo 1 đối tượng mới
8. Trong cửa sổ Database nút hành động **Design** dùng để:
  - Chỉnh sửa thiết kế 1 đối tượng
9. Khi đang làm việc với M.Access muốn tạo ra một tệp CSDL mới khác ta chọn **File** trên thanh **menu** sau đó chọn:
  - New
10. Khi đang làm việc với M.Access muốn mở 1 tệp CSDL khác(đã có) ta chọn **File** trên thanh **menu** sau đó chọn:
  - Open
11. Khi đang làm việc với M.Access muốn đóng tệp CSDL đang được mở ta chọn **File** trên thanh **menu** sau đó chọn:
  - Close
12. Khi đang làm việc với M.Access muốn thoát ra khỏi M.Access để trở về hệ điều hành Windows ta chọn **File** trên thanh **menu** sau đó chọn:
  - Exit

13. Một tệp CSDL của M.Access gồm có:

- Nhiều bảng

14. Một tệp CSDL của M.Access gồm có:

- Nhiều Query

15. Trong cửa sổ database muốn tạo bảng mới ta chọn Table (Bảng) sau đó chọn:

- New

16. Trong cửa sổ database muốn mở 1 bảng ở chế độ nhập liệu ta chọn Table (Bảng) sau đó chọn tên bảng cần mở rồi chọn tiếp:

- Open

17. Trong cửa sổ database muốn chỉnh sửa lại cấu trúc của bảng (ví dụ thêm 1 trường mới vào bảng) ta chọn Table (Bảng) sau đó chọn tên bảng cần chỉnh rồi chọn tiếp:

- Design

18. Khi tạo bảng, muốn nhập tên trường ta chọn cột:

- Field name

19. Khi tạo bảng, muốn chỉ định kiểu dữ liệu cho trường thì trên hàng của trường đó ta chọn cột:

- Data type

20. Khi tạo khoá chính cho trường thì trước hết ta chọn trường đó, sau đó chọn **insert** trên thanh menu rồi tiếp tục chọn:

- primary key

21. Khi tạo bảng dữ liệu, ta muốn khai báo một trường có **kiểu số** thì trên cột **Data Type** tại dòng tương ứng của trường ta chọn:

- Number

22. Khi tạo bảng dữ liệu, ta muốn khai báo một trường có **kiểu đối tượng hình ảnh** thì trên cột **Data Type** tại dòng tương ứng của trường ta chọn:

- OLE Object

23. Muốn xây dựng liên kết giữa các bảng có liên quan thì ta phải: thêm các bảng có liên quan vào cửa sổ

- Relationships window

24. Trong cửa sổ **Database** muốn tạo 1 query mới ta chọn **Query** sau đó chọn:

- New

25. Trong cửa sổ **Database** muốn chỉnh sửa 1 query ta chọn **Query** sau đó chọn tên query cần sửa rồi chọn tiếp:

- Design

26. Trong cửa sổ **Database** muốn thực hiện 1 query ta chọn **Query** sau đó chọn tên query thực hiện rồi chọn tiếp:

- Open

27. Truy vấn nào trong các truy vấn sau là truy vấn xóa:

- Delete query

28. Truy vấn nào trong các truy vấn sau là truy vấn bổ sung:

- Append query

29. Truy vấn nào trong các truy vấn sau là truy vấn tạo Bảng:

- Make-table query

30. Muốn tạo mới form bằng wizard trước hết trong cửa sổ Database, chọn Forms, nhấn New. Trong hộp thoại New Form chọn:

- Form wizard

31. Muốn thiết kế form theo nhu cầu trước hết trong cửa sổ Database, chọn Forms, chọn New. Trong hộp thoại New Form sau đó ta tiếp tục chọn:

- Design view

32. Muốn tạo mới một báo biểu trước hết tại cửa sổ Database ta chọn:

- Report

33. Sau khi đã thực hiện xong câu 32 muốn tạo Report ở chế độ Wizard ta chọn:

- Report Wizard

### **Bài tập thực hành**

Để thực hiện được bài này ta cần để ý những bước sau:

Tạo bảng: bảng danh sách khoa, bảng danh sách sinh viên

Tạo các query: Thêm mới (dùng append query), Tìm kiếm (dùng Select query có thêm điều kiện cần tìm), Xóa khoa, Xóa sinh viên (dùng Delete query)

Tạo form ở chế độ tự thiết kế (Design View): form “Nhập danh sách khoa” và form “Nhập danh sách sinh viên” theo hình vẽ. Cuối cùng mới tạo form chính là Form “Chương trình quản lý sinh viên”

Tạo Report: tạo Report Danh sách khoa và danh sách sinh viên

Chú ý khi tạo Form “Nhập danh sách khoa” và form “Nhập danh sách sinh viên”, sử dụng các nút trên thanh Toolbox để tạo các điều khiển, các nút đó có thể là: Text Box, Combo Box, Labe, Command Button. Tạo Command Button bằng Control Wizard (dùng Command button tạo ra các điều khiển Thêm mới, Tìm kiếm, Xóa khoa, Xóa sinh viên, các nút này có tác dụng mở query tương ứng đã xây dựng ở bước trước. Còn nút đóng form thì gọi tới Close Form. Chú ý nên dùng).

Đối với Form “Chương trình quản lý sinh viên” chỉ sử dụng các nút trên thanh Toolbox để tạo các điều khiển, các nút đó có thể là: Labe, Command Button. Tạo Command Button bằng Control Wizard (các điều khiển này để mở form và report tương ứng đã xây dựng ở bước trên. Còn điều khiển đóng form thì gọi tới chức năng của Close Form của hành động Action trong Categories, thoát khỏi Access thì gọi tới chức năng Quit Application của hành động Action trong Categories).



## PHỤ LỤC 1: BẢNG MÃ ASCII

BẢNG MÃ ASCII với 128 ký tự đầu tiên

| Hex      | 0        | 1         | 2        | 3       | 4       | 5       | 6        | 7        |
|----------|----------|-----------|----------|---------|---------|---------|----------|----------|
| <b>0</b> | NUL<br>0 | DLE<br>16 | SP<br>32 | 0<br>48 | @<br>64 | P<br>80 | `<br>96  | p<br>112 |
| <b>1</b> | SOH<br>1 | DC1<br>17 | !<br>33  | 1<br>49 | A<br>65 | Q<br>81 | a<br>97  | q<br>113 |
| <b>2</b> | STX<br>2 | DC2<br>18 | “<br>34  | 2<br>50 | B<br>66 | R<br>82 | b<br>98  | r<br>114 |
| <b>3</b> | ♥<br>3   | DC3<br>19 | #<br>35  | 3<br>51 | C<br>67 | S<br>83 | c<br>99  | s<br>115 |
| <b>4</b> | ♦<br>4   | DC4<br>20 | \$<br>36 | 4<br>52 | D<br>68 | T<br>84 | d<br>100 | t<br>116 |
| <b>5</b> | ♣<br>5   | NAK<br>21 | %<br>37  | 5<br>53 | E<br>69 | U<br>85 | e<br>101 | u<br>117 |
| <b>6</b> | ♠<br>6   | SYN<br>22 | &<br>38  | 6<br>54 | F<br>70 | V<br>86 | f<br>102 | v<br>118 |
| <b>7</b> | BEL<br>7 | ETB<br>23 | ‘<br>39  | 7<br>55 | G<br>71 | W<br>87 | g<br>103 | w<br>119 |
| <b>8</b> | BS<br>8  | CAN<br>24 | (<br>40  | 8<br>56 | H<br>72 | X<br>88 | h<br>104 | x<br>120 |
| <b>9</b> | HT<br>9  | EM<br>25  | )<br>41  | 9<br>57 | I<br>73 | Y<br>89 | I<br>105 | y<br>121 |
| <b>A</b> | LF<br>10 | SUB<br>26 | *<br>42  | :<br>58 | J<br>74 | Z<br>90 | j<br>106 | z<br>122 |
| <b>B</b> | VT<br>11 | ESC<br>27 | +<br>43  | ;<br>59 | K<br>75 | [<br>91 | k<br>107 | {<br>123 |

Phụ lục 1: Bảng mã ASCII

|          |          |          |         |         |         |         |          |            |
|----------|----------|----------|---------|---------|---------|---------|----------|------------|
| <b>C</b> | FF<br>12 | FS<br>28 | ,<br>44 | <<br>60 | L<br>76 | \<br>92 | l<br>108 | <br>124    |
| <b>D</b> | CR<br>13 | GS<br>29 | -<br>45 | =<br>61 | M<br>77 | ]<br>93 | m<br>109 | }<br>125   |
| <b>E</b> | SO<br>14 | RS<br>30 | .<br>46 | ><br>62 | N<br>78 | ^<br>94 | n<br>110 | ~<br>126   |
| <b>F</b> | SI<br>15 | US<br>31 | /<br>47 | ?<br>63 | O<br>79 | _<br>95 | o<br>111 | DEL<br>127 |

**BẢNG MÃ ASCII với ký tự số 128 - số 255**

| Hex      | 8        | 9        | A        | B        | C        | D        | E        | F        |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| <b>0</b> | Ç<br>128 | É<br>144 | á<br>160 | ⌘<br>176 | Ł<br>192 | ⌚<br>208 | α<br>224 | ≡<br>240 |
| <b>1</b> | ü<br>129 | æ<br>145 | í<br>161 | ⌘<br>177 | ł<br>193 | ⌛<br>209 | β<br>225 | ±<br>241 |
| <b>2</b> | é<br>130 | Æ<br>146 | ó<br>162 | ⌘<br>178 | Ṭ<br>194 | π<br>210 | Γ<br>226 | ≥<br>242 |
| <b>3</b> | â<br>131 | ô<br>147 | ú<br>163 | <br>179  | ƒ<br>195 | ℓ<br>211 | π<br>227 | ≤<br>243 |
| <b>4</b> | ä<br>132 | ö<br>148 | ñ<br>164 | †<br>180 | —<br>196 | ℓ<br>212 | Σ<br>228 | ∫<br>244 |
| <b>5</b> | à<br>133 | ò<br>149 | Ñ<br>165 | ‡<br>181 | †<br>197 | ƒ<br>213 | σ<br>229 | ∫<br>245 |
| <b>6</b> | å<br>134 | û<br>150 | ª<br>166 | ‡<br>182 | ƒ<br>198 | π<br>214 | μ<br>230 | ÷<br>246 |
| <b>7</b> | ç<br>135 | ù<br>151 | º<br>167 | π<br>183 | ‡<br>199 | ‡<br>215 | τ<br>231 | ≈<br>247 |
| <b>8</b> | ê<br>136 | ÿ<br>152 | ¿<br>168 | ‡<br>184 | ℓ<br>200 | ‡<br>216 | Φ<br>232 | °<br>248 |
| <b>9</b> | ë<br>137 | Ö<br>153 | ƒ<br>169 | ‡<br>185 | ƒ<br>201 | ∟<br>217 | Θ<br>233 | ·<br>249 |

|          |          |          |          |          |          |          |          |          |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| <b>A</b> | è<br>138 | Û<br>154 | ¬<br>170 | <br>186  | ≡<br>202 | Γ<br>218 | Ω<br>234 | •<br>250 |
| <b>B</b> | ï<br>139 | ç<br>155 | ½<br>171 | ¶<br>187 | ⌘<br>203 | ■<br>219 | δ<br>235 | √<br>251 |
| <b>C</b> | î<br>140 | £<br>156 | ¼<br>172 | ¶<br>188 | ⌘<br>204 | ■<br>220 | ∞<br>236 | n<br>252 |
| <b>D</b> | ì<br>141 | ¥<br>157 | ı<br>173 | ¶<br>189 | =<br>205 | █<br>221 | φ<br>237 | ²<br>253 |
| <b>E</b> | Ä<br>142 | ₹<br>158 | «<br>174 | ¶<br>190 | ⌘<br>206 | █<br>222 | ε<br>238 | ■<br>254 |
| <b>F</b> | Å<br>143 | f<br>159 | »<br>175 | ¶<br>191 | ≡<br>207 | █<br>223 | ∩<br>239 | 255      |

## PHỤ LỤC 2: CÁC LỖI THƯỜNG GẶP TRONG C

### 1. SOẠN THẢO VÀ CHẠY CHƯƠNG TRÌNH

#### a. Soạn thảo chương trình

Mỗi câu lệnh của C có thể viết trên một dòng hay nhiều dòng nhưng phải kết thúc bằng dấu ; Tuy nhiên khi nhập một chuỗi ký tự mà muốn chuyển sang dòng khác ta phải thêm dấu \ trước khi xuống dòng.

#### b. Dịch và chạy chương trình

Nếu chương trình chưa viết xong ta có thể nhấn F9 để dịch và sửa lỗi. Khi nhấn F9 thì đầu tiên chương trình được dịch sang tệp có đuôi là \*.obj, sau đó liên kết các tệp và dịch sang tệp có đuôi \*.exe có thể chạy được trong môi trường DOS. Khi chương trình đã tương đối hoàn chỉnh thì ta có thể nhấn Ctrl+F9 để dịch và chạy chương trình.

### 2. CÁC LỖI THƯỜNG GẶP

**Khi dịch chương trình có thể xuất hiện 3 loại lỗi sau đây:**

#### **Lỗi được thông báo bởi từ khóa error (lỗi cú pháp):**

Lỗi này thường xảy ra do khi ta soạn thảo chương trình không tuân theo đúng những quy tắc của C, thí dụ int thì ta gõ thành Int; hay ta gõ thiếu ngoặc đn, ngoặc kép chẳng hạn...

Sau đây là một số thông báo lỗi thường gặp loại này:

Unknown preprocessor directive

Chỉ thị tiền xử lý không đúng. Trong trường hợp này bạn phải xem lại các lệnh #include xem bạn có viết sai không.

#### **Declaration terminated incorrectly**

Khai báo kết thúc không đúng. Ví dụ bạn đánh dấu; sau hàm main như sau:

```
void main();
```

Nên lưu ý là sau tên hàm không được đánh dấu ; như trên đây. Sau tên hàm phải là dấu { và kết thúc hàm là dấu }.



Unexpected }

Thừa dấu }. Mỗi lần đánh dấu { thì bạn nên đánh dấu } rồi sau đó gõ các lệnh vào đoạn giữa, như vậy tránh được tình trạng thừa thiếu dấu { hoặc }.

Compound statement missing }

Thiếu dấu }.

Declaration syntax error

Khai báo sai. Ví dụ bạn viết

```
int a,b
printf("Chao");
```

thì máy báo lỗi ở dòng thứ 2. Sở dĩ như vậy là vì khi đọc qua dòng thứ nhất không có dấu; máy cho rằng lệnh chưa kết thúc và còn chuyển tiếp sang dòng thứ 2. Tuy nhiên sang dòng thứ 2 thì máy lại thấy lệnh không phù hợp nên báo lỗi ở dòng này. Cách viết trên đây tương đương với cách viết:

```
int a,b; printf("Chao");
```

Và máy thấy rằng đây là một lệnh không đúng. Còn nếu ta sử dụng các lệnh trên là

```
int a,b;printf("Chao");
```

thì máy không còn báo lỗi nữa vì nó chuyển xuống dòng thứ 2 gặp dấu; và biết là lệnh `int a,b;` được khai báo đúng.

Undefined symbol

Bạn đã sử dụng một biến nào đó mà chưa khai báo. Ví dụ bạn chưa khai báo biến `n` nhưng lại sử dụng trong lệnh:

```
printf("%d",n);
```

chẳng hạn thì máy báo là Undefined symbol 'n'

Function ... should have a prototype

Ví dụ trong lệnh trên bạn viết sai là

```
prinlf("%d",n);
```

thì máy báo là Function 'prinlf' should have a prototype. Nghĩa của câu này là: hàm `prinlf` cần phải có nguyên mẫu.

**Lỗi được thông báo bởi từ khóa Warning (lỗi cảnh báo):**

Lỗi này thường xảy ra do khi ta khai báo biến nhưng không sử dụng tới.

Ví dụ

**... is assigned a value that is never used**

Khai báo và đã gán giá trị cho biến nhưng không sử dụng.

Ví dụ bạn viết các lệnh

```
int n; n=10;
```

nhưng trong các phần tiếp theo không sử dụng n (để hiện ra màn hình chẳng hạn, hay dùng để tính giá trị của biến khác...) thì máy báo là

'n' is assigned a value that is never used

tuy nhiên đây chỉ là thông báo (warning). Khi bạn nhấn F9 để dịch chương trình thì máy vẫn báo là success

Hai loại lỗi trên đây được thông báo ngay khi dịch chương trình thành file \*.obj

**Loại lỗi thứ 3 có thể xảy ra trong quá trình liên kết:**

Lỗi này thường xảy ra, thí dụ khi có lời gọi hàm nhưng hàm chỉ mới có nguyên mẫu mà chưa có khai báo chi tiết.

## TÀI LIỆU THAM KHẢO

1. Nguyễn Tiến Huy, *Giáo trình Tin học căn bản*, TT Tin học Đại học Tổng hợp TP. Hồ Chí Minh, NXB Đồng Nai, 1995.
2. Nguyễn Xuân Quốc Hương, *Tin học A & B*, NXB Trẻ, TP. Hồ Chí Minh, 1996.
3. Tô Văn Nam, *Giáo trình nhập môn tin học*, NXB. Giáo dục, 2004.
4. Đinh Vũ Nhân, *Tin học căn bản*, NXB TP. Hồ Chí Minh, 1995.
5. Quách Tuấn Ngọc, *Giáo trình Tin học căn bản*, NXB Giáo dục, Hà Nội, 1995.
6. Dương Trần Đức - Chu Quang Ngọc, *Mạng cục bộ*, Tài liệu dùng cho các khóa học bồi dưỡng, Trung tâm đào tạo BCVT 1, 2001
7. Bùi Thế Tâm, *Tin học văn phòng*, NXB. Giao Thông Vận Tải, 2003.
8. Phạm Văn Át, *Kỹ thuật lập trình C*, Nhà xuất bản KHKT, 1995.
9. Quách Tuấn Ngọc, *Ngôn ngữ lập trình C*, NXB Thống kê, 2003.
10. Đỗ Xuân Lôi, *Cấu trúc dữ liệu và giải thuật*, NXB KHKT, 1994.
11. Nguyễn Duy Phương, *Kỹ thuật lập trình*, Giáo trình giảng dạy tại Học viện CN-BCVT
12. Brian Kerninghan, Denis Ritchie, *C Language. Norm ANSI*. Prentice Hall, 1988.
13. Bryon Gottfried, *Programming With C*. McGraw Hill, 1996.
14. Carl Townsend, *Understanding C. SAMS*, 1989.
15. Paul Davies, *The Inspensable Guide to C*. Addison Wisley, 1996.
16. Nikolus L.R. Wirth, *Program = Data Structure + Algorithms*. Prentice Hall, 1992.
17. Phạm Văn Át, *Hướng dẫn sử dụng Microsoft Access-* Nhà xuất bản khoa học và kỹ thuật, 1997.
18. Nguyễn Thiện Tâm, *Giáo trình Microsoft Access 2000* - Nhà xuất bản Đại Học Quốc Gia TP.HCM, 2003.
19. Phạm Thế Quế, *Giáo trình Cơ sở dữ liệu-* Nhà xuất bản Bru Điện, 2004.
20. Microsoft Access 2000: Buiding Application with Form and Report

---

## MỤC LỤC

|                                                         |    |
|---------------------------------------------------------|----|
| <b>Giới thiệu môn học</b> .....                         | 3  |
| 1. Giới thiệu chung .....                               | 3  |
| 2. Mục đích .....                                       | 4  |
| 3. Phương pháp nghiên cứu môn học .....                 | 4  |
| <b>Chương 1: Các khái niệm cơ bản</b> .....             | 7  |
| 1. Giới thiệu .....                                     | 7  |
| 2. Tóm tắt chương 1 .....                               | 7  |
| 3. Câu hỏi và bài tập .....                             | 8  |
| <b>Chương 2: Hệ điều hành</b> .....                     | 13 |
| 1. Giới thiệu .....                                     | 13 |
| 2. Tóm tắt chương 2 .....                               | 13 |
| 3. Câu hỏi và bài tập .....                             | 16 |
| <b>Chương 3: Các phần mềm ứng dụng thông dụng</b> ..... | 23 |
| 1. Giới thiệu .....                                     | 23 |
| 2. Tóm tắt chương 3 .....                               | 23 |
| 3. Câu hỏi và bài tập .....                             | 26 |
| <b>Chương 4: Ngôn ngữ lập trình C</b> .....             | 37 |
| 1. Giới thiệu .....                                     | 37 |
| 2. Tóm tắt chương 4 .....                               | 37 |
| 3. Câu hỏi và bài tập .....                             | 41 |
| <b>Chương 5: Hệ quản trị cơ sở dữ liệu ACCESS</b> ..... | 43 |
| 1. Giới thiệu .....                                     | 43 |
| 2. Tóm tắt chương 5 .....                               | 43 |
| 3. Câu hỏi và bài tập .....                             | 45 |
| <b>Gợi ý trả lời câu hỏi và bài tập</b> .....           | 57 |
| Chương 1 .....                                          | 57 |
| Chương 2 .....                                          | 59 |
| Chương 3 .....                                          | 61 |
| Chương 4 .....                                          | 65 |
| Chương 5 .....                                          | 79 |
| <b>Phụ lục 1: Bảng mã ASCII</b> .....                   | 83 |
| <b>Phụ lục 2: Các lỗi thường gặp trong C</b> .....      | 86 |

## BÀI 1: NHỮNG KHÁI NIỆM CƠ BẢN

### I. Khái niệm chung

#### 1.1. Khái niệm về tin học

Tin học là ngành khoa học công nghệ nghiên cứu các phương pháp, các quá trình xử lý thông tin một cách tự động dựa trên các phương tiện kỹ thuật mà chủ yếu là máy tính điện tử.

#### 1.2. Các lĩnh vực của tin học

• **Phần cứng:** Gồm những đối tượng vật lý hữu hình như vi mạch, bản mạch in, dây cáp nối mạch điện, bộ nhớ, màn hình, máy in, thiết bị đầu cuối, nguồn nuôi,... Phần cứng thực hiện các chức năng xử lý thông tin cơ bản ở mức thấp nhất tức là các tín hiệu nhị phân {0,1}

• **Phần mềm:** Là các chương trình (program) điều khiển các hoạt động phần cứng của máy vi tính và chỉ đạo việc xử lý dữ liệu. Phần mềm của máy tính được chia làm hai loại: Phần mềm hệ thống (System software) và phần mềm ứng dụng (Applications software). Phần mềm hệ thống khi được đưa vào bộ nhớ chính, nó chỉ đạo máy tính thực hiện các công việc. Phần mềm ứng dụng là các chương trình được thiết kế để giải quyết một bài toán hay một vấn đề cụ thể để đáp ứng một nhu cầu riêng trong một số lĩnh vực.

Máy tính cá nhân PC (Personal Computer). Theo đúng tên gọi của nó là máy tính có thể sử dụng bởi riêng một người.

#### 1.3. Đơn vị lưu trữ thông tin:

Đơn vị bé nhất dùng để lưu trữ thông tin là bit. Lượng thông tin chứa trong 1 bit là vừa đủ để nhận biết một trong 2 trạng thái có xác suất xuất hiện như nhau. Trong máy vi tính tùy theo từng phần mềm, từng ngôn ngữ mà các số khi đưa vào máy tính có thể là các hệ cơ số khác nhau, tuy nhiên mọi cơ số khác nhau đều được chuyển thành hệ cơ số 2 (hệ nhị phân). Tại mỗi thời điểm trong 1 bit chỉ lưu trữ được hoặc là chữ số 0 hoặc là chữ số 1. Từ bit là từ viết tắt của Binary Digit (Chữ số nhị phân).

Trong tin học ta thường dùng một số đơn vị bội của bit sau đây:

| tên gọi  | Viết tắt | Giá trị                 |
|----------|----------|-------------------------|
| Byte     | B        | 8 bit                   |
| Kilobyte | KB       | 1024 bytes = $2^{10}$ B |
| Megabyte | MB       | 1024KB = $2^{10}$ KB    |
| Gigabyte | GB       | 1024MB = $2^{10}$ MB    |

### II. Hệ Điều Hành MS-DOS

#### 2.1. Hệ điều hành là gì?

Hệ điều hành là phần mềm quan trọng nhất cho máy tính điện tử có nhiệm vụ điều khiển mọi hoạt động cơ sở của máy tính, giúp phần mềm của người sử dụng có thể chạy được trên máy tính. Các hệ điều hành thông dụng: MS-DOS, Windows, Unix, Linux, OS/2, Macintosh ...

#### 2.2. Hệ điều hành MS - DOS

MS-DOS (Microsoft Disk Operating System) là hệ điều hành(HĐH) của tập đoàn khổng lồ Microsoft. Phiên bản đầu tiên của MS-DOS được viết năm 1981.

MS-DOS là HĐH đơn nhiệm (tại một thời điểm chỉ chạy được một trình ứng dụng).

MS-DOS giao diện với người sử dụng thông qua dòng lệnh.

### 2.3. Khởi động hệ thống:

Để khởi động hệ thống, Chúng ta phải có một đĩa mềm gọi là đĩa hệ thống hoặc đĩa cứng được cài đặt ổ đĩa C là đĩa hệ thống. Đĩa hệ thống chứa các chương trình hạt nhân của hệ điều hành DOS. Ít nhất trên đĩa phải có các tập tin IO.SYS, MSDOS.SYS và COMMAND.COM. Chúng ta có thể khởi động MS-DOS bằng các cách sau:

● **TH1:** Khởi động từ ổ đĩa cứng ta chỉ việc bật công tắc điện của máy tính (Power).

● **TH2:** Khởi động từ ổ đĩa mềm: đặt đĩa khởi động vào giá đỡ của ổ đĩa mềm và bật công tắc điện.

● **TH3:** Khởi động từ HĐH Windows 98: Start/ Run/ Command/OK

● **TH4:** Khởi động từ HĐH Windows 2000/ XP: Start/ Run/ CMD/ OK

➤ Khởi động lại hệ thống:

Ta chọn 1 trong các cách sau:

- Ấn nút Reset trên khối hệ thống ( khởi động nóng).

- Dùng tổ hợp phím CTRL + ALT + DEL (khởi động nóng).

- Khi 2 cách này không có tác dụng, chúng ta phải tắt công tắc khối hệ thống và chờ khoảng 1 phút rồi khởi động lại ( khởi động nguội)

### 2.4. Tập tin (File):

➤ Tập tin (hay còn gọi là Tệp) là hình thức, đơn vị lưu trữ thông tin trên đĩa của Hệ điều hành.

➤ Tệp gồm có tên tệp và phần mở rộng (Phần mở rộng dùng để nhận biết tệp đó do chương trình nào tạo ra nó).

➤ Tên Tệp tin được viết không quá **8 ký tự** và không có dấu cách, + , - , \* , / . Phần mở rộng không quá **3 ký tự** và không có dấu cách. Giữa tên và phần mở rộng cách nhau bởi dấu chấm (.).

Tập tin có thể là nội dung một bức thư, công văn, văn bản, hợp đồng hay một tập hợp chương trình.

➤ Ví dụ:

COMMAND.COM Phần tên tệp là COMMAND còn phần mở rộng là COM

MSDOS.SYS Phần tên tệp là MSDOS còn phần mở rộng là SYS

BAICA.MN Phần tên tệp là BAICA còn phần mở rộng là MN

THO.TXT. Phần tên tệp là THO còn phần mở rộng là TXT

Người ta thường dùng đuôi để biểu thị các kiểu tập tin. Chẳng hạn tệp văn bản thường có đuôi DOC, TXT, VNS, ...

Tệp lệnh thường có đuôi COM, EXE

Tệp dữ liệu thường có đuôi DBF, ...  
Tệp chương trình thường có đuôi PRG, ...  
Tệp hình ảnh thường có đuôi JPG, BMP...

## 2.5. Thư mục và cây thư mục

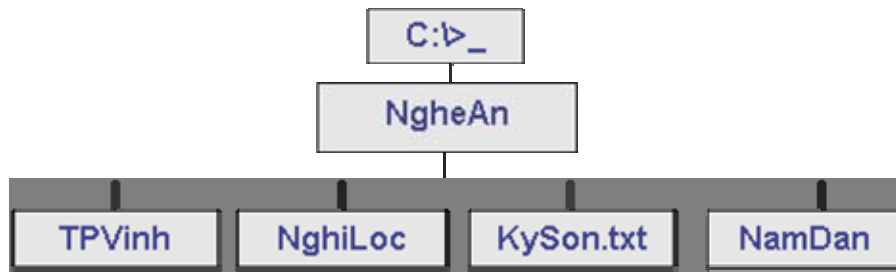
Để có thể tổ chức quản lý tốt tập tin trên đĩa người ta lưu các tập tin thành từng nhóm và lưu trong từng chỗ riêng gọi là thư mục.

Mỗi thư mục được đặc trưng bởi 1 tên cụ thể, quy tắc đặt tên thư mục giống như tên tệp. Các thư mục có thể đặt lồng trong nhau và tạo thành một cây thư mục.

Trong thư mục có thể tạo ra các thư mục con và cứ tiếp tục nhau do đó dẫn đến sự hình thành *một cây thư mục* trên đĩa. Như vậy các thư mục bạn tạo ra có thể là thư mục cấp 1 hay thư mục 2 ...

✦ **Thư mục gốc** là thư mục do định dạng đĩa tạo ra và chúng ta không thể xóa được. Mỗi đĩa chỉ có một thư mục gốc, từ đây người sử dụng tạo ra các thư mục con. Ký hiệu thư mục gốc là dấu (\).

➤ Ví dụ : Cây thư mục



Trong đó C:\>\_ là thư mục gốc. Sau đó là đến các thư mục con các cấp và các tệp.

✦ **Thư mục hiện hành** là thư mục đang được mở, và con trỏ đang nhập nháy chờ lệnh.

Khi thực thi, DOS sẽ tìm kiếm và thi hành ở thư mục hiện hành trước, sau đó mới tìm các thư mục và ổ đĩa được chỉ ra.

✦ **Đường dẫn.**

Khi cần tác động đến một thư mục hoặc tập tin ta phải chỉ ra vị trí của thư mục hay tập tin đó ở trên đĩa hay là phải chỉ ra đường dẫn, tên đường dẫn của thư mục hoặc tập tin tác động tới.

➤ Ví dụ:

Muốn truy xuất tới tệp dữ liệu kyson.txt ở cây thư mục trên ta phải tiến hành chỉ ra đường dẫn như sau;

C:\>Nghean\kyson.txt

✦ Ký tự đại diện \* và ?

**Ký tự đại diện \***: Nó có thể đứng trong phần tên chính hay phần tên mở rộng của tập tin, nó đứng ở vị trí nào sẽ đại diện cho ký tự đó hoặc nhóm ký tự từ vị trí đó đến ký tự sau nó.

➤ Ví dụ:

C:\>\_NGHEAN\A\*B.DOC : Nghĩa là chỉ ra tất cả các tệp có trong thư mục NGHEAN có

phần mở rộng là DOC mà có tên bắt đầu bằng kí tự A và kết thúc bằng kí tự B

C:\>\_NGHEAN\\*. \* : Nghĩa là chỉ ra tất cả các tệp có trong thư mục NGHEAN

C:\>\_NGHEAN\\*.TXT : Nghĩa là chỉ ra tất cả các tệp có trong thư mục NGHEAN mà có phần mở rộng là TXT

**Ký tự đại diện ?**: Nó có thể đứng trong phần tên chính hay phần tên mở rộng của tập tin, nó đứng ở vị trí nào sẽ đại diện cho 1 ký tự tại vị trí đó.

➤ Ví dụ:

C:\>\_?.TXT : Nghĩa là muốn chỉ ra Các tệp có trong ổ đĩa C mà có phần tên chỉ là một kí tự bất kỳ và có phần mở rộng là TXT

## 2.6. Ổ đĩa

Bao gồm :

- ✦ *Ổ đĩa mềm* - gọi là ổ đĩa A: Đĩa mềm có dung lượng 1,44 MB.
- ✦ *Ổ đĩa cứng* - Thường là ổ C,D,E...: và nó nằm ở trong thùng máy, thường có dung lượng lớn gấp nhiều lần so với đĩa mềm.
- ✦ *Ổ đĩa CD* - Là dùng để đọc các đĩa quang. Đĩa quang thường có dung lượng vài trăm MB.

## 2.7. Một số lệnh nội trú và lệnh ngoại trú

### ●1. Lệnh nội trú:

Lệnh nội trú là loại lệnh lưu thường trực trong bộ nhớ trong của máy tính. Nó được nạp vào khi nạp hệ điều hành.

Chúng ta thường gặp một số lệnh nội trú sau:

- ✦ *Lệnh xem danh sách thư mục và tập tin* : DIR
- ✦ *Lệnh tạo lập thư mục*: MD
- ✦ *Lệnh huỷ bỏ thư mục rỗng*: RD
- ✦ *Lệnh chuyển đổi thư mục*: CD
- ✦ *Lệnh thay đổi dấu nhắc lệnh* : PROMPT
- ✦ *Lệnh tạo lập tệp tin*: COPY CON
- ✦ *Lệnh đổi tên tập tin*: REN
- ✦ *Lệnh sao chép tập tin*: COPY
- ✦ *Lệnh hiển thị nội dung tệp tin*: TYPE
- ✦ *Lệnh xoá tập tin*: DEL
- ✦ *Lệnh xoá màn hình*: CLS
- ✦ *Lệnh sửa đổi giờ của hệ thống*: TIME
- ✦ *Lệnh sửa đổi ngày của hệ thống* : DATE
- ✦ *Lệnh hỏi nhãn đĩa*: VOL
- ✦ *Lệnh xem phiên bản của DOS*: VER

### ●2. Lệnh ngoại trú:

Lệnh ngoại trú là lệnh nằm trong bộ nhớ ngoài. Muốn thực hiện các lệnh ngoại trú thì buộc trên đĩa phải có các tệp này. Nếu không có thì phải COPY vào để thực hiện.

Chúng ta thường gặp một số lệnh ngoại trú sau:

- ✦ *Lệnh đặt nhãn đĩa*: LABEL
- ✦ *Lệnh hiển thị cây thư mục*: TREE
- ✦ *Lệnh tạo khuôn cho đĩa (định dạng đĩa)*: FORMAT
- ✦ *Lệnh kiểm tra đĩa*: CHKDSK
- ✦ *Lệnh gán thuộc tính* :ATTRIB
- ✦ *Lệnh in*: PRINT
- ✦ *Lệnh khôi phục tệp đã bị xoá*: UNDELETE



## BÀI 2: CÁC LỆNH HỆ ĐIỀU HÀNH CỦA MS- DOS


### I. Các lệnh nội trú

Lệnh nội trú là những lệnh thi hành những chức năng của HĐH, thường xuyên được sử dụng, được lưu trữ vào bộ nhớ của máy tính từ khi khởi động và thường trú ở trong đó cho tới khi tắt máy.

Cách viết chung:

□ □

#### 1.1. Một số lệnh về hệ thống

 Lệnh xem và sửa ngày: **DATE**

Current Date is Sat 02-04-2000


Enter new Date (mm-dd-yy)

Lúc này có hai tùy chọn

Nếu không thay đổi ngày giờ gõ Enter

Nếu sửa ngày hiện hành ở dòng một thì sửa theo khuôn mẫu (tháng -ngày-năm).

Bạn hãy thay đổi ngày lại cho máy tính ví dụ 31/07/2004.

 Lệnh xem và sửa giờ: **TIME**

Current time is 4:32:35.23a


Enter new time:

Lúc này có hai lựa chọn:

-Nếu không sửa giờ hiện hành của dòng một thì gõ Enter

- Nếu sửa giờ hiện hành thì sửa theo khuôn mẫu (giờ: phút:giây.% giây)

Bạn hãy thay đổi giờ lại cho máy tính thành 05 giờ 05 phút.

 Lệnh thay đổi dấu nhắc lệnh của DOS: **PROMPT**

Lệnh thay đổi dấu đợi lệnh để có thể hiện thị một số thông tin hiện hành theo ý riêng của Người sử dụng.

Prompt [Chuỗi ký tự]

\$P: Thư mục hiện hành \$D: Ngày hiện hành

\$G: Dấu > \$T: Giờ hiện hành


\$: Xuống dòng

 Ví dụ: C>PROMPT \$T \$P\$G ↵

 Lệnh xem phiên bản DOS: **VER**

VER ↵

Bạn muốn xem hiện tại mình đang giao tiếp với HĐH MS-DOS là phiên bản nào.

 Ví dụ: C:\VER ↵

Windows 98 [Version 4.10.2222]

 Lệnh xoá màn hình: **CLS**

CLS

Lệnh xoá toàn bộ dữ liệu trên màn hình đưa con trỏ về góc trên cùng bên trái màn hình.

✦. Chuyển đổi ổ đĩa

Gõ tên ổ đĩa và dấu hai chấm, sau đó nhấn **ENTER**.

*Ví dụ:* A:↵ C:↵

## 1.2. Các lệnh về thư mục

✦. Lệnh xem nội dung thư mục.

DIR [drive:] [Path] [Tên thư mục] [/A][/S][/P][/W]

Trong đó: /P : để xem từng trang  
/W: Trình bày theo hàng ngang  
/A : xem các tập tin có thuộc tính ẩn và hệ thống  
/S: Xem cả thư mục con

➤ *Ví dụ:*

```
DIR C:\WINDOWS /P /W↵
```

Lệnh trên sẽ hiển thị các tệp, thư mục thuộc thư mục WINDOWS nằm trong thư mục gốc ổ đĩa C thành 5 hàng và dừng lại sau khi hiển thị hết một trang màn hình. Muốn xem tiếp chỉ việc nhấn một phím bất kỳ.

✦. Lệnh chuyển về thư mục gốc và vào một thư mục nào đó.

+ Chuyển từ thư mục hiện thời về thư mục gốc của ổ đĩa hiện hành.

```
CD\↵
```

+ Lệnh chuyển về cấp thư mục cao hơn một bậc.

```
CD..↵
```

+ Chuyển vào một thư mục

Lệnh này thay đổi thư mục hiện hành cần làm việc của đĩa nào đó

```
CD [drive:]\[path] (tên thư mục cần vào)
```

➤ *Ví dụ:*

- Từ thư mục C:\ chuyển vào thư mục DAIHOC (Thư mục DAIHOC nằm trên ổ đĩa C)

```
C:\CD DAIHOC ↵
```

```
C:\DAIHOC>_
```

- Từ thư mục DAIHOC, chuyển sang thư mục BAITAP( Thư mục BAITAP nằm trên ổ đĩa C)

```
C:\DAIHOC>CD BAITAP ↵
```

```
C:\DAIHOC\BAITAP>_
```

✦. Lệnh xem ý nghĩa của câu lệnh

```
C:> [Tên lệnh] /? ↵
```

*Ví dụ:* Xem ý nghĩa của câu lệnh CD

```
C:\CD /? ↵
```

✦. Lệnh tạo thư mục con(MD):

```
MD [drive:]\[path] ↵
```

[drive:]\[path] : Chỉ ra đường dẫn đến nơi cần tạo thư mục.

➤ *Ví dụ:*

```
C:\MD HOC ↵ Tạo Thư mục HOC Trên thư mục gốc của Ổ đĩa C
```

C:\MD HOC\HOCDOS ↵ Tạo thư mục HOCDOS là thư mục con của thư mục HOC  
C:\MD A:\DAIHOC ↵ Tạo thư mục DAIHOC trên ổ đĩa A

✦.Lệnh xoá thư mục con(RD)

Lệnh huỷ bỏ (xoá) thư mục:

RD [drive:]\[path]

Chú ý: thư mục cần xoá không phi là thư mục hiện hành và phi là thư mục rỗng (Empty Directory) ( tức là không có một tệp hay một thư mục nào nằm trong nó).

➤ Ví dụ: C:\RD DAIHOC ↵ Xoá thư mục DAIHOC( Là thư mục rỗng) trên ổ đĩa C

### 1.3. Các lệnh làm việc với tập tin

✦. Lệnh sao chép tập tin(COPY):

Lệnh này sao chép một hay một nhóm tệp từ thư mục này sang thư mục khác.

Copy [drive1:]\[path1]\[Filename 1] [drive2:]\[path2]\[Filename 2]

Copy [ổ đĩa]\[đường dẫn]\[tên tệp nguồn] ổ đĩa \[đường dẫn đích]

➤ Ví dụ1: Chép tệp BAOCOA.VNS từ thư mục A:\BKED vào thư mục gốc của ổ đĩa C:  
và lấy tên là BAOCOA.VNS.

C:\>COPY A:\BKED\BAOCOA.VNS ↵

➤ Ví dụ2: Sao chép tệp TUHOC.TXT từ thư mục gốc ổ đĩa C thành tệp HOCTHUOC.TXT ở thư mục HOC nằm trong thư mục gốc ổ đĩa A.

C:\>COPY C:\TUHOC.TXT A:\HOC\HOCTHUOC.TXT ↵

✦. Lệnh cộng tệp:

Cộng nội dung file:

Copy [ổ đĩa]\[đường dẫn]\[tên tệp 1]+[ổ đĩa]\[đường dẫn] [tên tệp 2]+ ... [ổ đĩa]\[đường dẫn]\[tên tệp mới]

Trong cú pháp trên lệnh sẽ thực hiện như sau:

Lần lượt cộng nội dung của các tệp: Tên tệp 1, Tên tệp 2, ... thành một tệp duy nhất có tên là Tên tệp mới.

Chú ý: Trước tên tệp mới không có dấu (+), còn trước tên tệp muốn cộng phi có dấu cộng.

➤ Ví dụ: C:\>COPY CD1.TXT+CD2.TXT+CD3.TXT C:\MYDOCU~1\CD.TXT

✦. Lệnh tạo tệp tin(COPY CON):

Tạo ra file để lưu trữ nội dung của một vấn đề nào đó.

C:\>COPY CON [drive:]\[path]\[File name] ↵

.... Nhập nội dung của tệp

F6 ↵

1 file(s) is copied

C:\

(Nếu như tệp được tạo thì sau khi nhấn F6 sẽ có thông báo: 1 file(s) is copied trên màn hình, nếu như tệp không được tạo vì một lý do nào đó thì dòng thông báo sẽ là 0 file(s) is copied)

Ví dụ: C:\>COPY CON BAITHO.TXT Tạo tệp BAITHO.TXT trên ổ đĩa C

✦. Lệnh xem nội dung tập tin (TYPE):

Lệnh dùng để xem (hiển thị) nội dung một tệp tin trên màn hình.

TYPE <Đường dẫn>\<TỆP dung nội xem muốn tin> ↵

Sau khi nhập đúng đường dẫn và tập tin thì nội dung sẽ được hiển thị trên màn hình.

Nếu như trong cú pháp trên không đúng hoặc sai tên tệp tin, đường dẫn thì sẽ có dòng thông báo lỗi hiển thị ví dụ như:

- Bad command or filename
- Invalid drive Specification
- Path not found - ...
- Requient parameter missing

➤ Ví dụ: C:\>TYPE BAITHO.TXT Dùng để xem nội dung tập tin BAITHO.TXT

Trên Ổ đĩa C.

✦ **Đổi tên tệp tin(REN):**

Thay đổi tên file còn nội dung thì giữ nguyên.

REN [d:][path][fileName] <↵

➤ *Ví dụ:* C:\REN VANBAN\THUVIEN.DOC \VANBAN\HOPDONG.TXT ↵

Đổi tên file THUVIEN.DOC thành file HOPDONG.TXT nằm trong cùng một thư mục.

✦ **Xoá nội dung tệp tin(DEL):**

DEL [ổ đĩa][đường dẫn][tên tệp cần xoá] <↵

VD: C:\DEL C:\VANBAN\HOPDONG.TXT

Xoá tên file HOPDONG.TXT trong thư mục VANBAN ở ổ đĩa C:

## II. **Lệnh ngoại trú**

Là những lệnh thi hành chức năng nào đó của HĐH nhưng ít được sử dụng và đờ tồn bộ nhớ của máy người ta lưu trữ nó trên đĩa dưới dạng các tệp tin có phần mở rộng là: COM hoặc EXE

[d:] [path] [] [<TUY chọn>]

[d:] [path]: ổ đĩa đường dẫn đến tên lệnh

: là tên chính của tên tệp tin chương trình

### ✦ **Lệnh định dạng đĩa (FORMAT)**

Tạo dạng cho đĩa mềm hay đĩa cứng ...

[d:] [path] Format [d1] [/tham số]

[d:][Path]: ổ đĩa đường dẫn đến tên lệnh

[d:]: Tên ổ đĩa cần định dạng

Tham số:

/s: Tạo đĩa hệ thống.

/u: format mà sau đó không thể sử dụng lệnh UNFORMAT để lấy lại dữ liệu.

/q: định dạng nhanh

➤ *Ví dụ:* Định dạng đĩa mềm trong ổ đĩa A theo đúng dung lượng của ổ đĩa và sao chép các tệp cần thiết để khởi động máy vào đĩa.

C:\FORMAT A: /S <↵

### ✦ **Lệnh sao chép các tệp tin hệ thống:**

Tác dụng: cho phép chép các tệp tin hệ thống.

[d:][Path]sys [d1:]

*Ví dụ:* C:\sys A:

### ✦ **Lệnh phục hồi đĩa(UNDELETE)**

Phục hồi đĩa bị xoá bởi lệnh định dạng đĩa FORMAT

[d:][Path] UNDELETE [d1][path1][file Name]

### ✦ **Lệnh kiểm tra đĩa(CHKDSK):**

Kiểm tra đĩa và thông báo tình trạng đĩa.

CHKDSK[/F]

Tham số /F sẽ hiển thị số Sector bị hỏng khi kết thúc quá trình kiểm tra.

➤ *Ví dụ:* CHKDSK C:

### ✦ **Lệnh SCANDISK :**

Lệnh này dùng để kiểm tra cấu trúc tệp của đĩa và sau đó tiến hành kiểm tra các lỗi vật lý trên bề mặt đĩa.

SCANDISK tên ổ đĩa <↵

*Ví dụ:* SCANDISK A: <↵

### ✦ **Tệp lệnh bó:**

Lệnh bó thực chất là một tệp trong đó liệt kê thứ tự thực hiện các lệnh được liệt kê trong tệp. Trong số các tệp lệnh bó, quan trọng nhất bao gồm hai tệp sau:

Tệp AUTOEXEC.BAT Tệp lệnh Autoexec.bat là một tệp lệnh đặc biệt nằm ở thư mục gốc ổ đĩa khởi động. Khi khởi động hệ điều hành, các lệnh trong tệp Autoexec.bat sẽ thực hiện theo tuần tự.

➤ *Ví dụ:*

@ECHO - Lệnh hiển thị một dòng trắng trên màn hình

SMARTDRV.EXE - Gọi tệp tạo vùng đệm cho đĩa

MSCDEX /D:MSCD001 /V - Gọi tệp khởi tạo chương trình điều khiển ổ CD trên DOS.

SET BLASTER=A220 I5 D0 P300 - Lệnh đặt đường điều khiển âm thanh trên DOS ra loa.

SET PATH=%PATH%;C:\PROGRA~1\COMMON~1\AUTODE~1 - Lệnh thiết đặt đường dẫn mặc định khi tìm tệp.

Tệp CONFIG.SYS

Tệp lệnh CONFIG.SYS là tệp được gọi chạy đầu tiên của hệ điều hành khi khởi động máy. Sau khi nhận được lệnh khởi động máy từ ROM-BIOS, hệ điều hành sẽ lần lượt gọi chạy các tệp COMMAND.COM, MS-DOS.SYS, IO.SYS. Sau khi quá trình trên kết thúc, hệ điều hành sẽ tìm kiếm trên thư mục gốc ổ đĩa khởi động xem có tệp CONFIG.SYS hay không, nếu có nó sẽ tuần tự thực hiện các lệnh trong tệp đó, nếu có/không thì sau đó vẫn tiếp tục kiểm tra xem có tệp AUTOEXEC.BAT hay không, nếu có thì cũng tuần tự thực hiện các lệnh có trong tệp này, nếu có/không thì kết thúc quá trình khởi động và trả lại quyền điều khiển cho người sử dụng.

➤ *Ví dụ:* Nội dung của một tệp CONFIG.SYS

DEVICE=C:\WINDOWS\HIMEM.SYS - Nạp chương trình khởi tạo và điều khiển bộ nhớ mở rộng

DEVICE=C:\CDPRO\VIDE-CDD.SYS /D:MSCD001 Nạp chương trình khởi tạo và điều khiển ổ CD

FILES=90 - Quy định số tệp được mở đồng thời tối đa

BUFFERS=40 - Quy định số bộ nhớ đệm cho mỗi lần mở tệp

### III. VIRUS tin học

#### 3.1. Khái niệm VIRUS

VIRUS tin học là một chương trình máy tính do con người tạo ra nhằm thực hiện ý đồ nào đó.

Các chương trình này có đặc điểm:

- Kích thước nhỏ.

- Có khả năng lây lan, tức là tự sao chép chính nó lên các thiết bị lưu trữ dữ liệu như đĩa cứng, đĩa mềm, băng từ ...

- Hoạt động ngầm: hầu như người sử dụng không thể nhận biết được sự thực hiện của một chương trình VIRUS vì kích thước của nó nhỏ, thời gian thực hiện nhanh và người viết VIRUS luôn tìm cách che dấu sự hiện diện của nó. VIRUS nằm thường trú ở bộ nhớ trong để tiến hành lây lan và phá hoại. Hầu hết các VIRUS đều thực hiện công việc phá hoại như ghi đè lên các tệp dữ liệu, phá hỏng bảng FAT, khóa chế bàn phím, sửa đổi cấu hình hệ thống, chiếm vùng nhớ trong.

Tuỳ theo nguyên tắc hoạt động, có thể chia VIRUS thành hai loại:

- Boot VIRUS là các loại nhiễm vào Master Boot và Boot Sector. Những virus này có thể làm máy tính không khởi động được, làm mất hết dữ liệu trên đĩa cứng, thậm chí không khởi tạo được đĩa cứng.

- File virus là loại nhiễm vào các tệp chương trình có đuôi EXE và COM. VIRUS này làm các chương trình chạy sai hoặc không chạy. VIRUS thường nối thân của mình vào đầu hoặc cuối tệp chương trình, như vậy kích thước tệp tăng khi nhiễm.

- Nếu một đĩa mềm có VIRUS ta cho đĩa này vào máy có ổ cứng sạch thì ổ cứng của máy này sẽ bị nhiễm VIRUS. Nếu một máy tính có ổ cứng đã bị nhiễm VIRUS, ta cho một đĩa mềm sạch vào ổ A và chạy nhưng không đóng nút chống ghi lại thì đĩa mềm sẽ bị nhiễm VIRUS, từ đĩa mềm này khi ta mang đĩa mềm sang các máy khác để chạy thì VIRUS sẽ lan sang máy khác.

#### 3.2. Nguyên tắc phòng ngừa VIRUS.

Vì vật trung gian để lây VIRUS là đĩa mềm, để phòng VIRUS ta phải rất hạn chế dùng một đĩa mềm lạ. Nếu bắt buộc phải dùng thì ta kiểm tra VIRUS đĩa mềm lạ bằng các chương trình chống VIRUS trước khi sử dụng. Song điều đó không thể hoàn toàn tin tưởng vì các chương trình chống VIRUS chỉ có khả năng phát hiện và diệt những VIRUS mà chúng đã biết. Các VIRUS mới không phát hiện được. Các chương trình chống VIRUS của nước ngoài không thể phát hiện các VIRUS sản xuất trong nước. Cần kết hợp nhiều chương trình chống VIRUS và luôn cập nhật chương trình mới nhất.

Khi mang đĩa mềm của mình đi chạy ở các nơi khác có ổ cứng thì phải bật lẫy chống ghi để tránh

VIRUS xâm nhập vào đĩa. Hết sức lưu ý khi ghi thông tin từ máy khác vào đĩa của mình. Nếu trên máy có nhiều người sử dụng thì trước khi làm việc ta nên sử dụng các chương trình chống VIRUS để kiểm tra VIRUS trên đĩa cứng. Để phòng chống nên có một hệ thống sạch (không có virus) để khởi động máy từ ổ A, đĩa này ngoài các tệp cần thiết để khởi động máy còn cần có các tệp của DOS như: Format.com, Fdisk.exe, unformat.com, Undelete.com, Scandisk.exe. Đồng thời ta phải có các đĩa mềm chứa các chương trình chống virus với các phiên bản mới nhất bộ SCAN, FPROT, ATV, BKAV,D2... các chương trình này cũng phải lấy từ nguồn đáng tin cậy, các đĩa mềm luôn được dán nhãn bảo vệ.

### 3.3. Một số triệu chứng khi máy nhiễm virus

- Một số tệp có đuôi COM và EXE tự nhiên bị tăng thêm số byte, khi đó ta nghĩ máy nhiễm F-virus. Để biết điều đó ta nhờ kích thước của một số tệp quan trọng: command.com 54645 byte (của DOS6.22), foxpro.exe 411032 byte (forpro 2.6)
  - Tệp chương trình đuôi COM hoặc EXE không chạy hoặc sai.
  - Máy không khởi động được từ đĩa cứng hoặc không nhận biết được ổ cứng khi khởi động máy từ ổ đĩa mềm, khi đó ta nghi máy bị nhiễm B-virus.
  - Máy chạy bị treo.
- Tất nhiên các triệu chứng trên còn có thể là do lỗi phần cứng.

### 3.4. Cách xử lý khi máy bị nhiễm virus.

Khi máy bị nhiễm VIRUS chúng ta tiến hành các bước sau:

1. Tắt hoàn toàn máy tính để loại virus ra khỏi bộ nhớ trong. Khởi động lại máy bằng đĩa mềm hệ thống từ ổ A.
2. Sau khi thực hiện xong bước một máy nhận biết được ổ C thì thực hiện bước 3. Nếu máy không nhận được ổ C thì thực hiện bước 4:
3. Chạy các chương trình kiểm tra và diệt virus. Sau khi kết thúc quá trình trên thì khởi động lại máy từ ổ cứng và làm việc bình thường.
4. Chạy chương trình kiểm tra và sửa đĩa nếu như cần giữ lại thông tin trên đĩa cứng (NDD.EXE). Sau khi sao lưu dữ liệu nên làm theo các bước sau:

- a. Chạy FDISK.EXE để khởi tạo lại bảng Partition (FAT) cho ổ cứng.
  - b. Chạy FORMAT.COM C: /S để định dạng lại ổ đĩa.
  - c. Cài lại hệ điều hành và ứng dụng cần thiết.
  - d. Sao dữ liệu lại ổ đĩa và làm việc bình thường.
- Nếu như trên ổ đĩa không cần sao lưu dữ liệu lại thì có thể chạy ngay FDISK.EXE mà không cần chạy qua NDD.EXE.

## BÀI 3:Norton Commander (NC)

### ➡1. NC là gì?

NC là phần mềm quản lý tệp và thư mục, cho phép sử dụng các lệnh cơ bản của DOS dưới dạng nhanh chóng, hiệu quả hơn.

### ➡2.Phương pháp làm việc:

- Sử dụng phím chức năng để thi hành lệnh, Sử dụng hộp thoại để giao tiếp với người dùng.

- Sử dụng hình ảnh trực quan.
- Hỗ trợ việc sử dụng chuột trong các hoạt động của chương trình.
- Có phần hướng dẫn sử dụng (F1).

Để sử dụng được NC ít nhất ta phải có các tập tin sau:

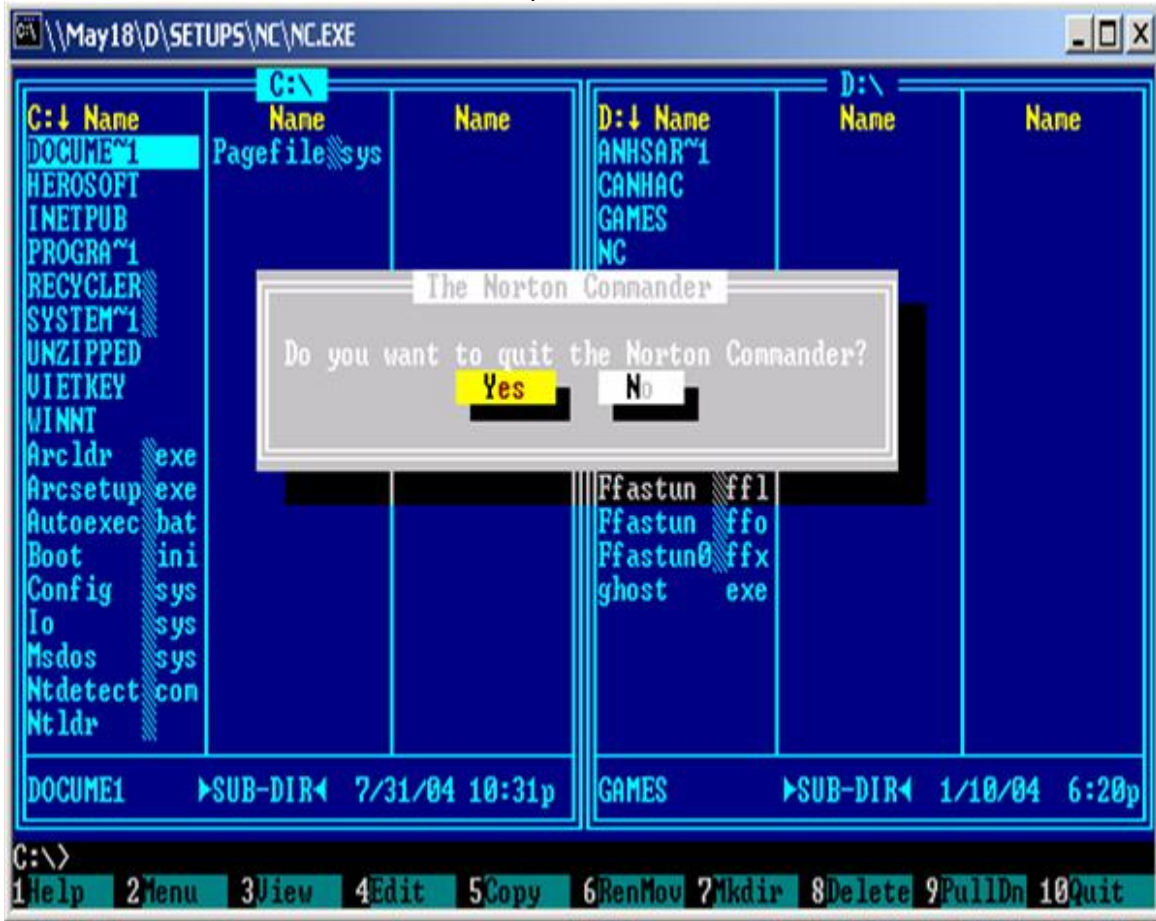
- NC.EXE
- NCMAIN.EXE
- NC.HLP
- NC.INI

### ➔ 3. Khởi động và thoát khỏi NC:

**3.1. Khởi động NC:** Chương trình NC.EXE thường được đặt trong thư mục NC của ổ đĩa C  
Để khởi động NC ta chạy tập tin NC.EXE từ dấu nhắc lệnh của MS-DOS như sau:

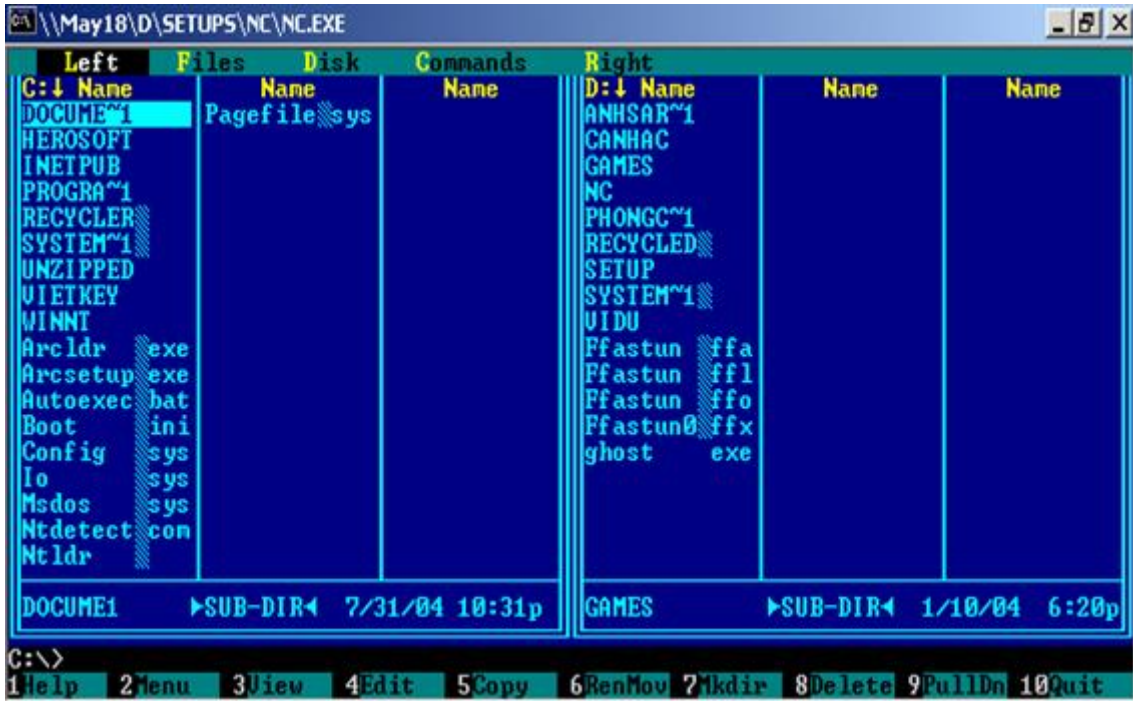
```
C:\>NC\NC
```

**3.2. Thoát khỏi NC:** Để thoát khỏi NC ta ấn phím F10 rồi chọn YES



### 3.3. Màn hình làm việc của NC:

Khi khởi động NC sẽ hiển thị như sau:



Dòng dưới cùng thể hiện một số phím chức năng tắt tương ứng với các phím từ F1 đến F10 và tổ hợp CTRL + Fn, ALT + Fn, (trong đó n = 1..10 ). trong đó:

- Lệnh F1: (Help)Lệnh trợ giúp
- Lệnh F2: (Menu)Lệnh tạo menu cho người sử dụng
- Lệnh F3: (View) Lệnh xem nội dung tệp
- Lệnh F4: (Edit)Lệnh xem và sửa nội dung tệp
- Lệnh F5: (Copy)Lệnh sao chép (copy)
- Lệnh F6: (RenMov)Lệnh đổi tên và di chuyển tập tin
- Lệnh F7: (MkDir)Lệnh tạo thư mục
- Lệnh F8: (Delete)Lệnh xoá thư mục và tập tin
- Lệnh F9: (PullDn)Truy cập thanh menu của NC
- Lệnh F10: (Quit)Thoát khỏi NC.
- Lệnh ATL + F1: Lệnh chuyển đổi ổ đĩa bên cửa sổ bên trái.
- Lệnh ATL + F2: Lệnh chuyển đổi ổ đĩa bên cửa sổ bên phải.
- Lệnh ATL + F3: Lệnh xem nội dung tập tin



- Lệnh ATL + F4: Lệnh soạn thảo tập tin
- Lệnh ATL + F5: Lệnh nén các tập tin
- Lệnh ATL + F6: Lệnh giải nén tập tin nén
- Lệnh ATL + F7: Lệnh tìm kiếm
- Lệnh ATL + F8: Lệnh hiển thị các dòng lệnh sử dụng gần đây nhất
- Lệnh ATL + F9: Lệnh chuyển đổi màn hình về dạng 40 dòng hoặc 25 dòng.

Lệnh ATL + F10: Lệnh hiển thị cây thư mục của ổ đĩa hiện thời.

- Lệnh CTRL + F1: Tắt mở khung panel bên trái
- Lệnh CTRL + F2: Tắt mở khung panel bên phải
- Lệnh CTRL + F3: Sắp xếp tệp và thư mục theo tên
- Lệnh CTRL + F4: Sắp xếp tệp và thư mục theo phần mở rộng
- Lệnh CTRL + F5: Sắp xếp tệp và thư mục theo thời gian tạo
- Lệnh CTRL + F6: Sắp xếp tệp và thư mục theo kích thước tệp
- Lệnh CTRL + F7: Bỏ chế độ Sắp xếp
- Lệnh CTRL + F8: So sánh 2 thư mục với nhau
- Lệnh CTRL + F9: In tệp
- Lệnh CTRL + F10: Chia hoặc trộn tệp

Ngoài các tổ hợp phím trên, NC còn chứa nhiều lệnh khác trên thanh menu trở xuống khi bạn nhấn phím F9.

Tất cả các công tác làm việc với tệp, trước tiên bạn phải dùng các phím di chuyển để di chuyển thanh sáng đến tệp cần làm việc. Dùng phím TAB để di chuyển thanh sáng qua lại giữa 2 khung Panel, dùng phím Insert để chọn các tệp cần làm việc với các lệnh như Copy, Move, Delete ... Dùng phím (+) bên phím số để chọn tất cả các tệp và thư mục trong thư mục hiện thời, phím (-) để bỏ chọn, phím (\*) để đổi chọn. Các nguyên tắc làm việc với thư mục giống như trên MS-DOS, tuy nhiên cần chú ý rằng chương trình đã tạo ra một giao diện thân thiện hơn để ta làm việc vì vậy mà có một số nguyên tắc làm việc trong MS-DOS có thể bỏ qua trong NC. Việc chuyển đổi thư mục hiện hành rất đơn giản, bạn chỉ việc di chuyển thanh sáng đến thư mục cần vào và nhấn ENTER, muốn thoát khỏi thư mục trở về thư mục cấp cao hơn, hãy di



chuyển thanh sáng về dòng có hai dấu chấm (..) và nhấn ENTER.

#### IV. Một số ví dụ về cách thực hiện các lệnh trong NC

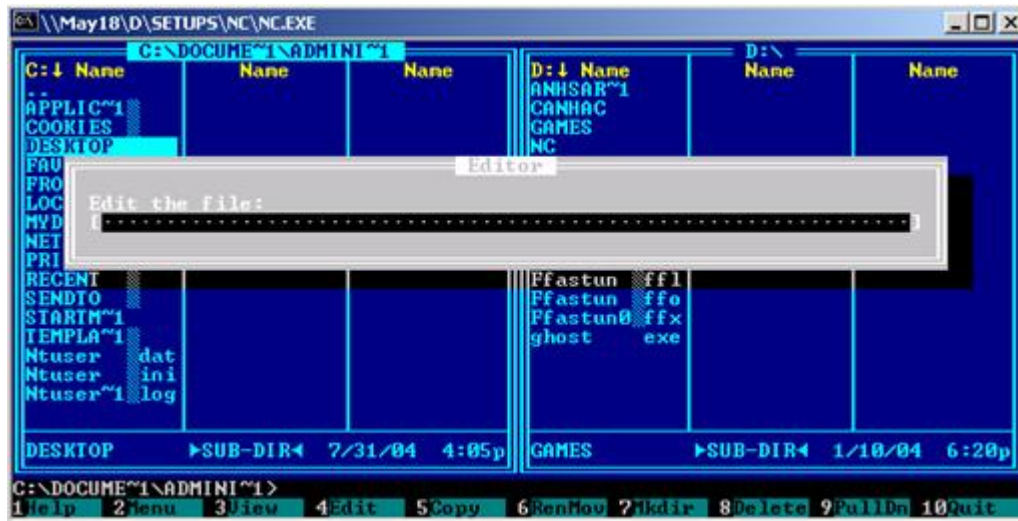
##### 4.1. Tạo mới một thư mục:



- Di chuyển thanh sáng đến vị trí cần tạo thư mục
- Nhấn F7
- Nhập tên thư mục cần tạo sau đó nhấn ENTER.

##### 4.2. Tạo mới tệp:

- Di chuyển thanh sáng đến vị trí cần tạo tệp
- Nhấn tổ hợp phím SHIFT+F4 và nhập tên tệp cần tạo



- NC sẽ mở ra cho bạn một khung hội thoại để nhập nội dung tệp, khi nhập xong bạn nhấn phím F2 để ghi lại nội dung và ấn ESC để thoát.

##### 4.3. Sao chép, di chuyển:

- Chọn các tệp, thư mục muốn sao chép hoặc di chuyển trên một khung panel
- Trên khung panel còn lại dùng thanh sáng di chuyển đến vị trí cần sao chép hoặc di chuyển đến.
- Di chuyển thanh sáng về khung panel chứa các tệp được chọn
- Nhấn F5 (copy) hoặc F6 (move)

#### 4.4. Nén các tập tin:

- Chọn các tập cần nén ở trên một khung Panel, trên khung panel còn lại di chuyển thanh sáng vào trong thư mục muốn để tập đích.
- Di chuyển thanh sáng về khung panel có tập chọn
- Nhấn ALT + F5
- Đưa tên tập đích vào (không cần phần mở rộng)

#### 4.5. Giải nén tập:

- Chọn tập cần giải nén ở trên một khung Panel, trên khung panel còn lại di chuyển thanh sáng vào trong thư mục muốn để các tập sau khi giải nén.
- Di chuyển thanh sáng về khung panel có tập chọn
- Nhấn ALT + F6

---

#### . Chia một tập thành nhiều tập:

- Chọn tập cần chia ra ở trên một khung Panel, trên khung panel còn lại di chuyển thanh sáng vào trong thư mục muốn để các tập sau khi chia.
  - Di chuyển thanh sáng về khung panel có tập chọn
  - Nhấn CTRL + F10
  - Sau khi đưa số tập cần chia ra trong mục Number of splids chọn Splids.
- Các tập sau khi được chia sẽ giữ nguyên tên như tập đích và phần mở rộng mặc định là \*. 001, \*.002, ...

#### 4.7. Trộn nhiều tập được chia ra thành một tập duy nhất:

- Chọn các tập cần gộp lại ở trên một khung Panel, trên khung panel còn lại di chuyển thanh sáng vào trong thư mục muốn để các tập sau khi được gộp lại.
- Di chuyển thanh sáng về khung panel có tập chọn
- Nhấn CTRL + F10
- Sau khi đưa đường dẫn, tên tập và phần mở rộng của tập được gộp trong mục Merge n files to sau đó chọn Merge. (trong đó n là tổng số tập chọn để gộp lại).

#### 4.8. Xoá tập và thư mục:

- Chọn các tập và thư mục cần xoá
- Nhấn F8
- Một hộp hội thoại sẽ mở ra và hỏi xem chắc chắn bạn muốn xoá các tập hoặc thư mục hay không, nếu chắc chắn các bạn chọn DELETE.



#### 4.9. In tệp:

- Di chuyển thanh sáng đến tệp cần in
- Nhấn CTRL + F9
- Chọn OK



#### 4.10. Tìm kiếm tệp:



- Di chuyển thanh sáng về ổ đĩa cần tìm kiếm
- Nhấn ALT + F7
- Đưa tên tệp cần tìm trong Find Files và chọn Start để bắt đầu. Nếu tìm thấy tệp thì trên danh sách phía trên sẽ hiển thị tên tệp, vị trí của tệp.

## BÀI 4: TỔNG QUAN

### I. Sử dụng Keyboard (Bàn phím) và Mouse (Chuột)

#### 1.1 - Keyboard

Bàn phím phổ biến hiện nay có từ 101 phím đến 105 phím. Trên bàn phím có 3 vùng từ trái sang phải như sau:

Nhóm phím chức năng: Gồm các phím

|                          |                                                                                                                       |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------|
| F1,F2,...F12             | Mỗi phím có một chức năng nhất định tùy theo phần mềm                                                                 |
| Printscreen              | Dùng để in nội dung từ màn hình ra máy in                                                                             |
| Pause                    | Tạm ngưng một hoạt động của máy                                                                                       |
| Capslock                 | Tắt mở chế độ gõ chữ in hoa                                                                                           |
| Shift trái và Shift phải | Một ký tự được gõ trong khi in ấn giữ phím Shift sẽ được in ra chữ hoa hoặc in ra ký tự ở trên nếu phím đó có 2 ký tự |
| Backspace                | Xoá lùi về bên trái                                                                                                   |
| Delete                   | Xoá ký tự tại vị trí con trỏ                                                                                          |
| Enter                    | Xuống dòng hoặc kết thúc một công việc hoặc thực hiện một công việc.                                                  |

Nhóm phím ký tự: Trên mỗi phím có thể được in một hoặc hai ký tự. Trong trường hợp trên phím có hai ký tự nếu muốn có ký tự phía trên thì giữ SHIFT trong khi gõ ký tự đó.

Nhóm phím điều khiển:

Các phím mũi tên ← ↑↓→ , Tab: điều khiển con trỏ

Home, End, Page Up, Page Down, Ctrl, Atl, Del,...

#### 1.2 - Mouse - Chuột

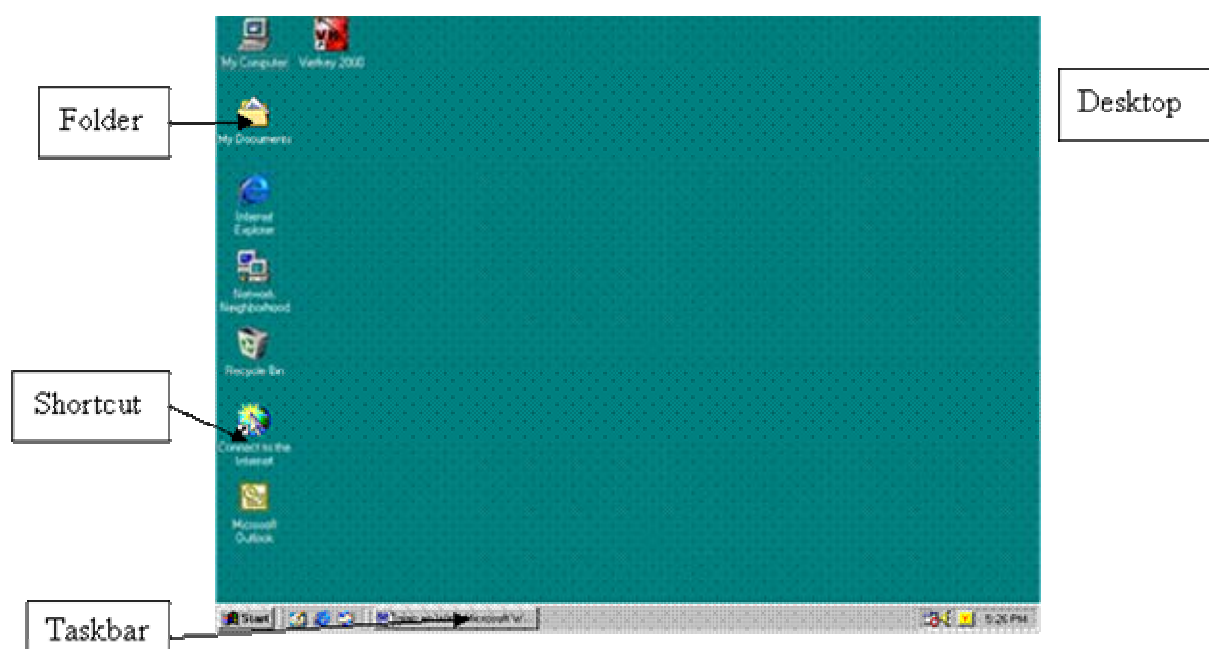
Các thao tác cơ bản của chuột - Mouse

| Thao tác        | Công dụng                                            |
|-----------------|------------------------------------------------------|
| Con trỏ (Point) | Di chuyển để con trỏ Mouse tới đúng vị trí mong muốn |

|                             |                                                                                                 |
|-----------------------------|-------------------------------------------------------------------------------------------------|
| Nhấp (Click)                | Nhấp nhả nút trái của chuột, thường dùng để chọn lệnh cần thi hành                              |
| Nhấp nút phải (Right Click) | Nhấp nhả nút phải chuột                                                                         |
| Nhấp đúp (Double Click)     | Nhấp nhả nút trái của chuột 2 lần với tốc độ nhanh                                              |
| Kéo (Drag)                  | Trỏ chuột đúng vị trí, bấm giữ nút trái của chuột đồng thời di chuyển chuột sang một vị trí mới |

## II- Khởi động Windows 98

Khi Microsoft Windows 98 sẽ hiện ra lúc ấy trên màn hình ta sẽ thấy có dạng tương tự như sau:



- Desktop: Vùng diện tích làm nền cho các mục trong Windows. Có thể tạo thêm các Folder và Shortcut khác để công việc sau này được thực hiện nhanh hơn.

- Shortcut: Các biểu tượng có hình dạng riêng với mũi tên đen nhỏ nằm ở góc dưới bên trái tượng trưng cho một chương trình ứng dụng, một tài liệu ...

- Folder: Có thể xem folder như một cặp tài liệu dùng để quản lý một chương trình ứng dụng

- Taskbar: Thanh hiển thị các cửa sổ chương trình đang được mở.

Ví dụ: Taskbar cho biết có ba chương trình cửa sổ ứng dụng đang được mở đó là MSWord Adobe Photoshop và MS Excel.

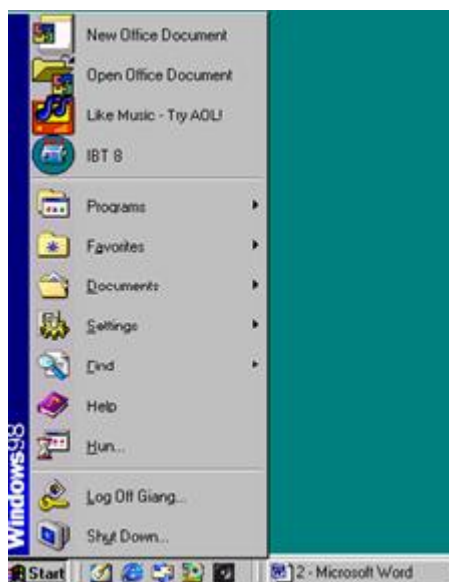


- Clock: Đồng hồ hệ thống báo cho chúng ta biết giờ hiện hành trong máy.

- Start button: Nhấp nút này để mở Menu Start

- Menu Start: Giúp khởi động chương trình nhanh hơn.

Chú ý: Có thể mở Menu Start bằng tổ hợp phím Ctrl+ ESC



### Thoát khỏi Windows 98

Nhấn nút Start để mở Menu start, nhấn chọn lệnh Shutdown... hoặc dùng tổ hợp phím Alt-F4



|                                      |                                                                                                                                                                                        |
|--------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Shutdown the computer?               | Thoát khỏi Windows bằng cách đóng tất cả các chương trình đang mở, sau đó hiện ra câu thông báo 'It's now safe to turn off your Computer'(Bây giờ có thể an toàn tắt máy tính của bạn) |
| Restart the Computer?                | Thoát khỏi chương trình bằng cách đóng tất cả các chương trình đang mở, sau đó tự khởi động lại máy tính.                                                                              |
| Restart the Computer in MS DOS mode? | Thoát khỏi Windows bằng cách đóng tất cả các chương trình đang mở và trở về dấu nhắc hệ thống của MS-D OS                                                                              |

### IV. Cửa sổ chương trình ứng dụng



Các chương trình ứng dụng khi chạy trong môi trường Windows sẽ được thể hiện trong một khung hình chữ nhật, đó là cửa sổ chương trình.

#### 4.1- Mở cửa sổ chương trình ứng dụng

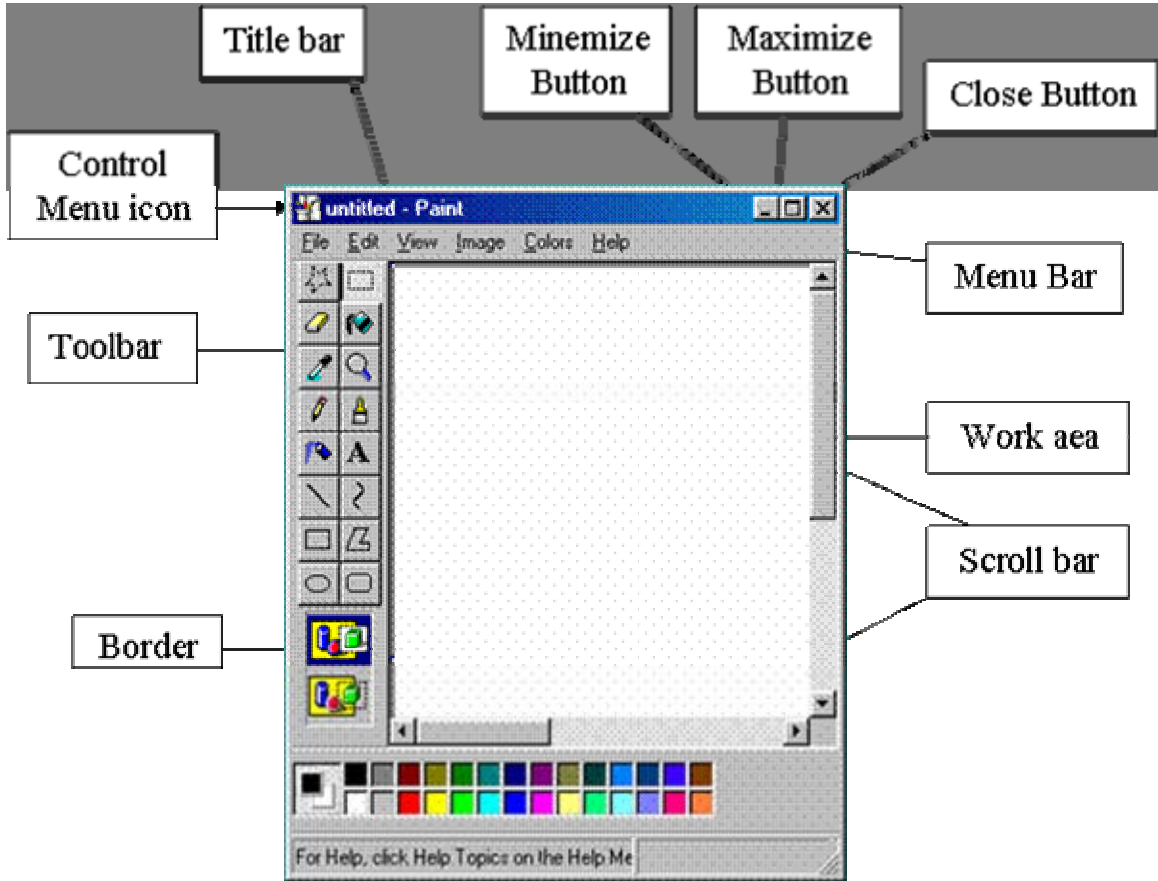
- Cách 1: Nhấp đúp một biểu tượng Shortcut trên nền Desktop

- Cách 2: Trên thanh Taskbar, ta nhấp nút Start để mở menu start, sau đó trỏ mouse vào từng Folder (Có ký hiệu hình tam giác ở bên phải) để mở ra một Sub menu (menu cũ có chứa các folder con và shortcut)... cuối cùng nhấp chọn tên một chương trình ứng dụng.



#### 4.2 - Các thành phần chính trong một cửa sổ

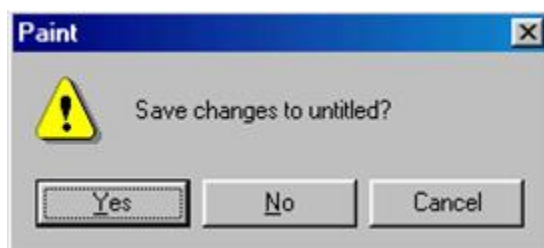
Một cửa sổ chương trình sẽ có các thành phần cơ bản sau (hình minh họa là cửa sổ chương trình paint dùng để vẽ hình)



| Thành phần        | Chức năng                                                                                                                              |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| Title Bar         | Thẻ hiện tên cửa sổ và tên tài liệu đang được mở. Nếu kích thước của cửa sổ nhỏ hơn khung nhìn, có thể kéo nó đi để di chuyển cửa sổ   |
| Borders           | Đường viền bao quang cửa sổ. Đường viền này chỉ hiện ra khi kích thước của cửa sổ nhỏ, có thể kéo nó để thay đổi kích thước của cửa sổ |
| Control Menu Icon | Nhấp vào nó sẽ hiện ra Menu gồm các lệnh dùng để điều khiển cửa sổ                                                                     |
| Close Button      | Nhấp vào nút này để đóng cửa sổ                                                                                                        |
| Minimize Button   | Nhấp vào nút này sẽ thu nhỏ cửa sổ thành biểu tượng nằm trên Taskbar                                                                   |
| Maximizer Button  | Nhấp vào nút này cửa sổ sẽ đạt kích thước tối đa                                                                                       |
| Restore Button    | Nút này chỉ xuất hiện sau khi cửa sổ đã đạt đến kích thước tối đa, nó có tác dụng trả lại kích thước ban đầu của cửa sổ                |
| Menu Bar          | Thanh chứa các lệnh theo kiểu liệt kê                                                                                                  |
| Work Area         | Vùng làm việc                                                                                                                          |
| Toolbar           | Các biểu tượng lệnh                                                                                                                    |
| Scroll Bar        | Thanh trượt để thể hiện vùng làm việc                                                                                                  |

### Đóng cửa sổ chương trình


- Cách 1: Nhấp Mouse vào biểu tượng Close Button
- Cách 2: Nhấp Mouse vào Menu File chọn lệnh Exit
- Cách 3: Dùng tổ hợp phím ALT + F4
- + Nhấp nút Yes - Đồng ý lưu (Phải cho thêm các thông tin cần thiết để biết sản phẩm tạo ra cất ở đâu, mang tên gì...)
- + Nút No - Không muốn lưu sản phẩm do mình tạo ra



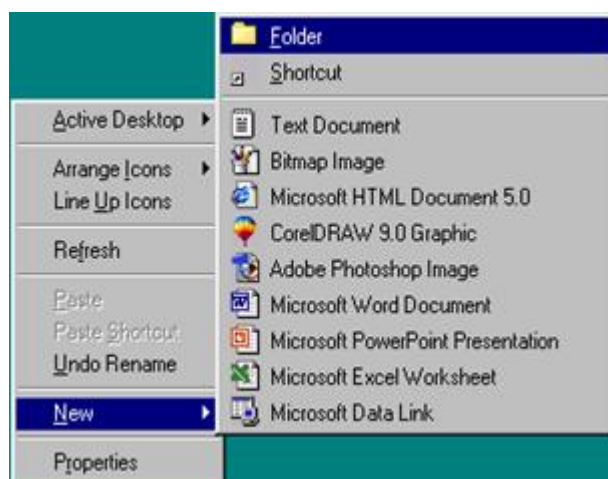
- + Nút Cancel - Huỷ bỏ lệnh lưu

### 📁 - Folder - Shortcut

#### 5.1- Tạo Folder - Thư mục

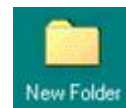
Folder được tạo ra để quản lý các Folder con và các Shortcuts có trong Folder đó. Trong Windows 9X Folder có biểu tượng  (tương tự như cặp đựng tài liệu)

##### a) Tạo Folder trên nền Desktop



Nhấp nút phải của Mouse ngay trên nền Desktop sẽ có ngay một Menu nhỏ hiện ra, sau đó nhấp vào lệnh New sẽ có một Flayout hiện ra và chọn lệnh tạo Folder

Chọn Folder ngay lập tức trên nền Desktop sẽ hiện ra biểu tượng



Sau đó đặt tên Folder bằng cách xoá 'New Folder', gõ tên mới vào. Ví dụ: gõ 'baitap'

\* Tạo Folder mới trong một Folder đã có trong nền Desktop

Bước 1: Nhấp đúp vào biểu tượng Folder đã có trên nền Desktop để mở cửa sổ Folder

Bước 2: Nhấp menu File, trở vào New và nhấp chọn Folder

Bước 3: Trong cửa sổ Folder(ví dụ Folder'baitap') hiện ra một folder con có tên là New Folder. Xoá bỏ tên này và bỏ tên mới vào(ví dụ đặt là 'btwin').

b). Tạo Folder trong My computer

Để mở ra cửa sổ này, cửa sổ sẽ cho chúng ta biết các thông tin về chính máy tính đang dùng



Nhấp menu File, trở vào New và chọn Folder, trong cửa sổ đĩa A: sẽ có thêm một biểu tượng Folder có tên New Folder, xoá bỏ tên này và gõ tên mới vào.

## 5.2 - Hiệu chỉnh Folder

### a). Di chuyển -Move

Kéo biểu tượng của Folder từ chỗ này sang chỗ khác.

### b). Đổi tên - Rename

Nhấp nút phải của Mouse vào biểu tượng của Folder để cho hiện menu đặc biệt, chọn lệnh Rename, vùng tên của Folder sẽ hiện ra con trỏ Text để chúng ta có thể xoá tên cũ và đặt tên mới.

### c). Chép - Copy

Cách 1: Nhấp nút phải của mouse vào biểu tượng của Folder nguồn chọn lệnh copy(Chép Folder vào bộ nhớ)

Nếu muốn thành một Folder cùng cấp thì nhấp nút phải của mouse vào vùng trống cho hiện menu đặc biệt, sau đó chọn lệnh Paste (dán Folder từ vị trí chúng ta cần). folder mới(folder đích) được chép sẽ có tên 'Copy of + tên Folder nguồn' để đảm bảo tính duy nhất của tên gọi (Hai Folder cùng tên không được ở cùng một chỗ) chúng ta nên dùng lệnh Rename đổi tên cho Folder mới được chép.

Nếu muốn chép vào một Folder khác thì nhấp nút phải vào biểu tượng đó, chọn lệnh Paste. Folder mới được chép sẽ là Folder con và có tên giống Folder nguồn.

Cách 2: Giữ phím Ctrl đồng thời kéo mouse từ vị trí nguồn sang vị trí đích (Biểu tượng khi được kéo đi sẽ có dấu +), nhả mouse trước, phím Ctrl nhả sau.

### d). Xóa

Cách 1: Chọn Folder cần xoá, gõ phím Delete

Cách 2: Chọn Folder cần xoá, kéo thả vào sọt rác Recycle Bin



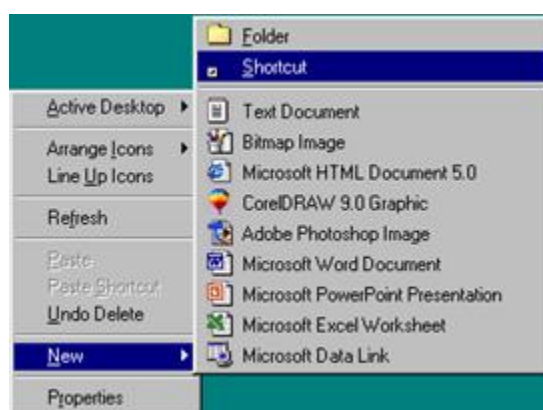
Trả lời bằng cách nhấp nút Yes để xoá hoặc nút No để không xoá

## Đường dẫn - PATH

Thông qua cách đặt Folder, chúng ta bắt đầu đặt ra câu hỏi: Folder được tạo ra ở đâu? trên Desktop, trong My computer, trong đĩa mềm A:, trong đĩa cứng C:... Nếu Folder con được tạo ra trong một Folder mẹ thì câu trả lời lại dài hơn, chẳng hạn như ví dụ chúng ta đã tạo một Folder con có tên là btwin nằm trong Folder baitap của Desktop... Đó chính là khái niệm về đường dẫn (path) - nó có tác dụng cho ta biết địa chỉ của một chủ thể đang ở đâu.

#### 5.4 - Cách tạo Shortcut

Shortcut là mối liên kết với 1 tập tin chương trình khác, nhờ đó ta có thể khởi động nhanh chương trình. các bước để tạo:



- Bước 1: Nhấp nút phải trên nền Desktop để hiện ra Menu trở Mouse vào New
- Bước 2: Nhấp chọn Shortcut , một hộp thoại (Dialog Box) có tên Creat Shortcut hiện ra



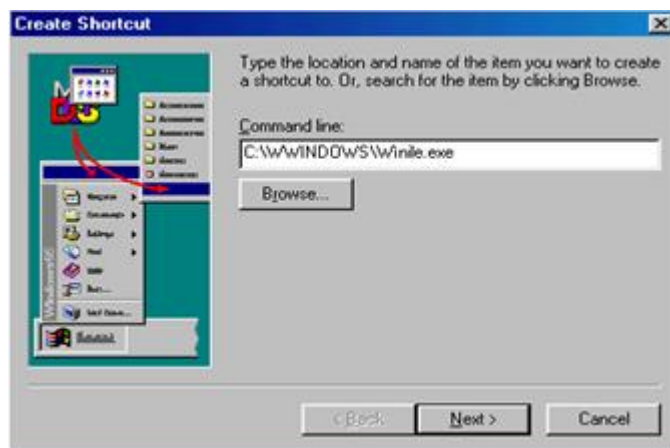
Trong mục Command line cần phải điền đầy đủ đường dẫn tới một tập tin chương trình, để thực hiện điều này cần sang Bước 3:



- Bước 3: Nhấp nút Browse... mở ra hộp thoại Browse để tìm chọn tên tập tin chương trình

Trong mục Look in, nhấp nút tam giác cho hiện ra bảng liệt kê để chọn một ổ đĩa, sau đó nhấp đúp vào folder và chọn File cần tạo Shortcut. Chọn nút Open. Trong hộp thoại Create Shortcut lúc này ở mục Command line đã hiện lên đầy đủ đường dẫn và tên tập tin chương trình

-Bước 4: Trong hộp thoại Create Shortcut nhấp nút Next hộp thoại mới Select a Title for the program



Sau đó nhấp nút Next và chọn nút lệnh Finish để kết thúc quá trình tạo Shortcut



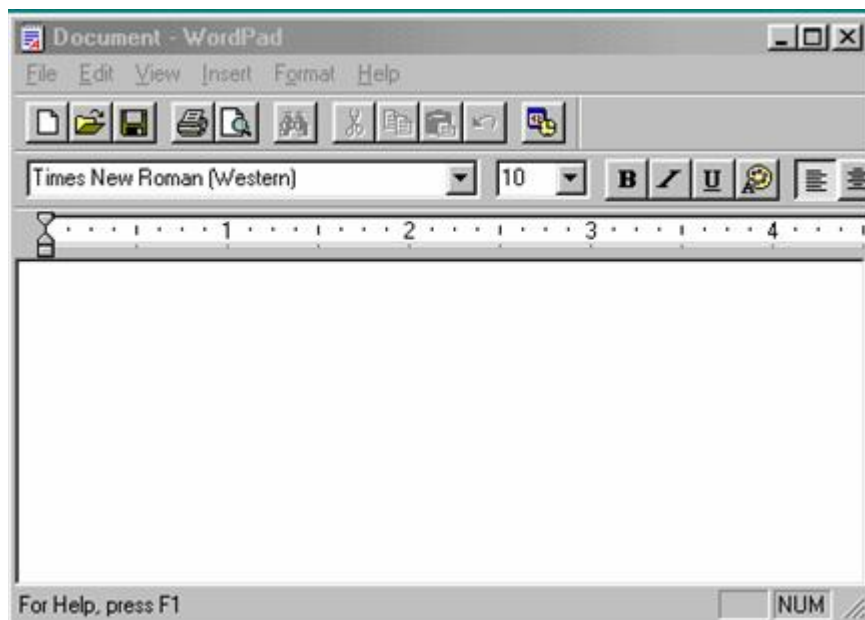
Như vậy một Shortcut mới được tạo ra trên nền Desktop mang biểu tượng sẽ có dạng

## VI - Chương trình soạn thảo Wordpad

Wordpad là một chương trình xử lý văn bản ở mức độ đơn giản

### 6.1 - Khởi động chương trình

Nhấp vào nút Start chọn Programs chọn Accessories nhấp chuột chọn Wordpad màn hình sẽ hiện ra như sau:





Ngoài các thành phần chung như đã nêu trên Wordpad còn có thêm một thước đo (Ruler) để canh đo văn bản.


## 6.2 - Soạn thảo văn bản

Mỗi khi chúng ta gõ vào một phím, trên màn hình sẽ hiện ra một chữ - Đó là phím ký tự

Nhiều ký tự hợp lại thành một từ (word) các từ được phân cách nhau bằng một khoảng trắng ( Gõ phím Space Bar)

Khi đang nhập văn bản mà gõ phím Enter thì đó là ngắt dòng và tạo ra một Paragraph mới

## 6.3 Lưu văn bản đã tạo vào đĩa


Sau khi gõ văn bản xong nhấp vào biểu tượng hình đĩa mềm  ( Save) hoặc vào Menu File chọn Save một hộp thoại hiện ra:

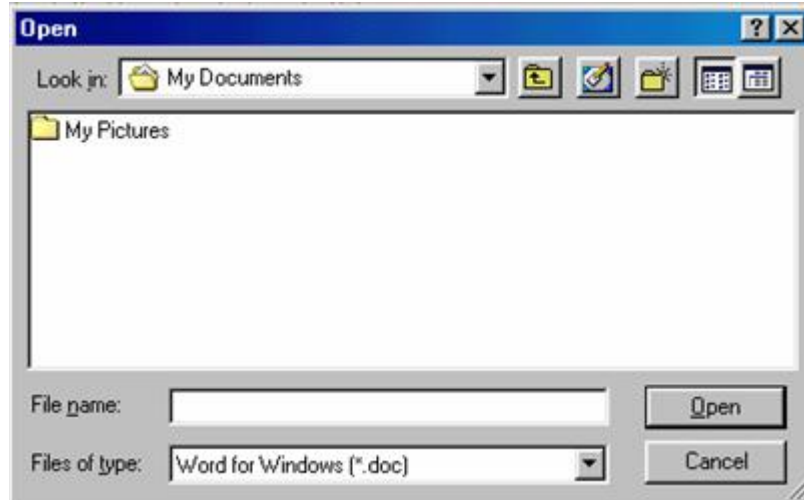


Mục Save in được ngầm định là Folder My document (Có thể chọn Folder khác)

Trong mục File name gõ tên ( ví dụ tên : 'hocwin' sau đó nhấn chuột vào nút Save, như vậy đoạn văn bản gõ đã được lưu với tên là hocwin nằm trong My Document

## 6.4 - Mở một văn bản đã có trong đĩa

Nhấp chuột vào biểu tượng  Open hoặc vào Menu File chọn Open sẽ hiện ra hộp thoại:



## BÀI 5: WINDOWS EXPLORER

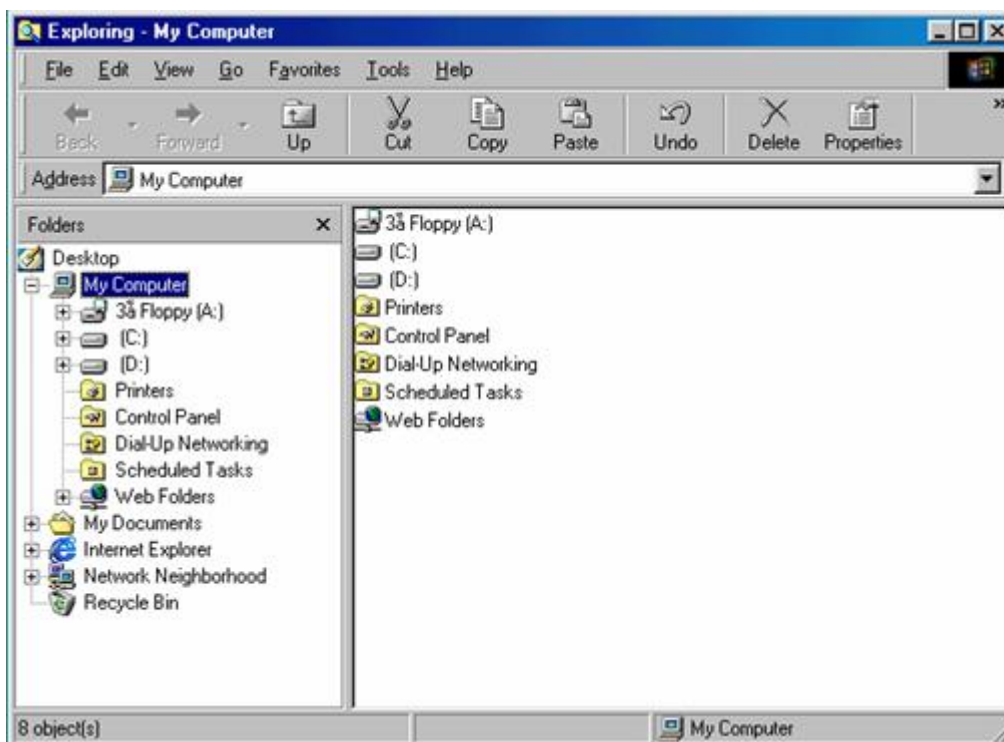
---

### →2.1. Khởi động Windows Explorer

Có thể được khởi động bằng nhiều cách:

- Cách 1: Nhấp nút Start, trỏ vào Programs và nhấp chọn Windows Explorer
- Cách 2: Trên nền Desktop, nhấp nút phải vào biểu tượng My computer để mở ra menu Object và chọn lệnh Explorer

Sau khi khởi động xong màn hình sẽ có được như sau:



## 2.2. Cửa sổ Windows Explorer

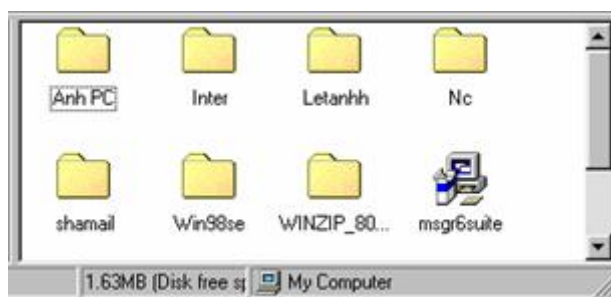
a). Cửa sổ Windows Explorer được chia thành 2 vùng tạm gọi là vùng bên trái và vùng bên phải của cửa sổ. Ranh giới giữa hai vùng được gọi là đường cắt (Split) - Kéo mouse ở đường này để thay đổi độ rộng giữa 2 vùng.

Vùng bên trái - All Folders - cho thấy cấu trúc thư mục của ổ đĩa cứng cùng với các tài nguyên có trong máy PC của bạn (My computer) bao gồm ổ đĩa mềm, ổ đĩa CD, các Folder hệ thống, mạng, print, Control panel, Recycle Bin

Vùng bên phải - Contents of (C:) - được thể hiện như một cửa sổ Folder thông thường và nó cũng có những chức năng tương tự như cửa sổ Folder

b). Thể hiện cửa sổ Windows Explorer

Cửa sổ Windows có thể thay đổi cách thể hiện cho vùng bên phải bằng cách nhấp các biểu tượng tương ứng trên Tool bar Hoặc nhấp Menu View chọn các lệnh:



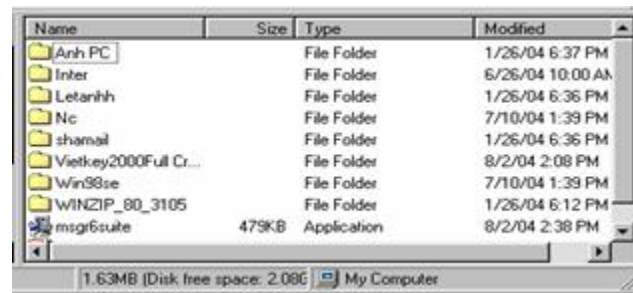
➤ Small Icons



➤ List



➤ Details



- Các lệnh Edit

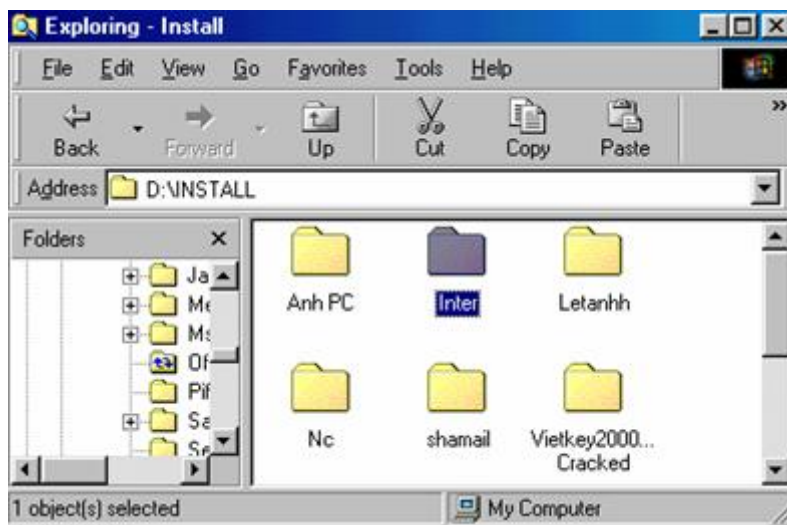
a) Đánh dấu chọn Folder và File

● Chọn một Folder hoặc một tập tin

- Nhấp chuột vào biểu tượng hoặc tên của biểu tượng

- Gõ gõ ký tự đầu tiên

- Dùng mũi tên để di chuyển thanh sáng



● Chọn một nhóm

- Chọn rời rạc: Giữ phím Ctrl - nhấp chọn từng biểu tượng
- Chọn liên tiếp: nhấp biểu tượng đầu , giữ Shift chọn biểu tượng cuối
- Chọn tất cả: Nhấp Menu Edit chọn Select All hoặc nhấn tổ hợp phím Ctrl +A

Chú ý: Dòng trạng thái sẽ hiện lên số chủ đề được chọn và tổng dung lượng của nó. Để nhả chọn chỉ cần nhấp chuột vào một biểu tượng.

b) Tạo Folder mới

Vào Menu File chọn New và chọn Folder, xoá tên New folder và gõ lại tên mới và nhấn Enter

Tạo Folder con: Để tạo một Folder con trước hết phải nhấp vào Folder cha ở vùng bên trái, sau đó nhấp chuột vào Menu File chọn New và chọn lệnh Folder.

c) Đổi tên

- Cách 1: Nhấp nút phải vào biểu tượng Folder ( hoặc File) để hiện ra Menu Object chọn lệnh Rename, xoá tên cũ và gõ tên mới và nhấn Enter để kết thúc
- Cách 2: Chọn Folder và gõ F2, xóa tên cũ và đặt tên mới.

d) Di chuyển (Move) sao chép (Copy) Folder, File, Shortcuts

● Đánh dấu các mục cần thiết sau đó thực hiện các thao tác sau:

- Cách 1: Đặt con trỏ ở vị trí nguồn, kéo chuột bằng nút phải từ nguồn sang đích nhả chuột ở vị trí đích, một Menu Object hiện ra và chọn lệnh Copy here ( Copy tới đây), Move here (di chuyển tới đây) hoặc Create Shortcut Here ( tạo shortcut ở đây).



- Cách 2: Sử dụng nút trái của chuột: Nhấp nút trái chuột


+ Move: Giữ phím Shift và kéo biểu tượng đã chọn từ nguồn tới đích

+ Copy: Giữ phím Ctrl và kéo biểu tượng đã chọn từ nguồn tới đích

- Cách 3: Dùng Toolbar ( hoặc Menu Edit)

Đánh dấu các folder hay File nguồn: Thực hiện lệnh di chuyển, (sao chép): Nhấp vào biểu tượng

lệnh Cut  ( copy  ) hoặc nhấp vào Menu Edit chọn Cut (Copy) để chuyển các

mục vào bộ nhớ, sau đó chọn Folder đích nhấp vào biểu tượng  Paste ( hoặc nhấp vào Menu Edit chọn lệnh Paste) để dán các mục từ trong bộ nhớ vào Folder đích.

Chú ý: Nếu trong lệnh sao chép một mục mà Folder nguồn trùng với Folder đích thì chủ đề mới được sao chép sẽ có tên " Copy of ... " và chúng ta nên đổi tên này bằng cách gõ tên mới vào rồi ấn Enter. Nếu việc thực hiện lệnh Move, Copy, Rename là không phù hợp với yêu cầu thì ta có thể dùng lệnh Undo để bỏ qua một lệnh vừa thi hành, bằng cách nhấp vào Menu Edit chọn lệnh

Undo hoặc nhấp biểu tượng lệnh Undo trên Toolbar



e) Xoá Folder, File, Shortcut

● Chọn các mục cần xoá và thực hiện các thao tác sau:

- Cách 1: Gõ phím Delete

- Cách 2: Nhấp chuột phải vào mục chọn và chọn lệnh Delete

- Cách 3: Nhấp chuột vào Menu File chọn Delete

- Cách 4: Kéo các mục chọn vào biểu tượng sọt rác Recycle Bin



Chú ý: Khi xoá các đối tượng Windows sẽ hiện ra bảng thông báo sau:

Nếu chọn Yes thì xoá, còn chọn No thì huỷ bỏ việc xoá

Phục hồi Folder, File, Shortcut đã bị xoá trong đĩa cứng: Chọn biểu tượng Recycle Bin ta sẽ có dạng sau:



Chọn Folder, File, Shortcut cần phục hồi, nhấp chuột vào Menu File chọn Restore

- Làm rỗng thùng rác:



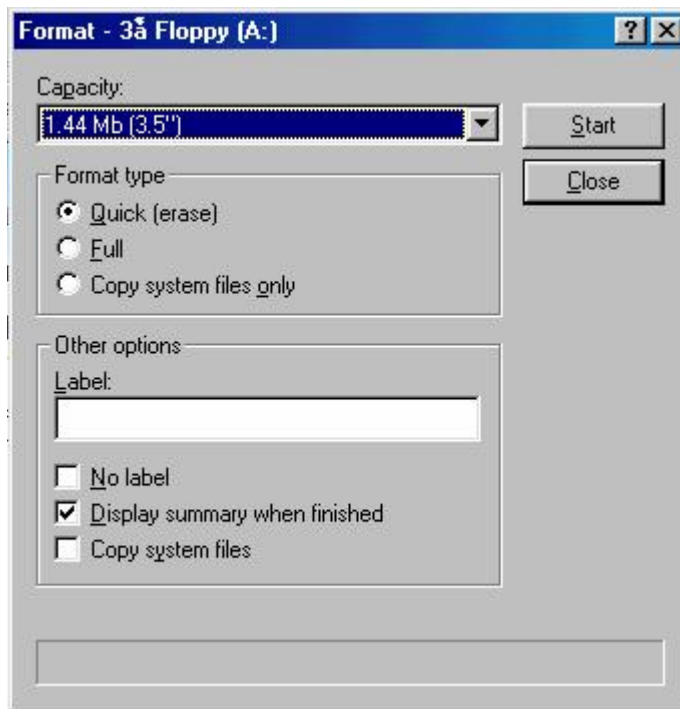
Nhấp vào Menu File chọn Empty Recycle Bin, một hộp thoại hiện ra , nếu chọn Yes thì thùng rác sẽ là trống rỗng còn chọn No thì huỷ bỏ lệnh.

## ➡ 2.4 - Định dạng đĩa mềm

Nhấp nút phải vào biểu tượng ổ đĩa mềm và chọn lệnh Format

Chọn Start để tiến hành Format.

- Capacity: Dung lượng đĩa mềm
- Format Type: Chọn cách định dạng đĩa
- Quick Bỏ qua các bước kiểm tra đĩa và chỉ thực hiện việc xoá tất cả các tập tin đã có trên đĩa mềm.
- Full: Định dạng đĩa và thực hiện việc kiểm tra để biết số sector bị hỏng trên đĩa.
- Label: Đặt tên riêng cho đĩa mềm
- Display summary when finished: cho hiện ra bảng r tổng kết về tình trạng của đĩa khi đã định dạng xong
- Copy system files: chép các tập tin hệ thống vào đĩa



Chú ý: Không thể sử dụng lệnh này để định dạng đĩa cứng khi đĩa đó đang chứa hệ điều hành. (Vi bản thân Windows cũng đang được lưu trữ trong đĩa cứng).

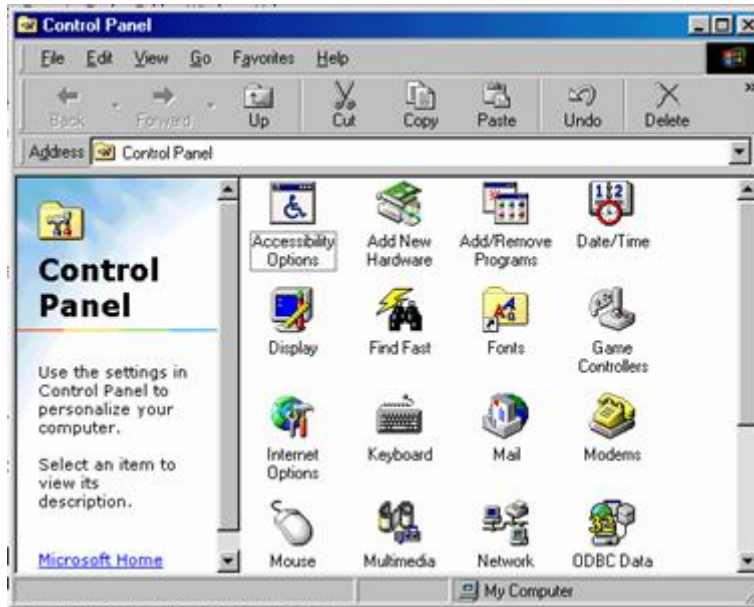
## BÀI 6: CONTROL PANEL

Chương trình dùng để điều khiển hệ thống trong Windows

Nhấp nút Start, trở vào Settings và nhấp chọn Control Panel để mở ra cửa sổ Control Panel,



trong cửa sổ này có nhiều biểu tượng tượng trưng cho một chức năng điều khiển hệ thống khác nhau cho máy PC cũng như cho Windows

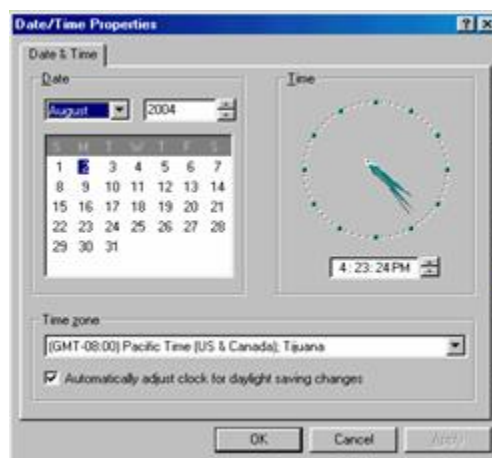


Trong phần này chúng ta chỉ xét đến vài tính năng điều khiển đặc trưng, cần thiết mà ở mỗi máy PC đều có.

### ➔ 3.1 - Date/Time

Đây là tính năng dùng để đặt ngày giờ hệ thống trong máy, nhấp đúp vào biểu tượng Date/time

- Date: Tiến hành các bước chọn ngày, tháng, năm



● Time: Để điều chỉnh đồng hồ , nhấp chuột vào từng vị trí, giờ , phút, giây và chỉnh bằng cách nhấp nút tăng, giảm



Ngoài ra còn có thể chọn múi giờ cho thích hợp

Nhấp nút OK để kết thúc

### ➔ 3.2 –Display

Tính năng điều khiển màn hình, nhấp đúp biểu tượng này sẽ mở ra một hộp thoại có nhiều lớp, trong mỗi lớp là có một chức năng điều khiển khác nhau:

#### ● Background

Chọn cách trang trí cho nền của Desktop gồm:

- Pattern: Trang trí cho nền Desktop bằng cách thể hiện mẫu hình lưới, Nếu chọn - - None thì nền chỉ có một màu xanh lá (đậm)

- Wallpaper: Một dạng ảnh Bitmap, ảnh này có thể đặt giữa Desktop, khi dùng lệnh - Center, còn nếu dùng lệnh Tile thì ảnh có khả năng tự lặp lại để che kín hết nền Desktop.

Sau khi chọn xong nhấp nút Apply để lệnh có hiệu lực và chọn sang lớp điều khiển khác.

#### ● Screen Saver

- Screen Saver: chọn mẫu hình ảnh di động ( nếu chọn None là không dùng đến chức năng này).

- Wait: Đặt khoảng thời gian để nhận biết màn hình tĩnh (đơn vị là phút)
- Password protected: Đặt mật khẩu để bảo mật cho công việc

Ngoài ra có thể nhấp vào mục Settings để đặt thêm một số chế độ cho ảnh động nếu có đặt Password Protected thì phải nhập đúng Password



#### ● Appearance:

Chọn cách thể hiện màu cho cửa sổ

- Scheme: Chọn một tên bộ màu thể hiện cho cách đặt màu cho tất cả các thành phần trong cửa sổ. Thường dùng mẫu Windows Standar

Chú ý: Các mục còn lại lệ thuộc vào Card điều khiển màn hình trong mỗi máy mà chúng ta sẽ có cách đặt khác nhau

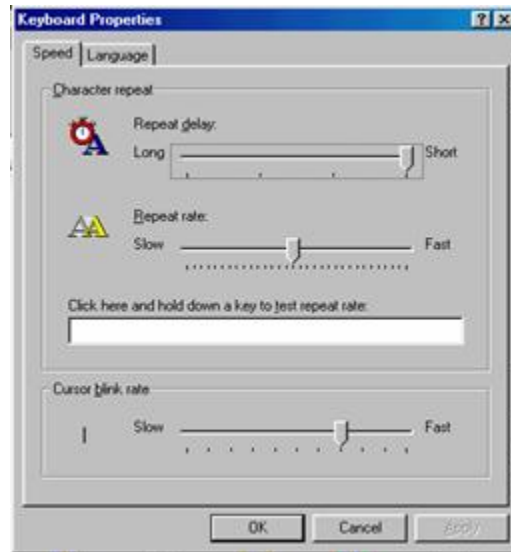
#### -Keyboard

Điều khiển bàn phím gồm các mục

- Repeat delay: Khoảng thời gian để nhận biết một ký tự được gõ lặp lại

Dùng Mouse kéo thanh trượt trong khoảng Long ... Short

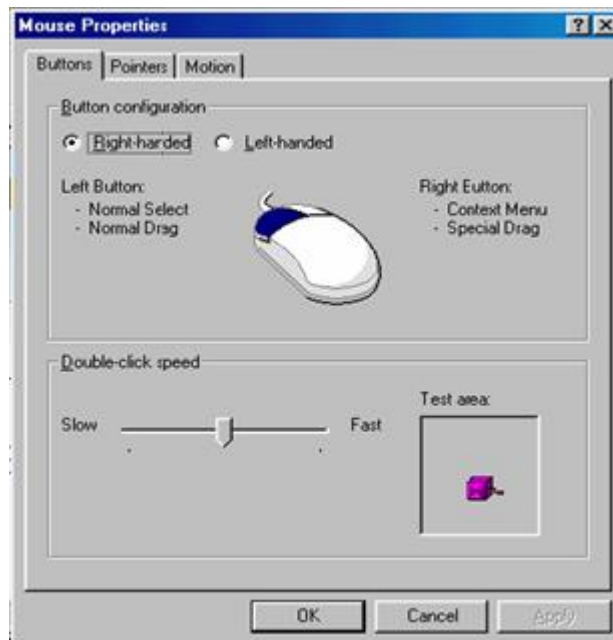
- Repeat rate: tốc độ lặp lại của phím được điều chỉnh bằng cách kéo chuột trong khoảng Slow ... fast



### 3.4 - Mouse- Chuột

- Điều khiển chuột:

- Button Configuration để thay đổi vị trí nút điều khiển của mouse.

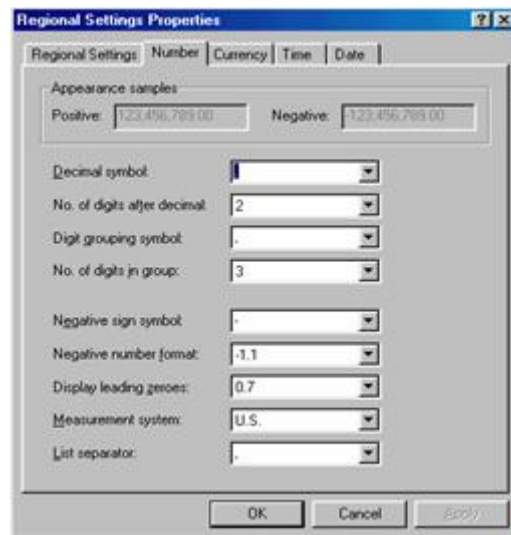


Trong lớp Motion chúng ta có thể dùng thêm lệnh:

- Pointer speed: Điều chỉnh tốc độ di chuyển con trỏ chuột trong khoảng Slow .. Fast
- Pointer trail: Nếu chọn lệnh (chèn) Showpointer trail thì con trỏ chuột để lại vết trên đường di chuyển được thực hiện trong khoảng Short ... Long

### ➡ 3.5 -Regional settings

- Regional Settings: Chọn tên nước, khi chọn cái này sẽ ảnh hưởng đến các cách chọn sau:
- Number



Thông thường máy tính được ngầm định dạng dùng hệ tiếng anh nên dấu chấm là dấu phân cách thập phân còn dấu phẩy là dấu phân cách nhóm 3 chữ số, trong khi người Việt Nam lại dùng theo hệ tiếng pháp có quy định ngược lại.

Nếu cần thiết chúng ta có thể thay đổi lại quy định này bằng các lệnh:

- Decimal symbol: Gỡ lại dấu chấm thay cho dấu phẩy
- Digit grouping symbol: gỡ dấu chấm thay cho dấu phẩy

#### ● Time

Thường dùng theo dạng hh:mm:ss (giờ:phút:giây)

#### ● Date

Hệ tiếng Anh dùng: MM/DD/YY

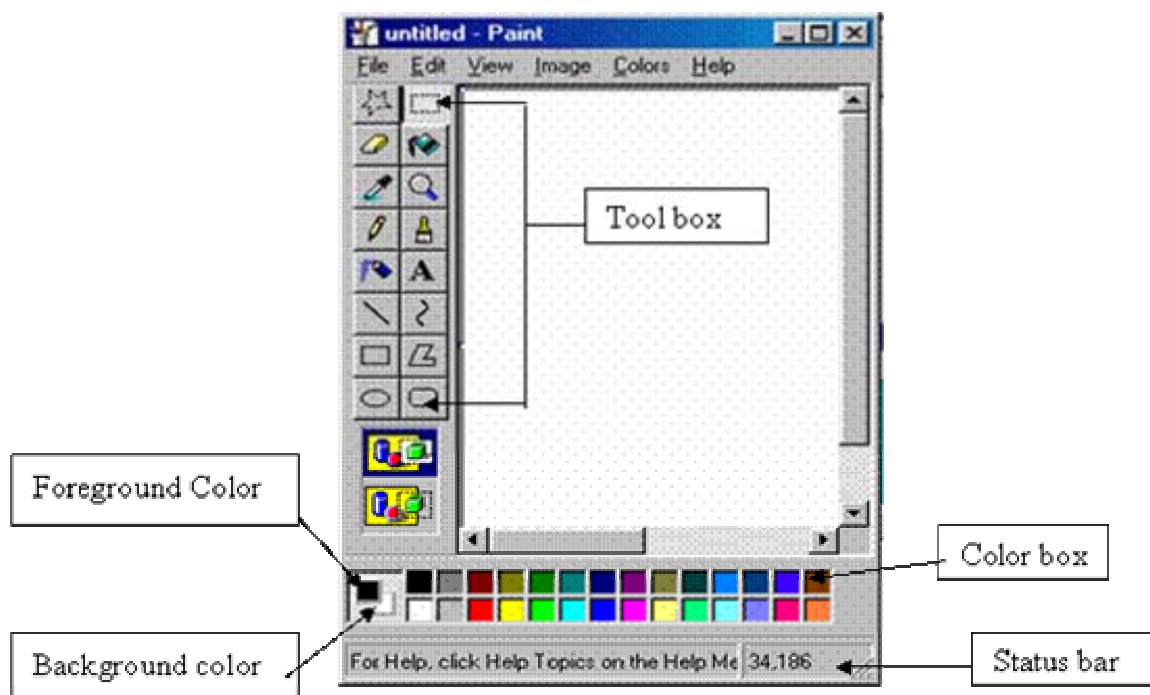
Người Việt dùng: DD/MM/YY

### Chương trình PAINT

Trong Windows 98 có một chương trình ứng dụng dùng để vẽ hình theo dạng Bitmap, đó là Paint. Để làm việc với chương trình này chúng ta phải mở cửa sổ chương trình bằng cách nhấp vào nút Start chọn Programs chọn Accessories và chọn Paint



















Cửa sổ Paint xuất hiện



● Tool box

Các công cụ để vẽ hình

| Biểu tượng                                                                          | Tên gọi          | Chức năng                          |
|-------------------------------------------------------------------------------------|------------------|------------------------------------|
|  | Free-form select | Tạo một mẫu cắt có dạng tùy ý      |
|  | Select           | Tạo một mẫu cắt hình chữ nhật      |
|  | Eraser           | Xoá hình                           |
|  | Fill with color  | Tô màu trong đường bao khép kín    |
|  | Pick Color       | Lấy mẫu màu trong một phần hình vẽ |
|  | Magnifier        | Phóng lớn một phần hình vẽ         |
|  | Pencil           | Vẽ đường tự do                     |

|                                                                                   |                 |                               |
|-----------------------------------------------------------------------------------|-----------------|-------------------------------|
|  | Brush           | Cọ vẽ tự do                   |
|  | Airbrush        | Tạo hiệu quả phun màu         |
|  | Text            | Vẽ chữ                        |
|  | Line            | Vẽ đoạn thẳng                 |
|  | Curve           | Vẽ đoạn cong hình dục         |
|  | Rectangle       | Vẽ hình chữ nhật              |
|  | Polygon         | Vẽ hình đa giác               |
|  | Ellipse         | Vẽ hình Elip                  |
|  | Round Rectangle | Vẽ hình chữ nhật tròn bốn góc |

Có thể chia các công cụ trên thành hai nhóm:

- Các công cụ dùng để vẽ hình: Pencil, Brush, Airbrush, Text, Line, Curve, Rectangle, Polygon, Ellipse, Rounded Rectangle
- Các công cụ dùng để hiệu chỉnh vẽ: Free-form select, Eraser, Select, Pick Color, Magnifer.

#### ● Color box

- Nhấp chọn màu bằng nút trái của chuột: màu Foreground
- Nhấp chọn màu bằng nút phải của chuột: Màu Background

#### ● Vẽ hình

\* Các bước để vẽ hình

- Bước 1: Chọn một công cụ vẽ thích hợp. mỗi công cụ vẽ sẽ có một hiệu quả riêng biệt

Chú ý: Hình dạng con trỏ chuột có thể thay đổi theo công cụ vẽ đang chọn

- Bước 2: Chọn độ dày của nét vẽ, dạng cọ vẽ hoặc hình chữ nhật từ nhóm biểu tượng ở dưới Tool box
- Bước 3: Chọn màu Foreground



- Bước 4: Chọn màu Background nếu muốn áp dụng để vẽ bằng công cụ Rectangle hoặc Ellipse

- Bước 5: Vẽ hình bằng cách kép chuột trong vùng vẽ

Chú ý: Nếu vẽ hình không thích hợp có thể dùng lệnh Undo bằng cách gõ tổ hợp phím Ctrl+Z

#### ● Sử dụng các công cụ vẽ

Các công cụ vẽ đều có thể sử dụng nút trái hoặc nút phải của chuột để vẽ. Vẽ bằng nút trái hình sẽ có màu của Foreground, vẽ bằng nút phải hình sẽ có màu Background

#### ● Lưu bản vẽ vào ổ đĩa cứng

Vào Menu File Chọn Save hộp thoại Save As hiện ra:



Gõ tên bản vẽ vào ô File name và chọn Save.

**TRƯỜNG ĐẠI HỌC NÔNG NGHIỆP I  
BỘ MÔN CÔNG NGHỆ PHẦN MỀM**

\*\*\*\*\*

Các Tác giả: Đỗ Thị mơ (Chủ biên)  
TS. Dương Xuân Thành, Ths. Nguyễn Thị Thủy, Ths. Ngô Tuấn Anh



**GIÁO TRÌNH  
TIN HỌC ĐẠI CƯƠNG**

**DÙNG CHO KHÓA A**



**Hà Nội - 2006**

## MỤC LỤC

|                                                              |           |
|--------------------------------------------------------------|-----------|
| <b>Phần I: Đại cương về Tin học.....</b>                     | <b>3</b>  |
| <b>Chương I: Giới thiệu chung .....</b>                      | <b>3</b>  |
| 1. Thông tin và tin học.....                                 | 3         |
| 1.1 Khái niệm thông tin.....                                 | 3         |
| 1.2 Khái niệm tin học .....                                  | 3         |
| 2. Hệ đếm trong máy vi tính .....                            | 3         |
| 2.1 Hệ 10 .....                                              | 3         |
| 2.2 Hệ 2 .....                                               | 4         |
| 2.3 Hệ 16 .....                                              | 4         |
| 2.4 Biểu diễn số trong máy tính và các đơn vị thông tin..... | 5         |
| 2.5 Chuyển đổi giữa các hệ đếm.....                          | 6         |
| 2.6 Các phép toán trong hệ 2 .....                           | 8         |
| 3. Tệp và thư mục.....                                       | 10        |
| 3.1 Tệp.....                                                 | 10        |
| 3.2 Thư mục.....                                             | 10        |
| 4. Mã hoá .....                                              | 11        |
| 4.1 Khái niệm mã hoá .....                                   | 11        |
| 4.2 Mã ASCII.....                                            | 12        |
| 5. Đại số logic .....                                        | 12        |
| 5.1 Khái niệm mệnh đề Logic, biến Logic, hàm Logic.....      | 12        |
| 5.2 Các toán tử Logic cơ bản.....                            | 13        |
| Bài tập -Câu hỏi ôn tập.....                                 | 15        |
| <b>Chương II: Cấu trúc máy vi tính .....</b>                 | <b>16</b> |
| 1. Chức năng và sơ đồ cấu trúc của máy vi tính .....         | 16        |
| 1.1 Chức năng của máy vi tính .....                          | 16        |
| 1.2 Sơ đồ cấu trúc máy vi tính.....                          | 16        |
| 2. Các bộ phận cơ bản của máy vi tính .....                  | 17        |
| 2.1 Bộ xử lý trung tâm.....                                  | 17        |
| 2.2 Bộ nhớ .....                                             | 18        |
| 2.3 Thiết bị ngoại vi .....                                  | 18        |
| 3. Tổ chức một phòng máy tính và cách bảo quản .....         | 21        |
| 3.1 Tổ chức một phòng máy .....                              | 21        |
| 3.2 Bảo quản đĩa từ, màn hình, bàn phím.....                 | 21        |
| Bài tập - Câu hỏi ôn tập.....                                | 22        |

|                                                                       |           |
|-----------------------------------------------------------------------|-----------|
| <b>Chương III: Hệ điều hành .....</b>                                 | <b>23</b> |
| 1. Khái niệm hệ điều hành.....                                        | 23        |
| 1.1 Khái niệm hệ điều hành.....                                       | 23        |
| 1.2 Phân loại hệ điều hành.....                                       | 23        |
| 1.3 Một số hệ điều hành thông dụng .....                              | 23        |
| 2. Hệ điều hành Microsoft windows 98/2000 .....                       | 25        |
| 2.1 Khởi động máy và các thành phần cơ bản.....                       | 25        |
| 2.2 Tạo, thay đổi các biểu tượng hoặc mục chọn cho một nội dung ..... | 28        |
| 2.3 Sử dụng hộp thoại CONTROL PANEL.....                              | 29        |
| 2.4 Sử dụng chương trình Windows Explorer.....                        | 35        |
| Bài tập - Câu hỏi ôn tập.....                                         | 39        |
| <b>Chương IV: Hệ soạn thảo văn bản Microsoft Word .....</b>           | <b>40</b> |
| 1. Giới thiệu chương trình MS-Word .....                              | 40        |
| 1.1 Giới thiệu chung.....                                             | 40        |
| 1.2 Cách sử dụng chuột.....                                           | 40        |
| 1.3 Khởi động và ra khỏi Word .....                                   | 41        |
| 1.4 Mô tả các thành phần trong màn hình Word .....                    | 42        |
| 2. Các phím gõ tắt (Shortcut-Key).....                                | 47        |
| 2.1 Các phím gõ tắt bắt đầu bằng phím ALT .....                       | 47        |
| 2.2 Các phím gõ tắt bắt đầu bằng phím CTRL.....                       | 47        |
| 2.3 Các phím gõ tắt bắt đầu bằng phím SHIFT .....                     | 47        |
| 3. Các thao tác định dạng .....                                       | 48        |
| 3.1 Phương pháp tiếng Việt.....                                       | 48        |
| 3.2 Định dạng kí tự.....                                              | 49        |
| 3.3 Định dạng đoạn văn bản .....                                      | 51        |
| 3.4 Định dạng trang giấy .....                                        | 51        |
| 4. Kê bảng biểu .....                                                 | 47        |
| 4.1 Chèn bảng biểu và văn bản.....                                    | 52        |
| 4.2 Một số thao tác sửa chữa bảng biểu .....                          | 54        |
| 4.3 Đánh số thứ tự trong một cột .....                                | 55        |
| 4.4 Tạo bảng băng bút chì .....                                       | 56        |
| 5. Các thao tác thường gặp .....                                      | 56        |
| 5.1 Viết chữ cái đầu tiên.....                                        | 56        |
| 5.2 Chia trang hoặc đoạn văn bản thành nhiều cột .....                | 56        |
| 5.3 Thuật ngữ viết tắt .....                                          | 57        |
| 5.4 Tạo đoạn văn bản mẫu.....                                         | 58        |
| 5.5 Viết số mũ và chỉ số .....                                        | 59        |

|                                                                   |           |
|-------------------------------------------------------------------|-----------|
| 5.6 Chèn vào văn bản một kí tự đặc biệt .....                     | 59        |
| 6. Tự động hoá quá trình soạn thảo .....                          | 59        |
| 6.1 Tạo dòng TAB. ....                                            | 59        |
| 6.2. Tạo các Style.....                                           | 60        |
| 6.3 Tạo Macro.....                                                | 62        |
| 7. Đồ hoạ .....                                                   | 65        |
| 7.1 Các thao tác vẽ cơ bản.....                                   | 65        |
| 7.2 Chữ nghệ thuật .....                                          | 68        |
| 8. Trộn tài liệu tạo thành văn bản mới .....                      | 71        |
| 8.1 Tạo mẫu giấy mời.....                                         | 71        |
| 8.2 Tạo danh sách .....                                           | 71        |
| 9. Các thao tác hỗ trợ.....                                       | 72        |
| 9.1 Khôi phục các thanh công cụ.....                              | 72        |
| 9.2 Hiện hoặc dấu các thanh công cụ.....                          | 73        |
| 9.3 Thêm bớt chức năng trên thanh công cụ.....                    | 74        |
| 9.4 Bảo mật văn bản.....                                          | 75        |
| Bài tập - Câu hỏi ôn tập.....                                     | 78        |
| <b>Chương V: Internet và cách sử dụng Internet .....</b>          | <b>83</b> |
| 1. Giới thiệu chung.....                                          | 83        |
| 1.1 Mạng máy tính .....                                           | 83        |
| 1.2 Nhà cung cấp dịch vụ Internet .....                           | 84        |
| 1.3 Kết nối Internet .....                                        | 85        |
| 2. Sử dụng trình duyệt Internet Explorer (IE).....                | 86        |
| 2.1 Khởi động Internet Explorer .....                             | 86        |
| 2.2 Ra khỏi Internet Explorer .....                               | 86        |
| 2.3 Tìm kiếm thông tin.....                                       | 86        |
| 2.4 Cách lưu trữ các địa chỉ và cách tổ chức lại các địa chỉ..... | 87        |
| 2.5 Một số tùy chọn Internet Explorer .....                       | 87        |
| 2.6 Lưu trang Web vào máy tính .....                              | 88        |
| 2.7 Lưu ảnh.....                                                  | 88        |
| 2.8 In trang Web .....                                            | 88        |
| 3. Thư điện tử (E-mail).....                                      | 89        |
| 3.1 Giới thiệu Email .....                                        | 89        |
| 3.2 Các mô hình hoạt động của dịch vụ Email .....                 | 90        |
| 3.3 Tạo hòm thư miễn phí với Yahoo .....                          | 90        |
| 3.4 Gửi và nhận thư với Yahoo.....                                | 90        |
| 3.5 Gửi nhận thư bằng Outlook Express .....                       | 92        |

|                                                                                     |            |
|-------------------------------------------------------------------------------------|------------|
| 3.6 Download và một số địa chỉ Internet.....                                        | 94         |
| Bài tập - Câu hỏi ôn tập .....                                                      | 95         |
| <b>Chương VI: Giải thuật .....</b>                                                  | <b>96</b>  |
| 1. Khái niệm giải thuật .....                                                       | 96         |
| 2. Các yêu cầu với giải thuật.....                                                  | 97         |
| 3. Các cách diễn tả giải thuật.....                                                 | 97         |
| 4. Thiết kế giải thuật.....                                                         | 99         |
| 5. Giải thuật sắp xếp.....                                                          | 100        |
| 6. Giải thuật tìm kiếm.....                                                         | 101        |
| 7. Giải thuật đệ quy .....                                                          | 102        |
| Bài tập chương VI. ....                                                             | 105        |
| <br>                                                                                |            |
| <b>Phần II: Ngôn ngữ lập trình Turbo Pascal.....</b>                                | <b>106</b> |
| <b>Chương I: Cấu trúc của chương trình Pascal.....</b>                              | <b>106</b> |
| <br>                                                                                |            |
| 1. Giới thiệu về ngôn ngữ lập trình .....                                           | 106        |
| 2. Yêu cầu phần mềm và cách khởi động.....                                          | 107        |
| 3. Những quy định về cách viết biểu thức và chương trình .....                      | 108        |
| 4. Cấu trúc chung của một chương trình Pascal .....                                 | 109        |
| Câu hỏi ôn tập chương I .....                                                       | 109        |
| <b>Chương II: Các kiểu dữ liệu cơ sở và cách khai báo .....</b>                     | <b>111</b> |
| 1. Khái niệm dữ liệu, kiểu dữ liệu .....                                            | 111        |
| 2. Các kiểu dữ liệu đơn giản.....                                                   | 111        |
| 3. Cách khai báo.....                                                               | 115        |
| 4. Biểu thức và câu lệnh.....                                                       | 117        |
| Bài tập chương II .....                                                             | 117        |
| <b>Chương III: Các thủ tục vào ra dữ liệu và các cấu trúc điều khiển .....</b>      | <b>118</b> |
| 1. Phép gán .....                                                                   | 118        |
| 2. Các thủ tục Vào-Ra dữ liệu.....                                                  | 118        |
| 3. Các câu lệnh điều kiện.....                                                      | 123        |
| 4. Các lệnh lặp .....                                                               | 126        |
| Bài tập chương III.....                                                             | 134        |
| <b>Chương IV: Kiểu dữ liệu có cấu trúc: kiểu mảng, kiểu xâu kí tự, kiểu tập hợp</b> | <b>135</b> |
| 1. Kiểu mảng.....                                                                   | 135        |
| 2. Kiểu xâu kí tự.....                                                              | 142        |

|                                                                             |            |
|-----------------------------------------------------------------------------|------------|
| 3. Kiểu tệp.....                                                            | 146        |
| Bài tập chương IV. ....                                                     | 150        |
| <b>Chương V: Chương trình con: Hàm và thủ tục</b> .....                     | <b>151</b> |
| 1. Cấu trúc của hàm và thủ tục.....                                         | 151        |
| 2. Biến toàn cục, biến cục bộ và truyền dữ liệu.....                        | 154        |
| 3. Tính đệ quy của chương trình con .....                                   | 154        |
| 4. Một số chương trình con của Turbo Pascal .....                           | 156        |
| Bài tập chương V. ....                                                      |            |
| <b>Chương VI: Kiểu dữ liệu có cấu trúc: Kiểu bản ghi và kiểu tệp.</b> ..... | <b>157</b> |
| 1. Kiểu bản ghi .....                                                       | 157        |
| 2. Kiểu tệp.....                                                            | 159        |
| Bài tập chương VI. ....                                                     | 171        |
| <b>Chương VII: Đồ họa và âm thanh</b> .....                                 | <b>172</b> |
| 1. Đồ họa .....                                                             | 172        |
| 2. Âm thanh .....                                                           | 182        |
| Bài tập chương VII. ....                                                    | 185        |
| <b>Tài liệu tham khảo.</b> .....                                            | <b>186</b> |



## MỞ ĐẦU

Tin học là một ngành khoa học được dạy trong tất cả các trường đại học, tuy nhiên nội dung dạy có khác nhau. Tùy theo yêu cầu của từng ngành mà chỉ định số tiết, ngành nông nghiệp được chia ra làm hai khối, khối A và khối B, thường dạy 60 tiết cho khối A và 45 tiết cho khối B.

Giáo trình Tin học đại cương này được viết cho sinh viên Đại học Nông nghiệp thuộc các chuyên ngành khối A. Nội dung viết theo đề cương môn học trong khung chương trình đào tạo kỹ sư Nông nghiệp của Bộ Giáo dục và Đào tạo năm 2004.

Giáo trình này cung cấp cho sinh viên những kiến thức cơ bản về Tin học, kiến thức lập trình trong ngôn ngữ Pascal và một số phần mềm được ứng dụng rộng rãi nhất hiện nay, từ đó giúp sinh viên nâng cao hiểu biết về tin học và vận dụng kiến thức Tin học vào các môn học chuyên ngành, cũng như vận dụng Tin học vào các công việc của mình.

Giáo trình này viết cho người học, do đó khi dạy các giáo viên cần tham khảo thêm sách viết về cấu trúc máy tính, hệ điều hành, hệ soạn thảo văn bản Microsoft Word, mạng máy tính, Internet, cấu trúc dữ liệu và thuật giải, lập trình trong Turbo Pascal.

Để nắm vững kiến thức trình bày trong sách không có cách nào tốt hơn là xem kỹ các thí dụ, làm đủ các bài tập, trả lời các câu hỏi ôn tập và tăng cường thực hành trên máy tính, ngoài ra còn phải có trình độ tiếng Anh tối thiểu là trình độ B.

Nội dung của giáo trình chia 2 phần; Phần I bao gồm 6 chương nêu lên những kiến thức đại cương về Tin học. Phần 2 gồm 7 chương nêu lên cách lập trình trong Turbo Pascal. Sau mỗi chương có phần bài tập hoặc câu hỏi ôn tập. Các bài tập đã được chọn lọc ở mức độ trung bình để sinh viên qua đó hiểu thêm bài giảng và thu hoạch thêm một số nội dung mở rộng gắn với thực tế.

Chương I - Phần 1 trình bày các khái niệm chung trong Tin học. Chương này giới thiệu các khái niệm cơ bản cần thiết nhất trong Tin học như: Thông tin và Tin học, hệ đếm sử dụng trong máy tính, mã hoá, đại số Logic, tệp và thư mục. Các khái niệm được trình bày một cách cơ bản và ngắn gọn nhất. Chỉ nêu những phần cần thiết trong Tin học để phù hợp với thời lượng của môn học. Cuối chương là các bài tập và câu hỏi ôn tập để củng cố kiến thức.

Chương II - Phần 1 trình bày cấu trúc của máy vi tính. Chương này giới thiệu các thành phần cơ bản của máy vi tính đó là: Bộ xử lý trung tâm, bộ nhớ và các thiết bị ngoại vi. Ngoài ra còn giới thiệu cách tổ chức và bảo quản một phòng máy tính. Chương này chỉ giới thiệu các khối chính của máy vi tính bao gồm cấu trúc và chức năng, tránh đi vào cấu tạo chi tiết làm cho sinh viên khó hiểu vì sinh viên không thuộc chuyên ngành Tin học nên thiếu các kiến thức về kỹ thuật điện tử và kỹ thuật số.

Chương III - Phần 1 giới thiệu hệ điều hành. Trước tiên là khái niệm hệ điều hành, phân loại hệ điều hành, trình bày khái quát các hệ điều hành thông dụng hiện nay. Phần chính của chương này trình bày cụ thể về hệ điều hành WINDOWS 98/2000 trong đó có chương trình WINDOWS EXPLORER. Trong chương này nên chú trọng phần thực hành.

Chương IV - Phần 1 trình bày về hệ soạn thảo văn bản Microsoft Word 2000. Trong đó quan trọng nhất là các lệnh về soạn thảo văn bản, trình bày văn bản, kẻ bảng biểu, các lệnh trên tệp, các lệnh vẽ. Chú trọng thực hành nhiều trên máy để củng cố các kiến thức và tăng cường các kỹ năng soạn thảo.

Chương V - Phần 1 trình bày về Internet. Chương này cung cấp cho người học kiến thức về Internet, cách sử dụng Internet, cách sử dụng thư điện tử và các dịch vụ khác trên Internet.

Chương VI - Phần 1 trình bày về giải thuật. Chương này cung cấp cho người học kiến thức về giải thuật, các giải thuật cơ bản về sắp xếp và tìm kiếm.

Chương I - Phần 2 trình bày cấu trúc của chương trình trong Turbo Pascal. Chương này nêu lên cấu trúc của chương trình và các bước cơ bản khi lập trình.

Chương II - Phần 2 trình bày các kiểu dữ liệu cơ sở. Chương này cung cấp các kiến thức về các kiểu dữ liệu và đi sâu vào các kiểu dữ liệu đơn giản.

Chương III - Phần 2 nêu ra những kiến thức tổ chức các thủ tục vào, ra dữ liệu và các cấu trúc điều khiển trong Pascal.

Chương IV - Phần 2 trình bày các kiểu dữ liệu có cấu trúc, bao gồm kiểu mảng, kiểu xâu kí tự, kiểu tập hợp.

Chương V - Phần 2 nêu ra cách tổ chức chương trình con là hàm và thủ tục, cách sử dụng biến toàn cục và biến cục bộ, vấn đề truyền dữ liệu trong và tính đệ quy trong chương trình con.

Chương VI - Phần 2 trình bày các kiểu dữ liệu có cấu trúc bao gồm kiểu bản ghi và kiểu tệp.

Chương VII - Phần 2 trình bày các kiến thức về đồ họa và âm thanh trong Pascal.

Khi biên soạn chúng tôi đã cố gắng đưa ra một bố cục của cuốn sách sao cho bám sát được đề cương chi tiết của môn học, với hy vọng rằng giáo trình này sẽ rất bổ ích và cần thiết đối với các sinh viên đại học Nông nghiệp và các ngành khác.

Giáo trình này được viết bởi các thầy cô trong khoa Công nghệ thông tin. Thầy Dương Xuân Thành viết chương IV phần 1 và chương VII phần 2. Cô Nguyễn Thị Thủy viết chương I, chương II và chương III phần 2. Thầy Nguyễn Tuấn Anh viết chương VI phần 1. Cô Đỗ Thị Mơ viết chương I, chương II, chương III, chương V phần 1 và chương IV, chương V, chương VI phần 2.

Trong quá trình biên soạn sách chúng tôi đã nhận được nhiều ý kiến đóng góp quý báu của các giáo viên trong bộ môn Công nghệ phần mềm và bộ môn khoa học máy tính - Khoa công nghệ thông tin trường đại học Nông nghiệp I Hà nội. Chúng tôi xin chân thành cảm ơn.

Tuy đã hết sức cố gắng nhưng chúng tôi nghĩ rằng cuốn sách này còn nhiều thiếu sót, rất mong sự góp ý của bạn đọc. Xin chân thành cảm ơn.

*Hà nội, ngày 20 tháng 2 năm 2006*

**Các tác giả**

# PHẦN I: ĐẠI CƯƠNG VỀ TIN HỌC

## CHƯƠNG I: GIỚI THIỆU CHUNG

Để tìm hiểu sâu hơn về các phần trong giáo trình này trước tiên phải có những kiến thức cơ bản ban đầu. Chương này cung cấp những khái niệm, những kiến thức cơ bản nhất trong Tin học. Nội dung chính của chương này bao gồm các phần: Khái niệm Thông tin và Tin học, hệ đếm được sử dụng trong máy tính, mã hoá, đại số logic, tệp và thư mục.

### 1 – Thông tin và Tin học

#### 1.1 - Khái niệm Thông tin

\* Thông tin là tập hợp các dấu hiệu, các đặc điểm, các tính chất cho ta hiểu biết về một đối tượng.

\* Thông tin có thể được biểu diễn dưới nhiều dạng phong phú như biểu tượng, kí tự, hình ảnh, âm thanh v.v... . Thí dụ biển đăng kí xe ô tô, xe máy bao gồm các kí tự dạng chữ và số, cho ta biết các thông tin về xe như xe đó thuộc tỉnh nào, xe công hay xe tư, .... Trên các bu rui kiện có hình cái cốc và hình cái ô để báo cho biết trong thùng có vật dễ vỡ và phải che để không bị thấm nước.

\* Nói tới thông tin phải đề cập tới quá trình xử lý thông tin. Ngày nay do xã hội phát triển nên lượng thông tin cần xử lý rất lớn, độ phức tạp của quá trình xử lý rất cao, do vậy xử lý thông tin bằng thủ công, cơ giới không thể đáp ứng được yêu cầu tự động hoá quá trình xử lý thông tin. Tin học ra đời để đáp ứng yêu cầu đó

#### 1.2 – Khái niệm Tin học

Từ "Tin học" (Informatique) được người Pháp tên là Phillipe Dreyfus dùng đầu tiên vào năm 1962 để định nghĩa cho một môn khoa học mới mẻ trong lĩnh vực xử lí thông tin. Sau đó vào năm 1966 viện hàn lâm khoa học Pháp đã đưa ra định nghĩa sau: "Tin học là môn khoa học về xử lí hợp lí các thông tin, đặc biệt bằng các thiết bị tự động, các thông tin đó chứa đựng kiến thức của loài người trong các lĩnh vực kĩ thuật, kinh tế và xã hội "

*Vậy ta có thể coi môn học Tin học là một môn học nghiên cứu việc tự động hoá quá trình xử lý thông tin.*

Định nghĩa trên cho phép ta phân Tin học thành hai lĩnh vực sau:

- Phần mềm (Soft Ware): Xây dựng các thuật toán, các chương trình máy tính để xử lý thông tin.

- Phần cứng (Hard Ware): Thiết kế, lắp đặt, bảo trì các thiết bị tự động để xử lý thông tin.

Ngày nay, các kĩ thuật viên tin học vừa có khả năng hoạt động trên lĩnh vực phần cứng cũng như phần mềm.

### 2- Hệ đếm trong máy vi tính

#### 2.1- Hệ 10 ( Hệ thập phân: Decimal)

Hệ 10 hay hệ thập phân là hệ đếm được sử dụng để đếm và tính toán trong đời sống hàng ngày.

Hệ 10 sử dụng 10 kí hiệu chữ số 0,1,2,3,4,5,6,7,8,9 để biểu diễn các số. Do sử dụng bộ 10 chữ số nên hệ đếm này có cơ số là 10.

Để phân biệt số trong các hệ đếm khác nhau người ta thường viết số trong hệ 10 kèm theo cơ số dạng sau: Nb ( số N trong hệ đếm cơ số b) hoặc viết chữ D vào sau số

Ví dụ:  $2092_{10}$ ;  $789,12_{10}$  ;  $1027D$ ;  $125,47D$

Ta có thể biểu diễn số theo cơ số của hệ đếm. Ví dụ biểu diễn số N trong hệ 10 theo cơ số của nó như sau:

$$8623,56_{10} = 8 \times 10^3 + 6 \times 10^2 + 2 \times 10^1 + 3 \times 10^0 + 5 \times 10^{-1} + 6 \times 10^{-2}$$

Tổng quát biểu diễn một số trong hệ đếm cơ số b qua cơ số của hệ đếm như sau:

Giả sử có số  $N_b$  có n+1 chữ số phần nguyên là  $a_n a_{n-1} \dots a_1 a_0$  và m chữ số phần lẻ là  $c_1 c_2 \dots c_m$ . Số này được viết tổng quát như sau  $N_b = a_n a_{n-1} \dots a_1 a_0, c_1 c_2 \dots c_m$

Số  $N_b$  được biểu diễn theo cơ số b như sau:

$$N_b = a_n a_{n-1} \dots a_1 a_0, c_1 c_2 \dots c_m = a_n \times b^n + a_{n-1} \times b^{n-1} + \dots + a_1 \times b^1 + a_0 \times b^0 + c_1 \times b^{-1} + c_2 \times b^{-2} + \dots + c_m \times b^{-m} \quad (*)$$

Công thức (\*) nêu trên sẽ được sử dụng để chuyển đổi số giữa các hệ đếm.

## 2.2- Hệ 2 (Hệ nhị phân: Binary)

Hệ 2 hay hệ nhị phân là hệ đếm sử dụng 2 chữ số 0, 1 để biểu diễn các số.

Bit là đơn vị cơ bản của thông tin theo hệ thống số nhị phân ( Binary digit). Các mạch điện tử trong máy tính sẽ phát hiện sự khác nhau giữa hai trạng thái (dòng điện mức cao và dòng điện mức thấp) và biểu diễn các trạng thái đó dưới dạng một trong hai số nhị phân 1 hoặc 0. Vì việc chế tạo một mạch điện tin cậy có thể phân biệt được sự khác nhau giữa 1 và 0 là tương đối dễ dàng và rẻ tiền, cho nên máy tính có khả năng xử lý nội bộ các thông tin nhị phân một cách rất chính xác, theo tiêu chuẩn, nó mắc ít hơn một lỗi nội bộ trong 100 tỉ thao tác xử lý

Do chỉ sử dụng tập 2 chữ số nên hệ 2 có cơ số là 2. Số N trong hệ 2 được kí hiệu  $N_2$  hoặc viết chữ B vào sau số

Ví dụ:  $10011_2$  ;  $11010111_2$  ;  $1110011B$  ;  $10100001B$

Biểu diễn tương đương hệ 10 và hệ 2 như sau:

|       |   |    |    |     |     |     |     |      |      |      |      |      |      |      |      |     |
|-------|---|----|----|-----|-----|-----|-----|------|------|------|------|------|------|------|------|-----|
| Hệ 10 | 1 | 2  | 3  | 4   | 5   | 6   | 7   | 8    | 9    | 10   | 11   | 12   | 13   | 14   | 15   | ... |
| Hệ 2  | 1 | 10 | 11 | 100 | 101 | 110 | 111 | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 | ... |

Trong máy tính sử dụng hệ 2. Như vậy mọi thông tin đưa vào máy tính đều được mã hoá và chuyển sang hệ 2. Hệ 2 trong máy tính được sử dụng do lý do kỹ thuật: chế tạo các linh kiện có 2 trạng thái (ứng với số 0 và số 1) thì đơn giản và có tính ổn định cao. Nếu sử dụng hệ 10 thì các linh kiện phải có 10 trạng thái tương ứng với 10 chữ số như vậy sẽ rất khó khăn và phức tạp.

## 2.3- Hệ 16 (Hexadecimal)

Hệ 16 sử dụng 16 kí hiệu 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F để biểu diễn các số. Do sử dụng 16 kí hiệu nên hệ này có cơ số là 16. Mỗi chữ số của hệ 16 tương ứng với 1 nhóm 4 bit trong hệ 2. Số N trong hệ 16 được kí hiệu  $N_{16}$  hoặc viết chữ H vào sau số

Ví dụ:  $10EF_{16}$  ;  $AE9F_{16}$  ;  $2EFBH$  ;  $45CDH$

Biểu diễn tương đương hệ 10, hệ 16 và hệ 2 như sau:

|       |      |      |      |      |      |      |      |      |      |      |      |      |
|-------|------|------|------|------|------|------|------|------|------|------|------|------|
| Hệ 10 | 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9    | 10   | 11   | 12   |
| Hệ 16 | 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9    | A    | B    | C    |
| Hệ 2  | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 | 1000 | 1001 | 1010 | 1011 | 1100 |

|       |      |      |      |           |          |     |
|-------|------|------|------|-----------|----------|-----|
| Hệ 10 | 13   | 14   | 15   | 16        | 17       | ... |
| Hệ 16 | D    | E    | F    | 10        | 11       | ... |
| Hệ 2  | 1101 | 1110 | 1111 | 0001 0000 | 00010001 | ... |

Hệ 16 được dùng để ghi địa chỉ các ô nhớ trong bộ nhớ của máy tính, địa chỉ các cổng vào / ra của máy tính.

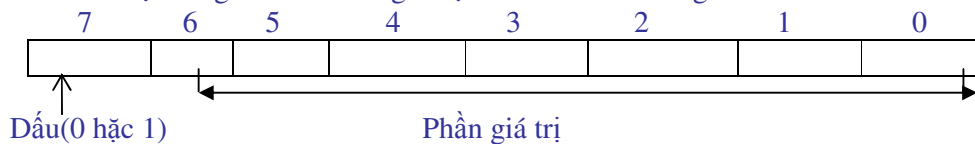
## 2.4- Biểu diễn số trong máy tính và các đơn vị thông tin

### a) Biểu diễn số trong máy tính

\* Trong máy tính các số được biểu diễn theo một khuôn thống nhất, độ dài khuôn có thể 8 bit hoặc 16 bit hoặc 32 bit, ...

Có thể biểu diễn số theo dạng dấu phẩy tĩnh hoặc dạng dấu phẩy động. Trong phần này ta chỉ xét cách biểu diễn số nguyên theo dạng dấu phẩy tĩnh.

\* Biểu diễn số nguyên dấu phẩy tĩnh: Số được biểu diễn theo khuôn thống nhất, bit trái nhất (bit cao nhất) dùng để biểu diễn dấu: dấu dương (+) ứng với 0, dấu âm (-) ứng với 1. Các bit còn lại dùng để biểu diễn giá trị của số. Giả sử dùng khuôn 8 bit như sau:



Ví dụ: Biểu diễn số  $+6_{10}$  trong khuôn 8 bit. Trước tiên phải đổi  $6_{10}$  sang hệ 2 là  $110_2$ , kết quả như sau:

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
|---|---|---|---|---|---|---|---|

\* Biểu diễn số nguyên âm dấu phẩy tĩnh theo phương pháp bù 2:

Quy tắc biểu diễn số nguyên âm theo phương pháp bù 2: Trước tiên biểu diễn số nguyên dương có giá trị tuyệt đối bằng với số đó trong khuôn qui ước, sau đó đảo từng bit ( 1 đổi thành 0, 0 đổi thành 1), cuối cùng cộng với +1 (sẽ giải thích sau ở phần phép cộng trong hệ 2).

Ví dụ: biểu diễn số  $-6$  trong khuôn 8 bit như sau:

- Biểu diễn  $+6$  :

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
|---|---|---|---|---|---|---|---|

- Đảo từng bit:

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
|---|---|---|---|---|---|---|---|

- Cộng +1

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
|---|---|---|---|---|---|---|---|

- Kết quả  $-6$  :

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |
|---|---|---|---|---|---|---|---|

### b) Đơn vị thông tin

\* BIT là đơn vị nhỏ nhất của thông tin, nó biểu thị một phần tử nhớ của máy tính. Các thiết bị máy tính đều xây dựng bằng các linh kiện điện tử chỉ có hai trạng thái khác nhau và được mã hoá bằng hai số 0/1 (gọi là con số nhị phân), nếu qui ước gọi trạng thái này là 1 thì trạng thái kia là 0 (thí dụ công tắc điện đóng=1/ngắt=0 ; bóng đèn sáng=1/tắt=0).

Các thông tin đưa vào máy phải được chuyển hoá thành các xung điện có mức điện thế cao (gọi là mức Logic1) hoặc mức điện thế thấp (mức Logic 0). Mỗi phần tử nhớ của máy tính chỉ có thể được thiết lập ở trạng thái 1 hoặc 0 nên mỗi phần tử đó gọi là một BIT (Binary digiT=chữ số nhị phân).

\* BYTE (đọc là Bai) là một nhóm 8 BIT. Máy tính dùng 8 phần tử nhớ để ghi nhớ một kí tự. Vậy mỗi kí tự ứng với một BYTE (8 bit). Bít thấp nhất có số thứ tự là 0

Ví dụ: Kí tự “A” có mã ASCII  $65_{10}$  tương ứng với  $1000001_2$ , được biểu diễn trong 1 byte như sau

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
|---|---|---|---|---|---|---|---|

\* WORD (Từ) là một nhóm gồm 2 hoặc 4 Byte (với các máy tính dùng mạch vi xử lí 80286 trở về trước mỗi từ đơn gồm 2 Byte, phù hợp với loại thanh ghi 16 Bit ; với các máy tính dùng mạch vi xử lí từ 80386 về sau mỗi từ kép gồm 4 Byte phù hợp với thanh ghi 32 Bit).

\* Cũng như các đơn vị đo lường khác, đơn vị đo lường thông tin cũng có các đơn vị bội như sau:

- 1 Bít = 0 hoặc 1
- 1 Byte= 8 Bít
- 1 Kilobyte (KB) =  $2^{10}$  Bytes= 1024 Bytes
- 1 MegaByte (MB) = 1024 KB
- 1 GigaByte (GB) = 1024 MB
- 1 TeraByte (TB) = 1024 GB

## 2.5- Chuyển đổi số giữa các hệ đếm

### a) Chuyển đổi số từ hệ đếm cơ số b sang hệ 10

\* Qui tắc: Muốn chuyển đổi số từ hệ đếm cơ số b sang hệ 10 ta dùng công thức (\*) trong mục 2.1 của chương này như sau:

$$N_b = a_n a_{n-1} \dots a_1 a_0, c_1 c_2 \dots c_m$$

$$N_b = a_n \times b^n + a_{n-1} \times b^{n-1} + \dots + a_1 \times b^1 + a_0 \times b^0 + c_1 \times b^{-1} + c_2 \times b^{-2} + \dots + c_m \times b^{-m}$$

Ví dụ 1 : Đổi số  $110101_2$  sang hệ 10, ở đây b=2

$$N_{10} = 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2 + 1 \times 2^0 = 32 + 16 + 4 + 1 = 53_{10}$$

Ví dụ 2: Đổi số  $110101,11_2$  sang hệ 10

$$N_{10} = 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} = 52 + 0,5 + 0,25 = 52,75_{10}$$

Ví dụ 3: : Đổi số  $10F_{16}$  sang hệ 10, ở đây b=16

$$N_{10} = 1 \times 16^2 + 0 \times 16 + 15 \times 16^0 = 256 + 15 = 271_{10}$$

### b) Chuyển đổi số từ hệ đếm 10 sang hệ đếm cơ số b

\* Chuyển đổi số nguyên dương từ hệ 10 sang hệ đếm cơ số b:

- Qui tắc: Lấy số hệ 10 chia nguyên liên tiếp cho cơ số b, kết quả số trong hệ đếm cơ số b là các số dư của phép chia lấy theo thứ tự ngược lại ( số dư của phép chia cuối cùng là số lớn nhất).

Chú ý: Chia nguyên liên tiếp cho cơ số b có nghĩa là thương của phép chia trước lại được lấy chia nguyên tiếp cho b. . . và chỉ dừng lại khi thương = 0.

Ví dụ 1: Đổi số  $30_{10}$  sang hệ 2, ta nên tạo ra bảng chia nguyên liên tiếp để theo dõi số dư cho dễ dàng như sau

| Số bị chia | Số chia (b) | Thương | Số dư    |
|------------|-------------|--------|----------|
| 30         | 2           | 15     | <b>0</b> |
| 15         | 2           | 7      | <b>1</b> |
| 7          | 2           | 3      | <b>1</b> |
| 3          | 2           | 1      | <b>1</b> |
| 1          | 2           | 0      | <b>1</b> |

Kết quả  $N_2 = 11110_2$

Ví dụ 2: Đổi số  $172_{10}$  sang hệ 16, ở đây  $b=16$

| Số bị chia | Số chia (b) | Thương | Số dư     |
|------------|-------------|--------|-----------|
| 172        | 16          | 10     | <b>12</b> |
| 10         | 16          | 0      | <b>10</b> |

Kết quả  $N_{16} = AC_{16}$  ( 10 ứng với A, 12 ứng với C )

\* Chuyển đổi phần lẻ từ hệ 10 sang hệ đếm cơ số b :

- Quy tắc: Lấy phần lẻ nhân liên tiếp với cơ số b, kết quả lấy phần nguyên của phép nhân theo đúng thứ tự thực hiện. Nếu phần lẻ bằng 0 thì dừng; nếu phần lẻ khác 0, muốn lấy bao nhiêu chữ số ta thực hiện bấy nhiêu phép nhân.

Ví dụ 1: Đổi số  $6,25_{10}$  sang hệ 2, ở đây  $b=2$

$6_{10}$  đổi sang hệ 2 là  $110_2$

$0,25_{10}$  đổi sang hệ 2 bằng cách thực hiện phép nhân liên tiếp như sau:

$$0,25 \times 2 = 0,50$$

$$0,50 \times 2 = 1,0$$

Như vậy  $0,25_{10}$  thì hệ 2 sẽ là  $0,01_2$

Kết quả  $6,25_{10}$  đổi sang hệ 2 là:  $110,01_2$

Ví dụ 2: Đổi số  $7,37_{10}$  sang hệ 2, ở đây  $b=2$

$7_{10}$  đổi sang hệ 2 là  $111_2$

$0,37_{10}$  đổi sang hệ 2 như sau:

$$0,37 \times 2 = 0,74$$

$$0,74 \times 2 = 1,48$$

$$0,48 \times 2 = 0,96$$

$$0,96 \times 2 = 1,92$$

Ta dừng ở 4 phép nhân với 4 chữ số phần lẻ, như vậy  $0,37_{10}$  sang hệ 2 là  $0,0101_2$

Kết quả  $7,37_{10}$  đổi sang hệ 2 là  $111,0101_2$

### c) Chuyển đổi số từ hệ 16 sang hệ 2 và từ hệ 2 sang hệ 16

\* Chuyển đổi số từ hệ 16 sang hệ 2:

- Quy tắc: Thay 1 chữ số hệ 16 bằng một nhóm 4 bit hệ 2 tương ứng.

Ví dụ: Đổi số  $105EF_{16}$  hệ 2

$N_2 = 0001\ 0000\ 0101\ 1110\ 1111_2$

\* Chuyển đổi số từ hệ 2 sang hệ 16:

- Quy tắc: Thay một nhóm 4 bit hệ 2 bằng 1 chữ số hệ 16 tương ứng, việc nhóm các bit hệ 2 được thực hiện từ phải qua trái, nhóm cuối cùng không đủ 4 bit thì sẽ thêm các bit 0 vào trước. Vì hệ 16 dùng để ghi địa chỉ nên chỉ dùng các số nguyên không có phần lẻ.

Ví dụ: Đổi số  $1110010101110_2$  sang hệ 16.

Ta thêm các số 0 vào trước cho đủ các nhóm 4 bit như sau:  $N_2 = 0001\ 1100\ 1010\ 1110$  tương ứng với  $N_{16} = 1CAE_{16}$

## 2.6- Các phép toán trong hệ 2

### a) Phép cộng

\* Quy tắc cộng 2 bit như sau: Thực hiện theo bảng cộng ( còn gọi là bảng chân lý) sau đây, trong Carry là bit nhớ chuyển sang bit cao hơn.

| A | B | A+B | Carry |
|---|---|-----|-------|
| 0 | 0 | 0   | 0     |
| 0 | 1 | 1   | 0     |
| 1 | 0 | 1   | 0     |
| 1 | 1 | 0   | 1     |

- Cách thức thực hiện phép cộng hai số hệ 2 có nhiều bit như sau: đầu tiên cộng từng cặp bit có cùng thứ tự của hai số với nhau, sau đó cộng bit kết quả vừa thực hiện với bit nhớ chuyển sang từ bit thấp hơn.

Ví dụ :           A = 0 0 1 0 1 1 0  
                       B = 0 1 0 1 1 0 1  
                       A+B = 1 0 0 0 0 1 1

### b) Phép trừ

\* Quy tắc trừ 2 bit như sau: Thực hiện theo bảng trừ sau đây, trong đó Carry là bit nhớ chuyển sang bit cao hơn của số trừ.

| A | B | A-B | Carry |
|---|---|-----|-------|
| 0 | 0 | 0   | 0     |
| 1 | 0 | 1   | 0     |
| 1 | 1 | 0   | 0     |
| 0 | 1 | 1   | 1     |

- Cách thức thực hiện phép trừ hai số hệ 2 có nhiều bit ta thực hiện trừ từng cặp bit theo bảng trừ trên, làm tương tự như trong hệ 10.

Ví dụ:   A=1101011  
           B=1001101  
           A-B=0011110

Cách khác thực hiện phép trừ: Lấy số bị trừ cộng với số trừ biểu diễn ở dạng số âm.  
 $a - b = a + (-b)$



Ví dụ: Thực hiện phép tính  $15_{10} - 6_{10}$  trong khuôn 8 bit.  
 Trong ví dụ này phải biểu diễn số  $-6_{10}$  trong khuôn 8 bit  
 (lấy kết quả ở ví dụ mục 2.4 a)

$15_{10}$  biểu diễn trong khuôn 8 bit:

$-6_{10}$  biểu diễn trong khuôn 8 bit :

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |
|---|---|---|---|---|---|---|---|

Kết quả cộng 2 byte trên ( $15_{10} - 6_{10}$ ) :

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
|---|---|---|---|---|---|---|---|

Khi cộng hệ 2 ta theo quy tắc ở trên, chú ý bit trái nhất là bit dấu nên khi cộng không nhớ sang bit cao hơn.

### c) Phép nhân

\* Quy tắc nhân 2 bit: Thực hiện theo bảng nhân sau đây

| A | B | A×B |
|---|---|-----|
| 0 | 0 | 0   |
| 1 | 0 | 0   |
| 0 | 1 | 0   |
| 1 | 1 | 1   |

- Cách thức thực hiện phép nhân hai số hệ 2 có nhiều bit ta thực hiện tương tự như trong hệ 10 với bảng nhân trên, phép nhân được thực hiện kết hợp giữa phép dịch trái và phép cộng.

Ví dụ:  $a = 10011_2$   $b = 1011_2$ , thực hiện  $a \times b$  như sau:

$$\begin{array}{r}
 10011 \\
 \times 1011 \\
 \hline
 10011 \\
 + 10011 \\
 00000 \\
 10011 \\
 \hline
 11010001
 \end{array}$$

Kết quả:  $11010001$

### d) Phép chia

Trong phần này chỉ xét phép chia nguyên.

\* Cách thực hiện: Phép chia được thực hiện tương tự như trong hệ 10.

Ví dụ:  $A = 1100$   $B = 100$   $A : B = 11$

Cách khác thực hiện phép chia: Lấy số bị chia trừ liên tiếp cho số chia, nếu hiệu là số dương hoặc là 0 thì thương được cộng với 1. Dừng thực hiện khi hiệu là số âm hoặc là 0.

Ví dụ:  $a=12_{10}$   $b=6_{10}$  thực hiện  $a : b$  trong hệ 2 với khuôn 8 bit  
 $a=12_{10}$  biểu diễn trong khuôn 8 bit như sau: 00001100  
 $b=6_{10}$  vậy  $-6_{10}$  biểu diễn như sau: 11111010  
Thực hiện phép trừ liên tiếp và cho thương như sau:  
Bước 1:  $12=$  00001100                      thương=0  
           $-6=$  11111010  
Bước 2: Hiệu 00000110                      thương= 0+1=1  
           $-6=$  11111010  
          Hiệu 00000000                      thương=1 +1=10                      Dừng thực hiện  
Vậy  $00001100 : 00000110 = 10$

### 3- Tập và thư mục

#### 3.1- Tập ( File)

\* Định nghĩa tập: Tập là tập hợp thông tin có liên quan với nhau và được lưu trữ trong bộ nhớ ngoài, bộ nhớ ngoài thông dụng hiện nay là đĩa từ. Các thông tin ta đưa vào trong máy tính như chương trình, văn bản, dữ liệu,... đều được lưu trữ trên đĩa từ dưới dạng tập.

Ví dụ: Một bài thơ, một bảng điểm, một chương trình được lưu trữ trên đĩa từ dưới dạng tập.

Mỗi tập được đặt một tên duy nhất, hai tập trong một mục không được trùng tên.

\* Tên tập: Tên tập gồm hai phần là tên chính và đuôi cách nhau bằng dấu chấm (.)

- Tên chính là một nhóm ký tự bất kỳ thường không chứa dấu chấm (.), tên chính dùng để phân biệt tập, tên tập thường được chọn phù hợp với nội dung tập. Bắt buộc phải có tên chính trong tên tập.

- Phần đuôi tập ( phần mở rộng tập: Extension) thường chứa 3 ký tự không chứa dấu chấm(.). Đuôi tập được dùng để phân loại tập. Tập không nhất thiết phải có đuôi.

Các tập có đuôi EXE , COM , BAT , PRG ... là các tập kiểu chương trình.

DAT , DBF ... là các tập dữ liệu

TXT , DOC là các tập kiểu văn bản.

PIC , GIF ... là các tập đồ họa và ảnh .

Ví dụ tên tập : DIEM.DBF ; BAOCALO.DOC ; TINHLUONG.PRG

- Trong tên tập cho phép dùng ký hiệu gộp (ký hiệu thay thế) : dấu? thay thế cho một ký tự bất kỳ và dấu \* thay thế cho một nhóm ký tự bất kỳ. Các ký hiệu gộp này chỉ dùng trong một số lệnh như tìm kiếm, thay thế, xóa,...

Ví dụ : Trong lệnh tìm kiếm ta muốn tìm các tập có tên là LOP1.DAT, LOP2.DAT, LOP3.DAT, có thể được đặt bằng một tên gộp là LOP?.DAT trong đó dấu ? đại diện cho 1 ký tự là 1,2 và 3 đứng sau chữ LOP.

\*.DOC là tên nhóm các tập văn bản trong WORD

\*.\* là tên nhóm đại diện cho tất cả mọi tập.

#### 3.2 Thư mục (Directory hay folder)

Để tìm kiếm thông tin nhanh và quản lý tốt các tập trên các thiết bị lưu trữ thông tin của máy tính người ta tổ chức thông tin theo thư mục.

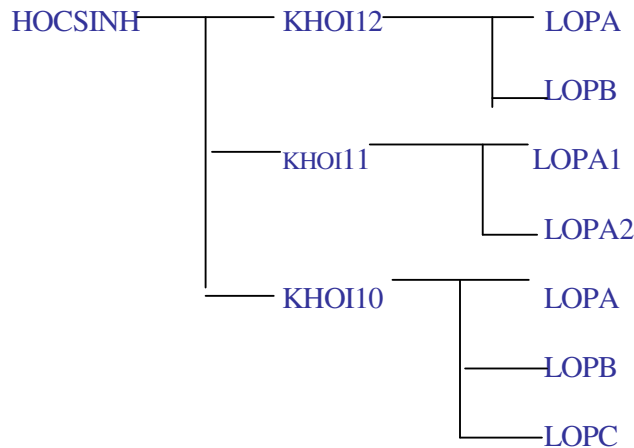
\* Thư mục là một ngăn logic chứa tập trong các thiết bị lưu trữ thông tin của máy tính (đĩa từ, băng từ hoặc đĩa CD). Giống như một tổ chức thư viện, các thông tin cũng được sắp xếp vào từng ngăn.

Mỗi thư mục được đặt một tên duy nhất và 2 thư mục không được trùng tên. Tên thư mục đặt giống như phần tên chính của tệp.

\* Tổ chức thư mục trên đĩa từ: Thư mục được tổ chức theo mô hình phân cấp ( mô hình cây : TREE): Mỗi ngăn chứa gọi là một thư mục (Directory), ngăn to nhất bao trùm trên toàn bộ đĩa gọi là thư mục gốc (ROOT), trong thư mục gốc có chứa các thư mục con ( ngăn con) và các tệp, trong thư mục con lại có chứa các thư mục con của nó ( thư mục cháu) và các tệp,...

Một mô tả đầy đủ cấu trúc của tổ chức thư mục kể từ thư mục gốc đến các thư mục con bên trong gọi là cây thư mục (TREE).

Ví dụ: Thư mục HOCSINH có cấu trúc như sau



## 4. Mã hoá

### 4.1- Khái niệm mã hoá

\* Mã hoá là thuật toán nhằm gán cho mỗi đối tượng trong một nhóm số.

Ví dụ: Đánh số báo danh cho các thí sinh trong phòng thi là mã hoá, kết quả là mỗi thí sinh sẽ nhận được một số báo danh, chỉ cần biết số báo danh của thí sinh ta sẽ tìm được mọi thông tin về thí sinh đó. Mỗi trường sẽ có cách đánh số báo danh riêng hay cách mã hoá riêng.

\* Như phần trên ta đã biết các thông tin trong máy tính đều được biểu diễn ở dạng hệ 2. Các thông tin đưa vào máy vi tính bao gồm các chữ cái la tinh A..Z, a..z, các số 0..9 và các dấu (gọi chung là kí tự - Character), bảng kí tự đó sẽ được mã hoá tương ứng với một bảng mã số. Số lượng các kí tự nhỏ hơn hoặc bằng 256, nếu dùng mã nhị phân (số hệ 2) thì chỉ cần dùng khuôn 8 bit là đủ ( có thể biểu diễn số từ 0 đến  $2^8 = 255$ ). Do vậy để mã hoá cho các kí tự được đưa vào máy vi tính người ta dùng mã nhị phân trong khuôn 8 bit.

## 4.2- Mã ASCII (American Standard Code for Information Interchange)

\* Bảng mã ASCII là bảng mã chuẩn của Mỹ và được sử dụng thông dụng hiện nay. Mã ASCII dùng mã nhị phân trong khuôn 8 bit. Bảng mã này đánh số thứ tự cho các ký tự như sau:

| Các ký tự                             | Mã số        |
|---------------------------------------|--------------|
| Các ký tự điều khiển :                | 0 ---> 31    |
| Khoảng cách (Space Bar) :             | 32           |
| Dấu ! " # \$ % & ' ( ) + , - . /      | 33 ---> 47   |
| Các số từ 0 đến 9 :                   | 48 ---> 57   |
| Dấu : ; < = > ? @                     | 58 ---> 64   |
| Các chữ cái to A ---> Z :             | 65 ---> 90   |
| Dấu [ \ ] ^ _ `                       | 91 ---> 96   |
| Các chữ cái nhỏ a --> z :             | 97 ---> 122  |
| Dấu {   } ~ <DEL>                     | 123 ---> 127 |
| Các ký tự đặc biệt , ký tự kẻ khung : | 128 ---> 255 |

ở đây ta sử dụng số hệ 10 để viết cho gọn và dễ nhớ, trong máy mỗi số sẽ được biểu diễn bằng một nhóm 8 bit trong hệ 2.

Ví dụ: Ký tự "A" có mã  $65_{10}$ , biểu diễn trong máy tính là  $001000001_2$

Ký tự "B" có mã  $66_{10}$ , biểu diễn trong máy tính là  $001000010_2$

## 5. Đại số logic

### 5.1- Khái niệm mệnh đề logic, biến logic, hàm logic

\* Mệnh đề logic: Mệnh đề logic là một câu nói hoặc câu viết có tính chất khẳng định hoặc phủ định một sự kiện. Mỗi mệnh đề logic đều có thể đặt được câu hỏi có đúng không hoặc có sai không.

- Các câu cảm thán, các câu mệnh lệnh hoặc các khẩu hiệu không gọi là mệnh đề logic. Các câu chung chung chẳng đúng mà cũng chẳng sai cũng không gọi là mệnh đề logic.

Ví dụ : "Tôi đã già" ; "Anh A học giỏi" là mệnh đề logic

"Hãy làm cho tôi việc này" không phải là mệnh đề logic

"Ôi hôm nay trời đẹp quá" là câu cảm thán không gọi là mệnh đề logic

- Mỗi mệnh đề chỉ nhận một giá trị hoặc "đúng" ( TRUE) hoặc "sai" (FALSE) tương tự như một mạch điện hoặc "đóng" hoặc "tắt". Môn logic mệnh đề không quan tâm đến câu cú ngữ pháp của mệnh đề mà chỉ xét đến tính đúng sai của mệnh đề.

- Từ các mệnh đề đơn giản ta có thể xây dựng lên các mệnh đề phức tạp hơn nhờ các phép liên kết "Không", "Và", "Hoặc"

Ví dụ: Các mệnh đề đơn A = "Hà nội đông dân "

B = "Hà nội có nhiều cây xanh "

X = "An là con liệt sĩ "

Y = "An là bộ đội "

Phép "Và" liên kết A và B cho C = "Hà nội đông dân và có nhiều cây xanh",

Phép "Không" với X cho Z= " Không phải An là con liệt sĩ".

Phép "Hoặc" liên kết X và Y cho Z = "An là con liệt sĩ hoặc An là bộ đội"

- Các phép "Không", "Và", "Hoặc" cùng với các mệnh đề làm thành một đại số gọi là đại số logic hay đại số mệnh đề

- \* Hai giá trị TRUE ( đúng) và FALSE ( sai) là 2 hằng logic, với TRUE > FALSE.
- \* Biến logic: Là biến chỉ có thể nhận 1 trong 2 giá trị logic là đúng (TRUE) hoặc sai ( FALSE).

Ví dụ: X= "Số m là số âm " , trong ví dụ này X là một biến logic vì nó có thể nhận giá trị TRUE hoặc FALSE tùy theo giá trị của m, giả sử m=-5 thì X nhận giá trị TRUE, nếu m=7 thì X nhận giá trị FALSE.

\* Hàm logic:

- Những bài toán Logic thường được phát biểu dưới dạng các câu nói hoặc câu viết xác định các yêu cầu và các ràng buộc đối với hệ thống mà bài toán giải quyết. Ta có thể biểu diễn sự liên kết giữa các mệnh đề bằng một biểu thức Logic hoặc được gọi là hàm Logic.

- Hàm logic là một hàm của các biến logic. Kết quả của hàm trả về 1 giá trị logic.

Ví dụ: Biến X="Sinh viên có hộ khẩu Hà nội"

Biến Y="Sinh viên có tuổi > 20 "

Hàm F= X "và" Y hay F= X AND Y có nghĩa F = "Sinh viên có hộ khẩu Hà nội và có tuổi > 20 ". Như vậy ứng với những giá trị X, Y khác nhau thì hàm F sẽ cho các giá trị khác nhau được mô tả theo bảng sau:

| X     | Y     | X AND Y |
|-------|-------|---------|
| FALSE | FALSE | FALSE   |
| FALSE | TRUE  | FALSE   |
| TRUE  | FALSE | FALSE   |
| TRUE  | TRUE  | TRUE    |

Các giá trị có thể của một hàm logic được biểu diễn dưới dạng một bảng, bảng này được gọi là bảng chân lý. Hàm logic có n biến thì bảng chân lý sẽ có  $2^n$  giá trị có thể của hàm.

## 5.2- Các toán tử logic cơ bản

Các phép toán trong đại số logic gọi là các toán tử logic. Sau đây sẽ trình bày 4 toán tử logic cơ bản theo thứ tự ưu tiên của các toán tử, các toán tử logic khác đều có thể được biểu diễn qua 4 toán tử cơ bản này.

\* Toán tử NOT ( phủ định hay đảo)

Bảng chân lý:

| X     | NOT X |
|-------|-------|
| FALSE | TRUE  |
| TRUE  | FALSE |

\* Toán tử AND (và )

Bảng chân lý:

| X     | Y     | X AND Y |
|-------|-------|---------|
| FALSE | FALSE | FALSE   |
| FALSE | TRUE  | FALSE   |
| TRUE  | FALSE | FALSE   |
| TRUE  | TRUE  | TRUE    |

\* Toán tử OR (hoặc )

Bảng chân lý:

| X     | Y     | X OR Y |
|-------|-------|--------|
| FALSE | FALSE | FALSE  |
| FALSE | TRUE  | TRUE   |
| TRUE  | FALSE | TRUE   |
| TRUE  | TRUE  | TRUE   |

\* Toán tử XOR (hoặc loại trừ )

Bảng chân lý:

| X     | Y     | X XOR Y |
|-------|-------|---------|
| FALSE | FALSE | FALSE   |
| FALSE | TRUE  | TRUE    |
| TRUE  | FALSE | TRUE    |
| TRUE  | TRUE  | FALSE   |

\* Thứ tự ưu tiên của 4 toán tử trên là: NOT, AND, OR, XOR

\* Biểu thức logic: Là sự kết hợp giữa các hằng logic, biến logic, hàm logic và các toán tử logic. Kết quả của biểu thức logic là 1 hằng logic ( TRUE hoặc FALSE).

Có thể tính giá trị của biểu thức logic theo thứ tự sau:

- Thay giá trị vào các biến nếu có.
- Thực hiện các phép tính số học, các phép tính so sánh nếu có.
- Thực hiện các toán tử logic theo thứ tự ưu tiên đã nêu.

Các phép tính số học và phép so sánh các số ta đã biết, chú ý hơn về so sánh 2 xâu kí tự.

\* So sánh 2 xâu kí tự: Khi so sánh 2 xâu kí tự ta tiến hành so sánh mã ASCII của từng cặp kí tự tương ứng từ 2 xâu, kết quả của phép so sánh theo 3 trường hợp sau:

- Nếu gặp một cặp có mã khác nhau thì xâu chứa kí tự có mã nhỏ hơn là xâu nhỏ hơn.
- Nếu hai xâu có tất cả các cặp kí tự có mã bằng nhau thì 2 xâu bằng nhau.
- Nếu trong quá trình so sánh một xâu đã hết các kí tự, một xâu vẫn còn kí tự thì xâu ít kí tự hơn là xâu nhỏ hơn.

Ví dụ: "ABCD" < "AX"

"ABCD" = "ABCD"

"ABC" < "ABCDE"

Ví dụ: Tính giá trị biểu thức logic sau

("AB" > "1234") OR ( sinx > 2) AND NOT ((X<sup>2</sup> + Y<sup>2</sup> + 2XY) ≥ 0 )

= TRUE OR FALSE AND NOT TRUE

= TRUE OR FALSE AND FALSE

= TRUE OR FALSE = TRUE

## Bài tập chương I

### Bài 1: Chuyển đổi số trong các hệ đếm

- a, Chuyển  $33_{10}$ ,  $35_{10}$  sang số hệ 2, hệ 16
- b, Chuyển  $10111001_2$ ,  $1001101_2$  hệ 10, hệ 16
- c, Chuyển  $20A_{16}$ ,  $3EB_{16}$  sang hệ 2, hệ 10

### Bài 2: Thực hiện các phép tính trong hệ 2

- a,  $x=36_{10}$ ,  $y=18_{10}$  Tính  $x+y$ ,  $x-y$ ,  $x*y$ ,  $x/y$
- b,  $x=20_{10}$ ,  $y=5_{10}$  Tính  $x+y$ ,  $x-y$ ,  $x*y$ ,  $x/y$

### Bài 3: Tính giá trị của các biểu thức logic sau

a, NOT (LOAI < 'C') AND ( GIA >= 500000) OR (LOAI <= 'B') AND (GIA <=200000)

Tính khi : 1- LOAI='D', GIA= 500000

2- LOAI='A', GIA= 300000

3- LOAI='B', GIA= 100000

b, NOT (DT>8) AND (KV=3) AND (DIEM>=16) OR (DT>8) AND (KV=1) AND (DIEM>=15)

Tính khi : 1- DT=6, KV=3, DIEM=17

2- DT=9, KV=1, DIEM=16

3- DT=6, KV=3, DIEM=15

c, (TEN='H\*') AND (TINH='T\*') OR NOT(TEN='D\*') AND (TINH < > 'H\*')

Chú ý: Dấu \* thay cho một dãy kí tự bất kỳ.

Tính khi : 1- TEN='HA', TINH='THAI BINH'

2- TEN='DUNG', TINH=' HA NOI'

3- TEN='MAI', TINH='THAI NGUYEN'

d,  $(\sin^2 X < 2)$  OR NOT( 'ABC' > 'AXY') AND  $((X+Y)^2 \geq 2XY)$

e,  $((\cos X + \sin X) > 2)$  OR ( 'ABC' < 'XY') AND NOT ((3<7) = ('1'>'a'))

f, NOT ((lg100 + 1!) > 2) AND ((3!+2)<6) AND NOT (('123'<'ABC') > ('abc'<'ABC'))

## Câu hỏi ôn tập chương I

1. Tin học là gì?
2. Trong máy tính dùng các hệ đếm nào?
3. Biểu diễn số trong máy tính như thế nào?
4. Trình bày các đơn vị thông tin sử dụng hiện nay.
5. Trình bày cách chuyển đổi số giữa các hệ đếm.
6. Trình bày các phép tính trong hệ 2.
7. Trình bày cách đặt tên tệp, tên thư mục.
8. Bảng mã thông dụng trong tin học hiện nay là bảng mã nào? Trình bày về bảng mã đó.
9. Trình bày các toán tử logic cơ bản theo thứ tự ưu tiên.
10. Trình bày cách so sánh 2 xâu kí tự.

## CHƯƠNG II: CẤU TRÚC MÁY VI TÍNH

Để sử dụng tốt máy vi tính phục vụ cho chuyên môn của mình trước tiên phải hiểu chức năng và cấu tạo của máy vi tính. Đây là một giáo trình Tin học dạy cho sinh viên không phải chuyên ngành Tin học do đó không đi sâu vào cấu tạo chi tiết máy vi tính mà chỉ giới thiệu cấu trúc của máy vi tính hay các khối chính của máy vi tính. Chương này cung cấp các kiến thức chính như sau: Chức năng và sơ đồ cấu trúc của máy vi tính, CPU, bộ nhớ, các thiết bị ngoại vi, cách tổ chức một phòng máy.

### 1- Chức năng và sơ đồ cấu trúc của máy vi tính

#### 1.1- Chức năng của máy vi tính

Máy vi tính là máy vi điện tử hoạt động theo chương trình và có các chức năng cơ bản sau đây:

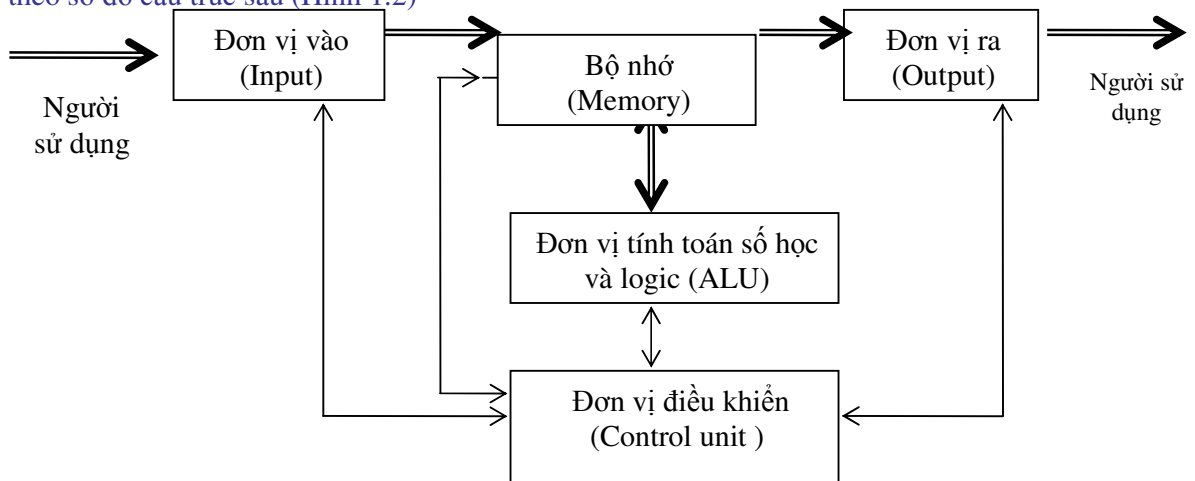
- \* Tiếp nhận thông tin vào và đưa thông tin ra ( Vào / Ra : Input / Output). Với chức năng này người sử dụng giao tiếp được với máy vi tính.

- \* Xử lý thông tin hay biến đổi thông tin, đây là chức năng quan trọng nhất của máy vi tính, thực hiện nhiệm vụ tự động hoá xử lý thông tin thay cho con người.

- \* Lưu trữ thông tin. Các thông tin sử dụng trên máy vi tính nếu cần sẽ được lưu trữ để trao đổi và sử dụng cho nhiều lần sau.

#### 1.2- Sơ đồ cấu trúc máy vi tính

Để đảm nhận được các chức năng cơ bản trên, máy vi tính được thiết kế với các khối chính theo sơ đồ cấu trúc sau (Hình 1.2)



Hình 1.2

Trong sơ đồ trên dùng 2 kí hiệu:

==> Tín hiệu thông tin

-> Tín hiệu điều khiển



\* Các tín hiệu thông tin từ người sử dụng qua đơn vị vào ( bàn phím, chuột, ổ đĩa,...) được đưa vào bộ nhớ, từ bộ nhớ các thông tin được chuyển vào đơn vị tính toán số học và logic để xử lý, xử lý xong kết quả được chuyển vào bộ nhớ, từ bộ nhớ chuyển đến đơn vị ra ( màn hình, máy in, máy vẽ, ...) và tới người sử dụng.

\* Các tín hiệu điều khiển được thiết lập giữa đơn vị điều khiển và các khối khác trong dàn máy vi tính để chuyển các tín hiệu điều khiển của đơn vị điều khiển tới các khối khác và chuyển các tín hiệu phản hồi từ các khối khác về đơn vị điều khiển.

\* Các khối trong máy vi tính được nối với nhau bằng các cáp truyền dẫn ( BUS), gồm 3 loại cáp đó là cáp địa chỉ (Bus Address), cáp Dữ liệu (Bus Data) và cáp điều khiển (Bus Control).

\* Trong máy vi tính 2 khối quan trọng nhất là đơn vị tính toán số học và logic, đơn vị điều khiển nằm trong cùng một vỏ của mạch vi xử lý (Micro Processor), được gọi là đơn vị xử lý trung tâm (CPU : Central Processor Unit) hay bộ vi xử lý(Micro Processor).

\* CPU và bộ nhớ nằm trong thân máy. Đơn vị vào/ ra nằm bên ngoài máy được gọi chung là thiết bị ngoại vi, ngày nay các thiết bị ngoại vi của máy vi tính rất phong phú và đa dạng.

\* Ngoài các khối cơ bản kể trên, để cho máy vi tính hoạt động cần có một bộ nguồn nối với máy.

## **2- Các bộ phận cơ bản của máy vi tính**

### **2.1- Bộ xử lý trung tâm(CPU: Central Processor Unit )**

#### **a) Chức năng của CPU**

CPU là bộ chỉ huy của máy vi tính, có các chức năng sau:

- Thực hiện việc nhận lệnh, giải mã lệnh và điều khiển các khối khác thực hiện lệnh.
- Thực hiện các phép tính số học, logic và các phép tính khác.
- Sinh ra các tín hiệu địa chỉ trên máy.

#### **b) Cấu tạo của CPU**

CPU bao gồm các phần sau:

\* Đơn vị điều khiển (Control unit): Thực hiện việc nhận lệnh, giải mã lệnh và điều khiển các khối khác thực hiện lệnh và sinh ra các tín hiệu địa chỉ trên máy để quản lý bộ nhớ.

\* Đơn vị tính toán số học và logic (Arithmeric – Logic Unit): Bao gồm các vi mạch tính để thực hiện các phép tính số học, logic và các phép tính khác.

\* Thanh ghi (Register) là một cấu trúc gồm 16 bit (hoặc 32 bit) nhớ liền kề nhau được thiết lập ngay trong mạch vi xử lý. Các thanh ghi này được phân thành 4 nhóm theo mục đích sử dụng sau:

- Nhóm 1: 4 thanh ghi đa năng kí hiệu là AX,BX,CX,DX được sử dụng cho nhiều mục đích khác nhau.

- Nhóm 2: 4 thanh ghi địa chỉ đoạn (Segment) bộ nhớ, thanh ghi CS (Code segment) chứa địa chỉ đoạn mã lệnh, thanh ghi DS (Data Segment) chứa địa chỉ đoạn dữ liệu, thanh ghi SS (Stack Segment) chứa địa chỉ đoạn ngăn xếp, thanh ghi ES (Extra Segment) chứa địa chỉ đoạn dữ liệu mở rộng.

- Nhóm 3: 5 thanh ghi địa chỉ tương đối (offset), thanh ghi IP (Instruction Pointer) trỏ tới ô chứa lệnh trong đoạn CS; thanh ghi SI (Source Index) trỏ tới ô chứa dữ liệu nguồn trong

đoạn DS, Thanh ghi DI (Destination Index) trở tới ô chứa dữ liệu đích trong đoạn DS, Thanh ghi SP (Stack Pointer) trở tới ô chứa dữ liệu trong đoạn ngăn xếp SS và thanh ghi BP (Base Pointer) trở tới ô dữ liệu trong đoạn chứa dữ liệu mở rộng ES.

Nhóm 4: 1 thanh ghi cờ (Flag) để ghi các trạng thái (ta gọi là cờ) xảy ra trong các phép toán số học.

## 2.2- Bộ nhớ ( Memory)

Bộ nhớ hay còn gọi là bộ nhớ trong ( bộ nhớ ) chính của máy vi tính.

\* Chức năng của bộ nhớ: Bộ nhớ dùng để chứa các thông tin cần thiết như chương trình, dữ liệu trong quá trình máy hoạt động.

\* Tổ chức của bộ nhớ:

- Các thông tin được cất trong bộ nhớ dưới dạng mã nhị phân ( các bit 0 hoặc 1).

- Một nhóm các bit có thể là 16 bit hoặc 32 bit,... gọi là một ô nhớ. Mỗi ô nhớ được đánh một số thứ tự trong hệ 16 gọi là địa chỉ của ô nhớ. Thông tin ghi bên trong ô nhớ là nội dung của ô nhớ. Dùng địa chỉ để truy nhập vào ô nhớ để đọc nội dung hoặc ghi nội dung vào ô nhớ.

- Kích thước của bộ nhớ hay dung lượng của bộ nhớ là số lượng thông tin mà bộ nhớ có khả năng chứa được, thường dùng đơn vị MB hoặc GB.

\* Phân loại bộ nhớ: Theo tính chất thông tin chứa trong bộ nhớ người ta chia thành bộ nhớ ROM và RAM.

- ROM ( Read Only Memory): là bộ nhớ cố định cho phép chỉ đọc thông tin mà không ghi thông tin vào được. ROM là bộ nhớ cứng do hãng chế tạo cài đặt sẵn các chương trình bên trong, bao gồm các chương trình kiểm tra và các chương trình cơ sở cốt lõi nhất của máy vi tính. Các thông tin trong ROM sẽ không bị mất đi khi ta tắt nguồn của máy.

- RAM (Random Access Memory): Là bộ nhớ mềm, có thể thay đổi, truy nhập một cách ngẫu nhiên. RAM làm từ các mạch vi mạch (gọi là Chip nhớ). RAM dùng để ghi chương trình của hệ điều hành nạp vào từ đĩa khởi động, chương trình và dữ liệu của người sử dụng. Có thể đọc và ghi thông tin vào RAM. Thông tin trong RAM sẽ bị mất đi khi ta tắt nguồn của máy, do vậy ta phải lưu trữ thông tin ra bộ nhớ ngoài.

## 2.3- Thiết bị ngoại vi

### a) Bàn phím (Keyboard)

\* Bàn phím là thiết bị vào thông dụng của máy vi tính. Dùng bàn phím có thể đưa vào máy các lệnh điều khiển, chương trình, dữ liệu.

\* Nguyên tắc của bàn phím: ấn một phím thì gây ra sự tiếp điện giữa một dây dọc và một dây ngang tạo ra một xung điện , xung điện này qua chương trình điều khiển bàn phím sẽ đưa vào máy mã ASCII của kí tự của phím đó.

\* Bàn phím gồm 5 khu vực sau:

- Khu phím máy chữ gồm các phím chữ, phím dấu và phím số như các phím trên bàn phím máy chữ trong đó có phím cách (Space bar)

- Khu phím chức năng F1,F2,..F12 có tác dụng thực hiện một chức năng hoàn chỉnh nào đó khi ta gõ phím đó, tác dụng của nó được cài đặt tùy theo chương trình. Thí dụ DOS dùng phím F1 để nhắc lại từng kí tự của câu lệnh vừa gõ trước, F3 để nhắc lại toàn bộ câu lệnh vừa gõ.

- Khu vực phím số ở bên phải bàn phím

- Khu phím dịch chuyển con trỏ màn hình gồm các phím dịch lên, dịch xuống, dịch trái, dịch phải, dịch về đầu màn hình (Home), dịch xuống cuối (End), dịch lên một trang (PgUp), dịch xuống một trang (PgDn), Dịch trái và xoá (Back space), phím TAB dịch con trỏ đi một khoảng 8 cột (gọi là một tab)

- Khu phím điều khiển:

ESC để thoát ra khỏi chương trình hoặc lệnh đang thực hiện

CapsLock để chuyển đổi thường xuyên kiểu chữ nhỏ sang to và ngược lại

Shift có tác dụng chuyển tạm thời chữ to thành nhỏ hoặc ngược lại khi đè giữ phím đó và gõ chữ. Với các phím có 2 kí tự thì nếu gõ được kí tự dưới, nếu đè shift và gõ thì cho kí tự trên.

Insert - chèn kí tự vào vị trí con trỏ

Delete - xoá kí tự tại vị trí con trỏ

Print Screen - In màn hình

Scroll Lock - bật hoặc tắt trạng thái cuộn màn hình

Pause/Break - Chờ hoặc ngắt chương trình

Ctrl và Alt dùng để mở rộng chức năng của bàn phím.

Num Lock - để kích hoạt bàn phím số, bàn phím số chỉ có tác dụng khi nào đèn Num Lock sáng.

## **b) Màn hình ( Display)**

\* Màn hình là thiết bị ra thông dụng tối thiểu của máy vi tính. Các kí hiệu ta gõ vào từ bàn phím, các kết quả xử lý, các thông báo của máy vi tính đều được hiện ra trên màn hình.

\* Màn hình có cấu tạo vật lý giống như một màn hình vô tuyến thông thường cũng gồm có ống phóng tia điện tử, mạch quét ngang và quét dọc, tia điện tử đập vào màn hình tạo thành một điểm sáng, bộ điều khiển CRT tạo tín hiệu bật hoặc tắt tia điện tử theo tín hiệu lấy ra từ bộ nhớ màn hình 1 hay 0, tia điện tử quét từ trái sang phải tạo thành một dòng, từ dòng trên cùng xuống dòng dưới cùng tạo thành một màn hình. Tần số quét màn hình bằng 60Hz nghĩa là trong 1 giây đồng hồ màn hình được làm mới lại từ đầu 60 lần, thời gian hồi dọc để tia điện tử chuyển về góc trái trên cùng là 1,25ms

\* Có 2 kiểu màn hình: kiểu màn hình văn bản và màn hình đồ hoạ:

- Màn hình văn bản (Text mode) được phân thành 2 chế độ: 80 cột ( kí tự) 25 dòng là dạng chuẩn hoặc 40 cột ( kí tự) 25 dòng.

- Màn hình Đồ hoạ (Graphic Mode) được thiết kế gồm nhiều điểm sáng (Pixel) theo hai chiều ngang và dọc. Màn hình đồ hoạ được phân thành nhiều chế độ khác nhau tùy theo độ phân giải, chẳng hạn chiều ngang 640 pixel chiều dọc 200 pixel (640x200), 640x350, 640x480, 1024x768. Màn hình càng có độ phân giải cao thì hình ảnh càng mịn.

- Màu của màn hình được tạo ra bởi sự pha trộn của 4 yếu tố sau: 3 màu cơ bản là đỏ (Red), xanh lá cây (Green), xanh da trời (Blue) và độ chói. Với tổ hợp của 3 màu cơ bản ở độ chói khác nhau của từng màu mà máy tính có thể tạo ra tới 256 màu khác nhau.

## **c) Ổ đĩa và đĩa từ**

Ổ đĩa và đĩa từ là thiết bị vào / ra : Có thể đọc thông tin từ đĩa từ và ghi thông tin vào đĩa từ.

\* Ổ đĩa là khối hình chữ nhật có chứa các thiết bị và linh kiện để làm việc với đĩa từ. Máy vi tính có 2 loại ổ đĩa từ là ổ đĩa cứng (Hard Disk/ Fixed Disk) và ổ đĩa mềm (Soft Disk/Flopy Disk).

- ổ đĩa cứng được lắp trong thân máy, nếu máy có 1 ổ đĩa cứng thì tên ổ đĩa là C: , nếu máy có 2 ổ đĩa cứng thì tên là C: , D: . ổ đĩa cứng chứa các đĩa cứng có dung lượng lưu trữ lớn từ 40 MB đến hàng chục GB.

- ổ đĩa mềm lắp ngay bên ngoài vỏ máy, 2 ổ mềm có tên là A: , B: , ổ đĩa mềm chứa các đĩa mềm có dung lượng lưu trữ nhỏ hơn đĩa cứng, hiện nay thông dụng là 1,44MB.

\* Đĩa từ có hình dạng tròn được cấu tạo bằng chất dẻo ( đĩa mềm ) hoặc nhôm ( đĩa cứng) có phủ các hạt sắt từ trên bề mặt để lưu trữ thông tin dựa theo đặc tính nhiễm từ của các hạt sắt từ. Các thông tin lưu trữ trên đĩa từ ở dạng mã nhị phân ( bit 0 hoặc 1).

- Mỗi đĩa từ mềm gồm có hai mặt từ hoá được đánh số là mặt 0/1, có 2 đầu từ tiếp xúc trên dưới để đọc và ghi thông tin, mỗi mặt phân thành nhiều rãnh (Track) đánh số từ 0 trở đi ,rãnh ngoài gần mép đĩa là rãnh số 0, mỗi mặt được phân thành nhiều cung (sector) được đánh số từ 1 trở đi, (đĩa 1,2M phân thành 80 rãnh và 15 sector, đĩa 1,44 M phân thành 80 rãnh và 18 sector ), mỗi cung chứa 512 byte.

- Đĩa cứng gồm nhiều mặt đĩa cùng trục quay tạo thành khối trụ gọi là cylinder, số mặt tùy theo dung lượng của ổ đĩa, chẳng hạn ổ đĩa có 8 đầu từ (Head) đánh số từ 0 đến 7 gồm 4 mặt đĩa, mỗi mặt của đĩa cứng được phân thành 17 sector, mỗi mặt gồm nhiều rãnh (Đĩa cứng XT gồm 305 rãnh, đĩa cứng AT gồm 615 rãnh ), mỗi rãnh trên một cung chứa 1024 byte.

#### **d) Máy in (Printer)**

\* Máy in là thiết bị ra của máy vi tính, dùng để in ra kết quả xử lý, dữ liệu, chương trình, thông báo của máy vi tính.

\* Máy in dùng trong máy tính gồm nhiều loại khác nhau hoạt động theo các nguyên lý khác nhau. Ta có thể phân thành các loại cơ bản sau:

- Máy in màu (Ploter) gồm 8 bút vẽ màu khác nhau hoạt động trên nguyên tắc bút vẽ.

- Máy in Laser hoạt động trên nguyên tắc dùng tia Laser ép nóng chảy bột mực khô bám trên lô

- Máy in kim (9 kim hoặc 24 kim) hoạt động trên nguyên tắc kim phun bắn vào băng mực và tạo ra trên giấy các nốt chấm đen.

Hiện nay máy in Laser được sử dụng rộng rãi vì có ưu điểm là in nhanh và chất lượng tốt nhưng giá đắt hơn máy in kim. Máy in kim giá rẻ hơn và bền hơn nhưng chất lượng không cao và in chậm hơn.

#### **e) Thiết bị chuột**

Chuột là một thiết bị vào của máy vi tính. Có thể chọn lệnh, chọn các đối tượng, di chuyển và thay đổi kích thước các đối tượng bằng chuột.

Thông thường chuột có 2 nút bấm. Nút trái dùng cho phần lớn các thao tác, nút phải tùy theo từng chương trình mà có các chức năng khác nhau, thông thường nhấn nút phải để hiện ra các lệnh có thể thực hiện được tại vị trí vừa chọn của chuột. lệnh này tùy thuộc phần mềm đang sử dụng.

#### **f) Các thiết bị khác**

Các thiết bị ngoại vi của máy vi tính hiện nay rất phong phú, ngoài các thiết bị cơ bản nêu trên còn các thiết bị khác như : Đĩa CD-ROM, máy chiếu (Projecter), máy quét ảnh

(Scanner), ... Tất cả các thiết bị ngoại vi giúp cho việc đưa thông tin vào máy ở nhiều dạng từ các kí tự đến âm thanh, hình ảnh.

### **3- Tổ chức một phòng máy tính và cách bảo quản**

#### **3.1- Tổ chức một phòng máy**

Khi thiết lập một phòng máy tính, cần chọn vị trí thoáng mát, tránh ánh nắng mặt trời, tránh nơi có nam châm và từ trường. Không nên đặt máy gần cửa sổ, nhà bếp. Nên đặt máy vị trí cách xa tường để tránh ẩm và nên để máy sao cho những người ngồi làm máy không quay lưng vào đuôi máy. Nhìn chung, máy tính nên được chạy qua ổn áp để tránh những thay đổi đột ngột điện áp làm ảnh hưởng đến máy tính. Nên lắp điều hoà, máy hút ẩm trong phòng máy để đảm bảo nhiệt độ và độ ẩm trong phòng được chuẩn. Tốt nhất là từ 20<sup>0</sup> C đến 25<sup>0</sup>C. Trong phòng máy nên trải thảm cách điện và máy nên nối đất để khử tĩnh điện.

Trong phòng máy phải dùng các biện pháp để ngăn chặn bụi bặm thâm nhập vào máy tính. Vì vậy, không nên đi giày, dép vào phòng máy tính, không hút thuốc trong phòng máy, không để các chất hay gây cháy nổ trong phòng máy. Không để kiến, gián và các loại côn trùng khác vào phòng máy tính.

#### **3.2 Bảo quản đĩa từ, màn hình, bàn phím**

\*Đối với đĩa từ :

- Đĩa cứng: Tuổi thọ của đĩa cứng thường từ 8.000 đến 20.000 giờ. Đầu máy phải đặt trên bàn bằng phẳng, vững chắc, không lung lay. Khi máy đang vận hành, nếu vô ý đập mạnh vào đầu máy, hoặc vô ý làm lung lay mạnh thì đầu từ sẽ đập mạnh vào đĩa cứng làm bong lớp từ phủ ngoài hoặc chính đầu từ bị vỡ hoặc lệch dẫn đến tình trạng ổ cứng không sử dụng được.

Khi nhiệt độ tăng cao thì lớp từ tính trên mặt đĩa sẽ bị bong ra hoặc đĩa bị cong. Trong trường hợp này, dữ liệu trên đĩa cứng không thể lấy ra được và bạn cũng không truy nhập thông tin lên đĩa được.

Khi di chuyển đầu máy, để đảm bảo an toàn cho đĩa cứng, bạn cần chạy chương trình “Park” để di chuyển đầu từ ra mé ngoài của đĩa cứng vì ở vị trí này không có dữ liệu.

Nhìn chung, bạn nên thường xuyên kiểm tra tình trạng của đĩa cứng bằng các chương trình tiện ích như Scandisk, NDD...

Khi mở máy, bạn phải tuân thủ các nguyên tắc về vận hành máy như sau: cắm ổn áp vào nguồn lưới, bật ổn áp, bật công tắc ở đầu máy. Khi tắt máy, bạn làm động tác ngược lại: tắt máy tính, tắt ổn áp, rút phích cắm ổn áp ra khỏi điện lưới.

- Đĩa mềm thường có chất lượng không được tốt bằng đĩa cứng, Vì vậy, nếu bạn lưu trữ dữ liệu trên đĩa mềm thì cần phải thường xuyên kiểm tra tình trạng của đĩa và dữ liệu được lưu trữ trên đĩa. Khi dùng đĩa, bạn không được chạm tay vào phần phía trong của đĩa mà chỉ được cầm phần vỏ nhựa bên ngoài. Không để bụi rơi vào phần lõi nhựa bên trong, không để đĩa gần nơi có từ tính, tránh xa các nơi có tia X quang. Không để vật nặng đè lên đĩa, không được để đĩa bị cong, tránh tuyệt đối để đĩa mềm nơi ẩm ướt, nhiệt độ cao. Tránh va chạm mạnh vào ổ đĩa. Khi di chuyển máy, cần có đĩa giả đưa vào ổ đĩa để đầu từ không bị lắc, rung. Khi đĩa bị bẩn do bụi, bạn không nên đưa đĩa vào ổ vì có thể sẽ làm bẩn đầu từ hoặc thậm chí có thể làm hỏng đầu từ.

\*Màn hình: Khi không dùng máy, bạn phải phủ kín máy để che bụi, khói và nhất là chất lỏng bám vào. Không được bít các lỗ thông hơi ở hai bên hông và phía trên màn hình. Thường

xuyên lau chùi bụi bặm ở các khe rãnh thông hơi. Bạn cũng cần lưu ý rằng: khi làm việc, màn hình và đầu máy phát nhiệt nhiều nên hút bụi rất mạnh.

Khi làm việc, điện thế trong màn hình có thể lên tới hàng nghìn vôn nên có thể gây chết người trong giây lát. Vì vậy, khi máy đang làm việc, bạn không được mở nắp nhựa, tránh để những nơi ẩm thấp. Bạn không nên để màn hình ở những vị trí có thể bị nước mưa hắt vào. Không được đung chạm mạnh vào phía trước màn hình. Khi vận chuyển màn hình khỏi khu vực làm việc, bạn bắt buộc phải cho màn hình vào hộp có đệm xốp.

\* Bàn phím: ở dưới các phím đều có các lò xo để đàn hồi khi gõ phím. Vì vậy, bạn không nên gõ mạnh lên mặt phím. Phải đẩy bàn phím khi không dùng tới. Không được để bàn phím bị bụi bẩn, chất lỏng hoặc nước hắt vào sẽ làm bàn phím bị kẹt, không sử dụng được. Khi bảo quản bàn phím, bạn nên dùng cồn để lau mặt bàn phím nhưng không được để cồn chảy xuống phía dưới bàn phím.

## **Câu hỏi ôn tập chương II**

1. Nêu các chức năng cơ bản của máy vi tính.
2. Tại sao CPU lại là bộ não của máy vi tính? Các thành phần của CPU.
3. Để chứa các thông tin cần thiết khi máy làm việc thì cần có bộ phận nào? Tại sao phải lưu trữ thông tin ra bộ nhớ ngoài?
4. Tại sao lại gọi là các thiết bị ngoại vi? Trình bày các thiết bị ngoại vi thông dụng của máy vi tính.
5. Cách bảo quản đĩa từ, màn hình, bàn phím.

## CHƯƠNG III: HỆ ĐIỀU HÀNH

Để sử dụng máy vi tính trước tiên phải hiểu và biết sử dụng bộ chương trình điều khiển các hoạt động chung của máy vi tính. Bộ chương trình đó chính là hệ điều hành. Chương này trình bày các kiến thức chính về hệ điều hành: Khái niệm hệ điều hành, các hệ điều hành thông dụng hiện nay, hệ điều hành Microsoft Windows 98/2000, khởi động và các thành phần cơ bản của windows 98/2000, thay đổi các biểu tượng hoặc mục chọn trong Windows 98/2000, sử dụng Windows Explorer trong windows 98/2000.

### 1 - Khái niệm hệ điều hành

#### 1.1 - Khái niệm hệ điều hành

\* *Hệ điều hành* là hệ thống các chương trình điều khiển các hành vi cơ bản của dàn máy vi tính. Chỉ khi hệ điều hành được nạp vào trong bộ nhớ thì máy tính mới hoạt động.

Hệ điều hành mục đích giúp người sử dụng máy tính dễ dàng và hiệu quả.

\* Chức năng cơ bản của Hệ điều hành:

- Hệ điều hành điều khiển tất cả hoạt động của máy tính và các thiết bị ngoại vi.
- Hệ điều hành là người thông dịch, cầu nối giữa người sử dụng và máy vi tính.

#### 1.2 - Phân loại hệ điều hành

Có thể chia hệ điều hành thành 2 loại: Hệ điều hành máy tính cá nhân, hệ điều hành mạng

\* Hệ điều hành máy tính cá nhân là hệ điều hành viết để điều khiển một máy tính riêng lẻ còn gọi là máy tính cá nhân.

Các hệ điều hành máy tính cá nhân thông dụng như: MS-DOS, WINDOWS 95, WINDOWS 98, WINDOWS 2000, WINDOWS XP

\* Hệ điều hành mạng là hệ điều hành viết để điều khiển một mạng máy tính bao gồm 1 máy chủ kết nối với các máy trạm, hệ điều hành được cài đặt trong máy chủ.

Theo khoảng cách địa lý có thể phân ra các loại mạng máy tính sau: **Mạng** cục bộ, mạng đô thị, mạng diện rộng, mạng toàn cầu.

- Mạng cục bộ ( LAN - Local Area Network ): là mạng được cài đặt trong phạm vi tương đối nhỏ hẹp như trong một toà nhà, một xí nghiệp...với khoảng cách lớn nhất giữa các máy tính trên mạng trong vòng vài km trở lại.

- Mạng đô thị ( MAN - Metropolitan Area Network ): là mạng được cài đặt trong phạm vi một đô thị, một trung tâm văn hoá xã hội, có bán kính tối đa khoảng 100 km trở lại.

- Mạng diện rộng ( WAN - Wide Area Network ): là mạng có diện tích bao phủ rộng lớn, phạm vi của mạng có thể vượt biên giới quốc gia thậm chí cả lục địa.

- Mạng toàn cầu ( GAN - Global Area Network ): là mạng có phạm vi trải rộng toàn cầu đó là Internet

Các hệ điều hành mạng thông dụng hiện nay là: WINDOWS NT, UNIX, WINDOWS 2000 SERVER

#### 1.3 - Một số hệ điều hành thông dụng

##### \* Hệ điều hành MS-DOS

Hệ điều hành MS DOS là một hệ điều hành ra đời cách đây khá lâu và rất phổ dụng trước khi có sản phẩm cùng hãng của nó là hệ điều hành WINDOWS ra đời.

DOS quản lý, lưu trữ thông tin dưới dạng các tập tin và thư mục.

Giao diện của DOS với người sử dụng là giao diện dòng lệnh.

### \* Hệ điều hành WINDOWS 98/2000

Hệ điều hành WINDOWS 98/2000 do hãng phần mềm MICROSOFT phát hành. Đây là một hệ điều hành theo phong cách hoàn toàn mới và nó nhanh chóng trở thành một trong những hệ điều hành phổ dụng và được yêu thích nhất hiện nay.

Trước kia để làm việc được với hệ điều hành MS DOS, cần phải nhớ rất nhiều lệnh với cú pháp dài dòng và rắc rối, cùng với việc phải đối diện với một màn hình tối om sẽ làm cho công việc trở nên nhàm chán. Hệ điều hành WINDOWS ra đời, tương thích với hệ điều hành MS DOS, đã mang lại rất nhiều tiện lợi trong việc sử dụng. Hệ điều hành WINDOWS vì sử dụng giao diện đồ họa do đó rất dễ sử dụng. Một số đặc điểm nổi trội của WINDOWS 98/2000

- Cung cấp một giao diện đồ họa người-máy thân thiện (GUI- Graphic User Interface).
- Cung cấp một phương pháp điều khiển thống nhất cho mọi ứng dụng trên môi trường WINDOWS 95
- Hoạt động ở chế độ đa nhiệm
- Môi trường Nhúng - Liên kết các đối tượng (OLE - Object Linking and Embeding)
- Tự động nhận dạng và cài đặt trình điều khiển các thiết bị (Plus and Play).
- Hỗ trợ mạng.

### \* Hệ điều hành WINDOWS NT

Windows NT là hệ điều hành mạng được người dùng tin cậy. Qua sử dụng, nó chứng tỏ là hệ điều hành mạng tích hợp nhiều tính năng như các giao thức truyền tin chuẩn, tính năng tìm đường (routing), truy cập từ xa, tốc độ, bảo mật mức C2, giao diện đơn giản, dễ quản trị, đặc biệt đây là hệ điều hành nền tảng cho rất nhiều chương trình và ứng dụng phổ biến hiện nay như MS SQL Server, MS Mail Server, MS Exchange Server 4.0, Internet Information Server (bao gồm FTP Server, Gopher Server, W.W.W. Server), MS Proxy Server, ...

Windows NT là hệ điều hành 32 bit, đa nhiệm có ưu tiên nhằm khai thác hết khả năng của các bộ vi xử lý như Intel x86, RISC và các hệ thống đa xử lý đối xứng (symmetric multiprocessing system).

Bên ngoài là một giao diện người sử dụng giống như Windows, Windows NT đã được thiết kế lại phần hạt nhân (kernel) cần thiết với các hệ điều hành đã có Windows NT thực hiện được hầu hết các chương trình đang chạy trên các họ máy tính x86 và RISC dưới MS-DOS, Windows, MS OS/2 version 1.x và các ứng dụng cùng với các chức năng phát triển về bảo mật và quản trị.

Tính mở: Để duy trì tương thích ví dụ với các ứng dụng viết trên UNIX theo chuẩn POSIX. Tuy nhiên để tận dụng hết khả năng của **WINDOWS NT** ta cần thực hiện các ứng dụng viết riêng cho hệ điều hành 32 bit này mà MS SQL Server là một ví dụ.

### \* Hệ điều hành UNIX

UNIX là hệ điều hành mạng đa nhiệm, đa người dùng. UNIX với sức mạnh và tính tin cậy truyền thống, cộng thêm những khả năng mũi nhọn về truyền thông, kết nối mạng qua TCP/IP, thư điện tử, cơ sở dữ liệu, tính an toàn cao. Đặc biệt, các ứng dụng chuyên nghiệp đều được viết rất hoàn thiện trên UNIX.

Hệ điều hành UNIX đã được phát triển tại phòng thí nghiệm AT&T Bell tại Murray bang New Jersey - một trong những phòng nghiên cứu lớn nhất trên thế giới. Từ khi phiên bản đầu tiên của hệ điều hành UNIX được Ken Thompson thiết kế năm 1969, nó đã trải qua



một quá trình phát triển và ngày càng hoàn thiện. Khi hệ điều hành UNIX đã phát triển, rất nhiều máy tính vẫn còn chạy ở chế độ đơn nhiệm, nghĩa là máy tính chỉ phục vụ được một người trong cùng một khoảng thời gian, do đó người sử dụng không khai thác hết được năng lực cũng như tốc độ của máy tính. Hơn thế nữa, môi trường làm việc của các lập trình viên bị cô lập với các lập trình viên khác. Điều đó tạo nên sự khó khăn trong việc chia sẻ dữ liệu và chương trình, điều đó giảm năng suất làm việc của những người làm việc trong cùng một dự án hay cùng một mục đích. Hệ điều hành UNIX ra đời đã cung cấp ba tiến bộ chủ yếu so với hệ thống đơn nhiệm cũ là:

- UNIX cho phép nhiều hơn một người có thể sử dụng máy tính hoặc nhiều chương trình cùng xử lý trong cùng một lúc (đa nhiệm).

- UNIX cho phép từng cá nhân có thể thông tin trực tiếp với các máy tính khác thông qua thiết bị đầu cuối .

- Cuối cùng UNIX làm cho sự chia sẻ dữ liệu và chương trình giữa các cá nhân với nhau dễ dàng hơn.

Hệ điều hành UNIX được xây dựng trên bốn phần chính bao gồm:

- Phần lõi (*kernel*)
- Hệ thống tệp (*file system*)
- Phần vỏ (*shell*)
- Các lệnh (*commands*)

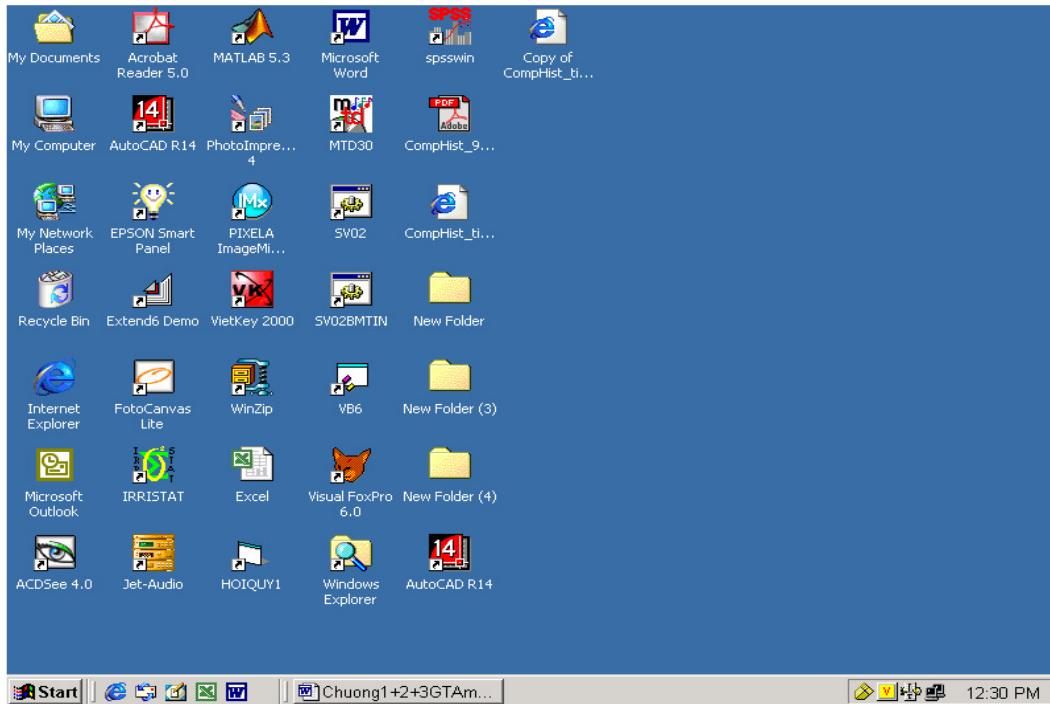
Hiện nay có rất nhiều hệ điều hành UNIX do nhiều hãng phát triển.

## **2 - Hệ điều hành Microsoft windows 98/2000**

### **2.1 - Khởi động máy và các thành phần cơ bản**

#### **a) Khởi động hệ điều hành**

Hệ điều hành được cài đặt trong ổ C: của máy vi tính. Khi mở máy (ấn nút Power ) thì hệ điều hành được nạp vào trong bộ nhớ của máy vi tính, khi nạp xong sẽ hiện ra màn hình chính của Windows ( Desk top) như sau (hình 1.3)



Hình 1.3

- Phía trên là các biểu tượng của các trình ứng dụng. Khi muốn mở 1 trình ứng dụng ta chỉ việc kích đúp chuột vào biểu tượng của nó.

- Dưới cùng là thanh ứng dụng TASKBAR

## b) Thanh ứng dụng TASKBAR

Thanh ứng dụng Taskbar như sau:



Thanh ứng dụng này chứa nút Start và danh sách các ứng dụng đã kích hoạt. Tất cả các cửa sổ ứng dụng khi đưa về chế độ cực tiểu đều xuất hiện trên thanh Taskbar. Khi này, các bạn chỉ cần nhấp đúp chuột lên biểu tượng của nó trên thanh này.

- *Di chuyển Taskbar*

Thanh Taskbar có thể thay đổi ở các vị trí khác nhau trên màn hình. Muốn thay đổi vị trí, các bạn chỉ cần thao tác nháy và kéo lên nền của Taskbar. Tại nơi xuất hiện mũi tên hai chiều, các bạn nháy và kéo lên các biên phía trong để thay đổi kích thước của Taskbar.

- *Các thao tác chính.*

Khi bạn nhấp nút phải chuột lên nền của Taskbar sẽ xuất hiện bảng chọn.

Các mục chọn trong bảng chọn có ý nghĩa như sau:

+ Toolbars: Khi đưa con trỏ đến mục chọn này thì hộp thoại xuất hiện.

Nếu chọn Desktop thì các biểu tượng trên màn hình chính của Windows xuất hiện trên thanh Taskbar.

Nếu chọn New Toolbar thì hộp thoại New Toolbar xuất hiện. Giả sử bạn chọn Printer thì hộp thoại Printer xuất hiện theo dạng rút gọn trên thanh Taskbar.

+ Cascade Windows: Các cửa sổ đang mở xếp lợp lên nhau.

+ Tile Windows Horizontally: Các cửa sổ ứng dụng xếp lợp lên nhau theo chiều ngang.

+ Tile Windows Vertically: Các cửa sổ ứng dụng xếp cạnh nhau theo chiều đứng.

+ Minimize All Windows: Cực tiểu hóa các cửa sổ đang mở và đưa về Taskbar.

+ Properties: Định nghĩa cách thể hiện của Taskbar và thay đổi các thành phần trong Start Menu Programs. Khi chọn mục này, hộp thoại Taskbar Properties xuất hiện. Các mục chọn trong hộp thoại này có các chức năng như sau:

Always on top: Taskbar luôn xuất hiện trên màn hình ở tất cả các ứng dụng.

Auto hide: Taskbar bị che dấu đến khi con trỏ chuột di chuyển đến vị trí của nó.

Show small icon in Start menu: Thay đổi kích thước các biểu tượng trên bảng chọn Start.

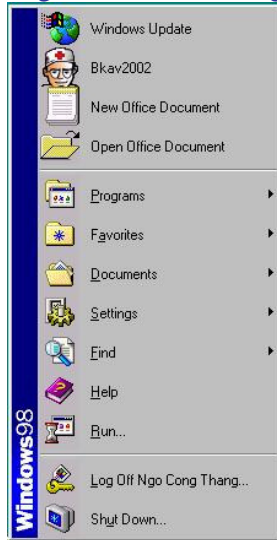
Show Clock: Hiện hay ẩn đồng hồ trên Taskbar.

### c) Nút Start

Nút Start thực hiện nhiều chức năng quan trọng: Thực hiện các ứng dụng, mở tệp, gọi cửa sổ trợ giúp, tìm tệp, thoát khỏi Windows 98, ... và các công cụ thiết lập cấu hình khác.

Khi bạn nháy chuột ở nút Start, trên màn hình xuất hiện bảng chọn Start Menu Programs. Tùy theo việc cài đặt mà trong bảng chọn Start sẽ xuất hiện các mục khác nhau.

Bảng chọn Start có dạng như sau (Hình 2.3)



Hình 2.3

Các mục chọn chính trong bảng chọn này có chức năng như sau:

- *Programs*: Thực hiện các chương trình ứng dụng trên Windows.

- *Documents*: Mở các tài liệu văn bản, đồ họa.

- *Settings*: Thiết lập cấu hình Control Panel, máy in, Taskbar.

- *Find / Search*: Tìm kiếm tệp hoặc thư mục.

- *Help*: Mở cửa sổ trợ giúp.

- *Run*: Chạy các tệp chương trình.

- *Shut Down*: Khi chọn mục này, một mục chọn xuất hiện, bạn có thể chọn một trong các chức năng:

|                  |                                               |
|------------------|-----------------------------------------------|
| <b>Shut down</b> | Tắt máy.                                      |
| <b>Restart</b>   | Khởi động lại hệ điều hành Windows.           |
| <b>Stand by</b>  | Khởi động lại Windows và vào chế độ MS – DOS. |

#### d) Mở trình ứng dụng

Để mở 1 trình ứng dụng ta có thể thực hiện 1 trong 3 cách sau:

Cách 1: Kích đúp chuột vào biểu tượng của trình ứng dụng trên màn hình chính của Windows

Cách 2: Chọn nút **START**/ chọn mục **PROGRAMS**/ chọn trình ứng dụng

Cách 3: Chọn nút **START**/ chọn mục **RUN** / chọn **BROWSE**/ chọn thư mục và trình ứng dụng/ chọn **OK**

#### e) Tắt máy

Chọn nút **START/ SHUTDOWN / SHUTDOWN/ OK**

Sau đó tắt màn hình.

#### f) Khởi động lại

Trong quá trình làm việc máy bị treo (không hoạt động tiếp), để khởi động lại máy ta ấn nút **RESET** ( hoặc ấn đồng thời 3 phím **CTRL, ALT, DEL**).

### 2.2 - Tạo, thay đổi các biểu tượng hoặc mục chọn cho một nội dung

#### a) Tạo biểu tượng hoặc mục chọn cho một chương trình độc lập.

Đối với các chương trình độc lập, chẳng hạn như: Microsoft Excel, Microsoft Word, Foxpro, ta có thể tạo một mục chọn nằm trong một bảng chọn nào đó hoặc tạo một biểu tượng trên màn hình Windows để khi chọn mục chọn hoặc biểu tượng tương ứng thì chương trình thực hiện mà không phải thực hiện các thao tác tìm kiếm phức tạp.

Trình tự thực hiện như sau:

- Nhấn nút Start.
- Chọn mục Settings. Khi này, một bảng chọn xuất hiện.
- Chọn Taskbar. Khi này bảng chọn Taskbar Properties xuất hiện.
- Chọn Start menu Programs, nhấn nút Add... trong khung Customize Start Menu.

Khi này hộp Create Shortcut xuất hiện.

- Nhấn nút Browse...
- Chọn thư mục chứa chương trình.
- Chọn tệp chương trình .
- Chọn Next. Khi này, bảng chọn Select Program Folder xuất hiện.
- Trong khung Select folder to place shortcut in, ta phải chọn vị trí để chứa biểu tượng:
  - + Nếu ta nhấn chuột trên Desktop, biểu tượng sẽ xuất hiện trên màn hình chính của Windows .

+ Nếu chọn Programs, biểu tượng sẽ xuất hiện trong khung Programs.

+ Nếu chọn Start Menu, mục chọn được đặt trong khung Start.

+ Nếu chọn StartUp, chương trình sẽ được thực hiện ngay sau khi khởi động Windows.

+ Nếu chọn New Folder, Windows sẽ tạo một bảng chọn mới trong bảng chọn Programs để chứa biểu tượng của chương trình.

- Sau khi chọn xong, bảng chọn Select a title for the Program xuất hiện.
- Trong khung Select a name for the Shortcut, ta cần đưa tên cho mục chọn này, nếu ta không đưa tên mới vào khung trên thì Windows sẽ đưa tên ngầm định
- Nhấn nút **Finish** kết thúc.

## b) Thay đổi biểu tượng và tên của biểu tượng

\* Thay đổi biểu tượng:

Giả sử cần thay đổi biểu tượng cho chương trình đặt trong bảng chọn Programs, các bước tiến hành như sau:

- Đưa con trỏ đến biểu tượng này
- Nhấn chuột phải. Khi này, hộp thoại Properties xuất hiện.
- Nhấn chuột phải vào nút Change Icon để thay đổi biểu tượng. Khi này, hộp thoại Change Icon xuất hiện:

Chọn biểu tượng cần thiết hoặc có thể nhấn nút Browse để tìm biểu tượng thích hợp trong các thư mục khác. Các tệp là biểu tượng có phần mở rộng là \*.ICO. Cũng cần lưu ý rằng, trong thư mục System của Windows có tệp Shell32.dll là một tệp chứa rất nhiều biểu tượng. Bạn có thể mở tệp này và chọn biểu tượng cần thiết.

Xin lưu ý: Có thể tự tạo biểu tượng bằng cách tạo hoặc sửa một hình ảnh và đặt tên tệp có phần mở rộng là \*.ICO. Khi này, hình ảnh được tạo có thể làm biểu tượng cho một mục chọn nào đó.

\* Thay đổi tên biểu tượng

Để thay đổi tên biểu tượng tiến hành như sau:

- Nếu biểu tượng nằm trên màn hình của Windows (Desktop) thì có thể nhấn chuột vào tên biểu tượng. Khi này, con trỏ nằm ở tên biểu tượng và sửa lại tên này. Cũng có thể tiến hành như dưới đây.

Vào bảng chọn chứa biểu tượng, chọn biểu tượng cần sửa lại tên, nhấn chuột phải. Khi này, một bảng chọn xuất hiện.

Chọn mục Rename. Khi này, hộp thoại xuất hiện. Sửa lại tên biểu tượng (mục chọn) trong khung New name.

Xin lưu ý: Trong bảng chọn còn có các mục sau:

- + Chọn mục Delete sẽ xóa mục chọn (biểu tượng) đã chọn.
- + Chọn open sẽ thực hiện nội dung của mục chọn.
- + Chọn mục Create Shortcut sẽ tạo một biểu tượng cho mục chọn này trên màn hình chính của Windows (Desktop).

## 2.3 - Sử dụng hộp thoại control panel

Hộp thoại Control Panel là hộp thoại rất quan trọng dùng để cài đặt các tham số liên quan đến giao diện của Windows, phần mềm ứng dụng, phần cứng của máy ...

Để mở hộp thoại Control Panel, tiến hành các bước sau:

- Vào bảng chọn Start, chọn mục Settings.
- Chọn Control Panel. Khi này, hộp thoại Control Panel xuất hiện như hình sau ( Hình 3.3)



Hình 3.3

**a) Thiết lập tốc độ của bàn phím và con trỏ màn hình**

Trong hộp thoại Control Panel, chọn biểu tượng:



Hộp thoại xuất hiện:

- Trong khung Repeat rate: Chọn tốc độ của bàn phím bằng cách kéo thanh trượt trong khung này. Tốc độ khi gõ các phím, kéo con trỏ bằng các phím mũi tên và ngay cả khi sử dụng NC, sẽ càng nhanh khi kéo gần đến vị trí Fast và ngược lại, sẽ chậm dần khi kéo thanh trượt đến gần với Slow.
- Trong khung Cursor blink rate: Thay đổi tốc độ nhấp nháy của con trỏ. Khi thanh trượt càng gần đến vị trí cuối (Fast) thì tốc độ nhấp nháy càng nhanh. Ngược lại, khi càng gần đến vị trí đầu (Slow) thì tốc độ càng chậm.

**b) Thiết lập các tham số cho màn hình**

Trong hộp thoại Control Panel, chọn biểu tượng:



Display

Khi này, hộp thoại Display Properties xuất hiện:

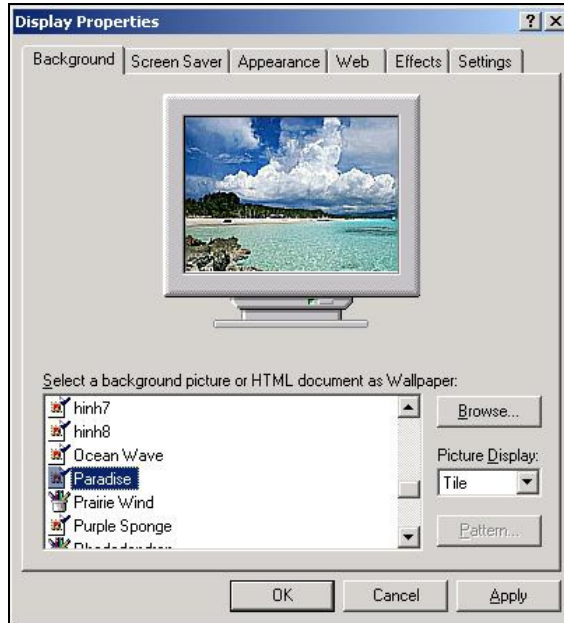
Ta chọn các mục sau:

*\* Mục Background*

Mục này chọn ảnh hoặc màu nền cho màn hình.

Khi chọn khung này, hộp thoại Display Properties có dạng như hình sau.

Trong khung Wallpaper, chọn ảnh hoặc màu nền cho màn hình Windows. Có thể sử dụng nút Browse để chọn ảnh cho màn hình trong thư mục hoặc ổ đĩa khác.



Hình 4.3

*\* Mục Screen Saver*

Mục này chọn hình ảnh xuất hiện khi không sử dụng chuột hay bàn phím và cài đặt mật khẩu.

Khi chọn mục này thì hộp thoại Display properties xuất hiện.

- Trong khung Screen Saver của hộp thoại này chọn một hình ảnh. Hình ảnh này sẽ xuất hiện sau một khoảng thời gian không sử dụng chuột hay bàn phím. Khoảng thời gian này tính bằng phút và được xác định trong khung Wait.

- Khi chọn nút Password protected, chỉ chọn được nút Password protected khi đã chọn một hình ảnh trong khung Screen Saver (nếu chọn mục None thì không chọn được nút này). Nhấn chuột vào nút này, hộp thoại xuất hiện. Khi đưa mật khẩu vào trong khung New password, cần nhắc lại mật khẩu này một lần nữa trong khung Confirm new password. Khi đã thiết lập chế độ này, sau một thời gian xác định trong khung Wait mà ta không sử dụng đến máy tính bằng cách tác động lên chuột hoặc các phím của bàn phím, hình ảnh đã được chọn trong khung Screen Saver sẽ xuất hiện. Nếu ấn một phím hoặc nhấn chuột thì hộp thoại xuất hiện. Ta cần phải đưa vào mật khẩu đã cài đặt để có thể làm việc với máy tính. Nếu mật khẩu đưa vào không đúng, ta không thể tiếp tục công việc.

- Khi cần xóa mật khẩu hoặc sửa đổi lại mật khẩu, bạn vào hộp thoại Change Password thực hiện tương tự như trên.

- Chọn mục Power sẽ xuất hiện:

Khung Turn off monitor, xác định khoảng thời gian mà sau khoảng thời gian này không sử dụng đến máy tính (không nhấn các phím trên bàn phím hoặc nhấn chuột) thì màn hình sẽ tự động tắt.

Tương tự với các thao tác trên đối với ổ cứng khi chọn thời gian trong khung Turn off hard disks.

*\* Mục Appearance*

Khung này để chọn phong chữ, kích thước chữ cho các biểu tượng, các bảng chọn trong các ứng dụng của Windows ...

Khi chọn khung này, hộp thoại Display Properties xuất hiện

- Trong khung Scheme, chọn các cặp màu cho màn hình giao diện của Windows cũng như trong các ứng dụng khác: Microsoft Excel, Microsoft Word ...

- Trong khung Item, chọn phông chữ, kích thước chữ, màu sắc chữ cho các bảng chọn, các dòng thông báo, biểu tượng, ... Khi nhấn chuột lên mũi tên xuống trong khung này, hộp thoại xuất hiện.

+ Chọn đối tượng cần chọn phông chữ, kích thước chữ trong hộp thoại trên. Một số đối tượng như một mục chọn trong bảng chọn, nội dung văn bản trong cửa sổ, ... được miêu tả trong hộp thoại Display Properties và bạn có thể nhấn chuột lên các đối tượng này để chọn thay cho việc chọn chúng trong khung Item.

+ Chọn phông chữ trong khung Font (khung này sẽ trở thành sáng nét khi đã chọn một đối tượng).

+ Chọn kích thước chữ trong khung Size và chọn màu trong khung Color cho đối tượng đã lựa chọn.

#### \* Mục Effects

Hộp thoại này dùng để thay đổi các biểu tượng (Icon) trên giao diện của Windows (Desktop).

Khi chọn khung này, hộp thoại Display Properties xuất hiện.

- Nhấn chuột vào biểu tượng cần thay đổi.

- Nhấn chuột vào biểu tượng **Change Icon**

Khi này, một hộp thoại là danh sách các biểu tượng xuất hiện. Chọn biểu tượng cần thiết.

#### \* Mục Web

Khung này để cài đặt các tham số cho các trang Web khi máy của bạn được nối với Internet.

#### \* Mục Settings

Mục này khai báo số màu của màn hình và độ phân giải của hình.

Khi chọn khung này, hộp thoại Display Properties xuất hiện.

- Trong khung Color palette, chọn số màu cho màn hình. Số màu có thể có được còn phụ thuộc vào Card màn hình mà bạn đang sử dụng.

- Thanh trượt trong khung Desktop area dùng để chọn độ phân giải cho màn hình.

- Nhấn chuột vào khung Advanced Properties để xem các thông tin về Card màn hình và màn hình. Trong trường hợp cần thiết, bạn có thể thay đổi Card màn hình khi chọn nút này.

Cuối cùng chọn OK kết thúc.

Xin lưu ý: Thay cho việc chọn biểu tượng Display trong hộp Control Panel, có thể thao tác như sau: Trên màn hình của Windows, nhấn chuột phải, chọn Properties.

### c) Thay đổi ngày tháng và thời gian cho hệ thống

Trong hộp thoại Control Panel, chọn biểu tượng:



Hoặc nhấn chuột vào vị trí ghi thời gian bên phải dưới đáy màn hình. Khi này, hộp thoại Date/Time Properties xuất hiện.

- Trong khung Date: Chọn ngày, tháng, năm cho hệ thống.



- Trong khung Time: Chọn thời gian. Bạn cần chọn ba lần cho các giá trị: giờ, phút và giây.
  - Trong Time zone: chọn múi giờ.
- Sau khi đã thiết lập xong các tham số, nhấn nút Apply phía dưới hộp thoại để chấp nhận các giá trị.


#### d) Cài đặt máy in

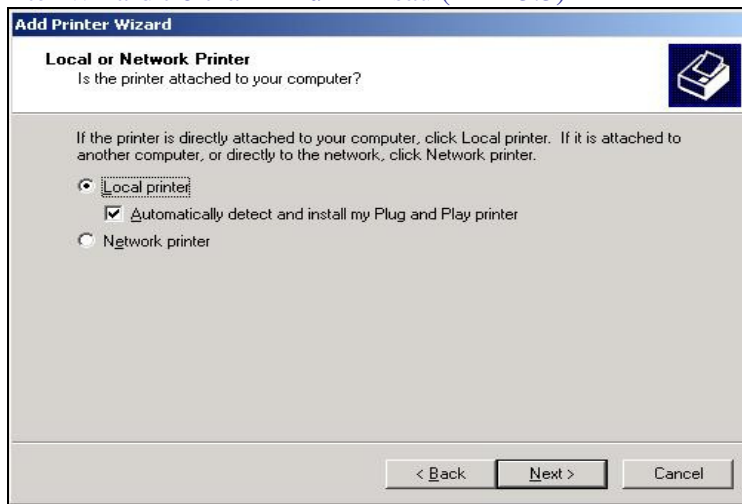
Khi sử dụng một máy in mới, bạn cần cài đặt chương trình điều khiển máy in thì máy in mới sử dụng được. Các bước tiến hành như sau: Trong hộp thoại Control Panel, chọn biểu tượng:



Khi này, hộp thoại Printers xuất hiện. Trong hộp thoại Printers, nhấn chuột lên biểu tượng



Hộp thoại Add Printer Wizard xuất hiện. Nhấn chuột vào nút  hộp thoại Add Printer Wizard trở thành như hình sau (Hình 5.3)

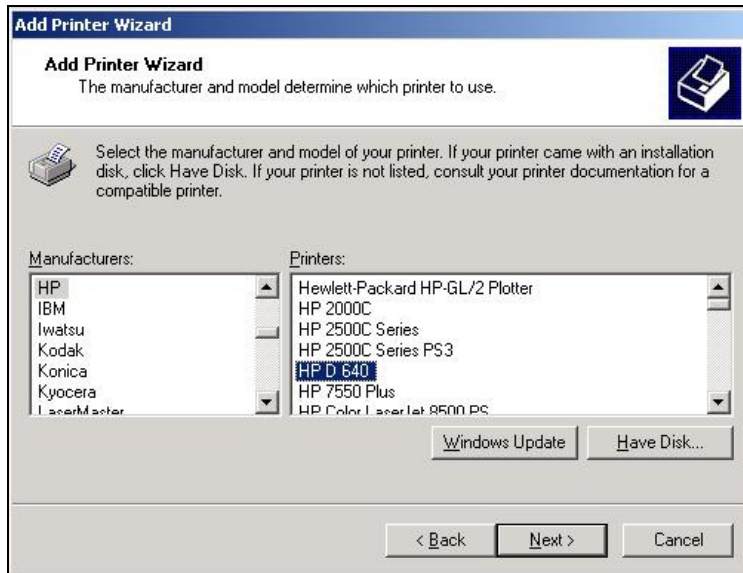


Hình 5.3

Chọn một trong hai phương án:

- Local printer: Cài đặt máy in cục bộ (được nối trực tiếp vào máy tính đang sử dụng).
- Network printer: Máy in cài đặt trên mạng.

Giả sử bạn chọn Local printer. Nhấn chuột vào nút Next. Khi này, hộp thoại Add Printer Wizard trở thành như hình sau (Hình 6.3)



Hình 6.3

Trong khung Manufacturers, chọn nhóm máy in. Khi này, trên cửa sổ bên trái xuất hiện các máy in của nhóm này như hình trên. Chọn máy in cần cài đặt. Nhấn chuột vào nút



Khi này, hộp thoại Install From Disk xuất hiện.

Nhấn chuột vào nút OK để cài đặt chương trình điều khiển máy in.

Khi mua một máy in, nhà cung cấp bao giờ cũng đưa cho bạn một, thậm chí một vài đĩa là chương trình điều khiển máy in. Nếu là đĩa mềm bạn chỉ cần đưa đĩa vào ổ A và nhấn chuột vào nút  . Nếu chương trình điều khiển được đặt trong thư mục hoặc ổ đĩa khác,

bạn nhấn chuột vào nút  khi chọn thư mục hoặc nhấn chuột vào mũi tên xuống trong khung Copy manufacture's files from để chọn ổ đĩa cần thiết. Có một số máy in, Windows tự cài đặt chương trình điều khiển và bạn chỉ cần chọn thư mục Windows (là thư mục cài đặt chương trình Windows) trong hộp thoại trên. Các thế hệ Windows càng về sau thì khả năng này càng lớn. Khi chọn được vị trí chứa chương trình điều khiển, nhấn chuột vào .

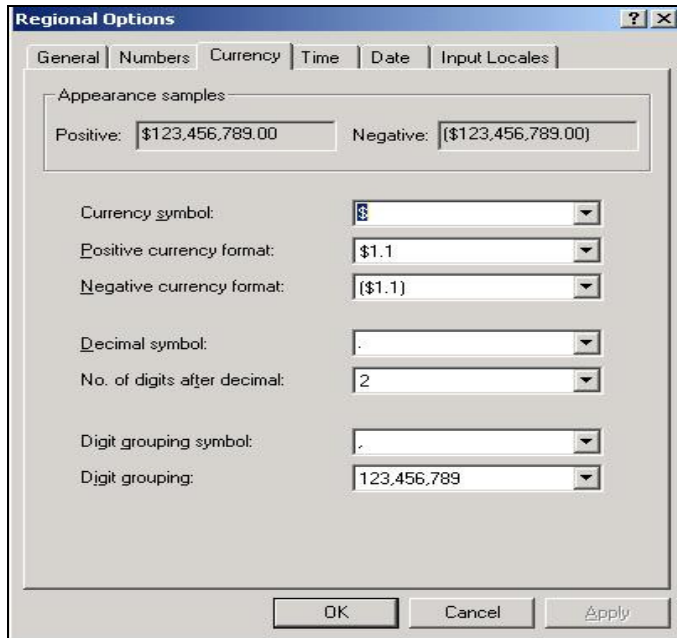
### e) Thiết lập chế độ hiển thị của dữ liệu

Để thiết lập chế độ hiển thị của dữ liệu (dạng số, tiền tệ, thời gian, ...), các bạn tiến hành như sau:

- Trong hộp thoại Control Panel, chọn biểu tượng



Khi này, hộp thoại Reional Settings Properties xuất hiện. Nhấn chuột vào nút Number để chọn chế độ hiển thị cho các dữ liệu dạng số. Hộp thoại Reional Settings Properties có dạng như hình sau:



Hình 7.3

- Trong khung Decimal symbol, chọn ký hiệu biểu thị phân cách phần thập phân. Bạn có thể chọn dấu phẩy (,) và khi này, các ứng dụng của Windows như Microsoft Excel, Microsoft Word... sẽ sử dụng ký hiệu này trong các phép toán, công thức, trình bày kết quả... của dữ liệu kiểu số.
- Trong khung No.of digits after decimal: Xác định số chữ số thập phân.
- Trong khung Digit grouping symbol: Xác định ký hiệu dùng làm dấu phân cách giữa các số hàng nghìn, triệu, tỷ ...
- Trong khung No. of digits in group: Xác định số chữ số tạo thành một nhóm và được phân cách bằng ký hiệu ở mục trên. Thông thường, số lượng này là 3.

Các khung còn lại dùng để thiết lập cách thể hiện số âm. Bạn có thể chọn số âm được viết:

-123.456,78

hoặc

(123.456,78)

bằng cách chọn trong khung Negative number format.

Bạn có thể nhấn chuột lên khung Currency, Time và Date trong hộp thoại Regional Settings Properties để xác lập chế độ hiển thị cho dữ liệu kiểu tiền tệ, thời gian và ngày tháng.

## 2.4 - Sử dụng chương trình Windows Explorer

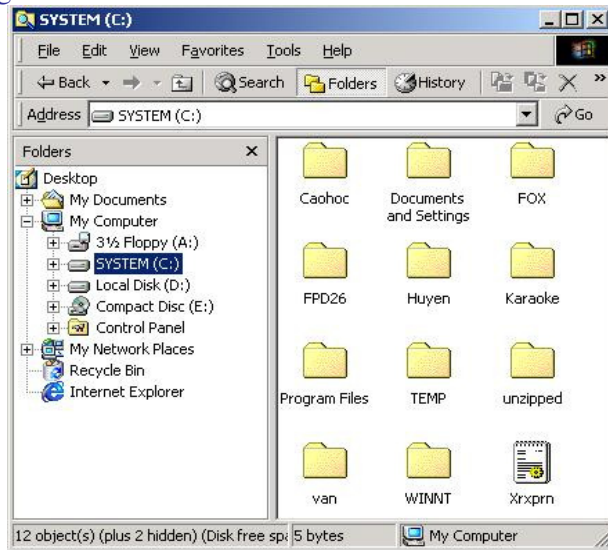
Chương trình Windows Explorer là một chương trình tiện ích nằm trong các bộ Windows 95/98/2000, ... dùng để quản lý đĩa. Bạn có thể sử dụng chương trình này để sao chép, xóa, đổi tên các tệp, tạo các thư mục mới ...

### a) Khởi động chương trình Windows Explorer

Có thể khởi động chương trình Explorer bằng một trong 3 cách sau:

- Cách 1: Kích đúp chuột vào biểu tượng **Windows Explorer** trên Desktop
- Cách 2: Chọn nút **Start** / chọn mục **Program**/ chọn **Windows Explorer**
- Cách 3: Chọn nút **Start** / chọn mục **Run** / chọn nút **Browse** / chọn thư mục và chọn chương trình ứng dụng

Sau khi khởi động bằng một trong ba cách trên, cửa sổ làm việc của Windows Explorer có dạng như hình sau:



Hình 8.3

Cửa sổ làm việc của Windows Explorer được chia thành hai cửa sổ con: Khi đưa con trỏ đến một thư mục hay ổ đĩa ở cửa sổ bên trái thì trên cửa sổ bên phải sẽ xuất hiện nội dung của thư mục hay ổ đĩa này.

### b) Ra khỏi Windows Explorer

Chọn mục FILE/ chọn mục EXIT

### c) Các ứng dụng của Windows Explorer

\* *Tạo thư mục mới*

Để tạo một thư mục mới, các bạn tiến hành các bước như sau:

- Trên cửa sổ bên trái, đưa con trỏ đến ổ đĩa hoặc thư mục cần tạo thư mục mới. Nếu con trỏ nằm trên ổ đĩa thì thư mục được tạo sẽ nằm trực tiếp trên ổ đĩa này. Nếu con trỏ nằm ở thư mục nào thì thư mục mới sẽ nằm trực tiếp trong thư mục này.
- Vào bảng chọn File, chọn New. Khi này, một hộp thoại xuất hiện.
- Chọn Folder. Khi này, trên cửa sổ bên phải xuất hiện một biểu tượng như hình sau:



Con trỏ đang nằm ở New Folder. Đổi tên thành tên thư mục cần tạo. Ví dụ sửa thành



*\* Xem thông tin*

- Giả sử, khi bạn đưa con trỏ vào thư mục **Giaotrinh** ở cửa sổ bên trái thì trên cửa sổ bên phải xuất hiện nội dung của thư mục Giaotrinh.

- Khi ấn vào phím + bên trái thư mục, các thư mục trực thuộc thư mục này (nằm trực tiếp trong thư mục) xuất hiện phía dưới để có thể xem tiếp nội dung của các thư mục con.

- Khi muốn xem thư nội dung của thư mục con trong một thư mục thì đưa con trỏ đến thư mục con này và nhấn chuột. Nội dung sẽ được xuất hiện trên cửa sổ bên phải bao gồm cả tệp và các thư mục con trực thuộc. Trên cửa sổ bên trái, dưới thư mục này xuất hiện danh mục các thư mục nhưng chỉ xuất hiện các thư mục trực thuộc.

- Khi cần quay trở về thư mục mẹ thì nhấn chuột lên dấu trừ ( - ) bên cạnh thư mục, danh mục các thư mục phía dưới của thư mục trên cửa sổ bên trái biến mất.

*\* Sao chép*

Giả sử cần sao chép thư mục **Excel** trong thư mục **Giaotrinh** sang thư mục **Sach**, có thể tiến hành theo hai cách như sau:

• Cách 1:

- Đưa con trỏ đến thư mục Excel, nhấn chuột phải. Khi này, một hộp hội thoại xuất hiện

- Chọn Copy.

- Nhấn chuột vào thư mục Sach, nhấn chuột phải, chọn Paste.

• Cách 2:

- Trên cửa sổ bên trái, đưa con trỏ đến thư mục Giaotrinh. Khi này, trên cửa sổ bên phải xuất hiện nội dung của thư mục Giaotrinh.

- Nhấn chuột vào thư mục Excel. Khi này, thư mục đổi màu. Chú ý: Nếu bạn nhấn đúp chuột thì sẽ xuất hiện nội dung của thư mục này trên cửa sổ bên phải.

- Vào bảng chọn Edit, chọn Copy, nhấn chuột trên thư mục Giaotrinh, vào bảng chọn Edit, chọn Paste hoặc nhấn Ctrl+V.

*Xin các bạn lưu ý*

Để chọn một nhóm các thư mục hoặc tệp để sao chép hoặc dịch chuyển, trước khi chọn tệp hoặc thư mục mới, nhấn và giữ phím Ctrl.

*\* Dịch chuyển*

Dịch chuyển một thư mục, tệp hoặc một nhóm thư mục hoặc tệp được thực hiện tương tự như khi sao chép chỉ khác là khi sao chép thì chọn Copy trong các bảng chọn còn khi dịch chuyển thì thay bằng mục chọn Cut trong các bảng chọn.

Cũng có thể tiến hành bằng cách kéo chuột như sau:

- Chọn thư mục cần dịch chuyển (có thể ở cửa sổ bên phải hoặc bên trái).

- Nhấn và giữ chuột, kéo thư mục đến thư mục cần dịch chuyển và thả chuột (khi này, thư mục hoặc ổ đĩa cần dịch chuyển đến đổi màu).

*\* Xóa tệp hoặc thư mục*

Nhấn chuột vào thư mục hoặc tệp cần xóa và thực hiện một trong ba cách sau:

- Nhấn chuột phải, chọn mục Delete hoặc chọn Menu Edit/ Delete .

- Nhấn chuột vào biểu tượng Delete

- Nhấn phím Delete trên bàn phím.

Khi thực hiện một trong ba cách trên, hộp thoại Confirm Folder Delete xuất hiện, chọn Yes nếu muốn xóa, chọn No nếu muốn hủy bỏ lệnh.

*\* Đổi tên*

Nhấn chuột vào tệp hoặc thư mục cần đổi tên, nhấn chuột phải chọn Rename.

*\* Mở tệp hoặc thư mục*

- Chọn tệp hoặc thư mục cần mở. Nhấn chuột phải. Khi này, một hộp thoại xuất hiện.

- Chọn Open.

+ Nếu là thư mục thì tạo một cửa sổ để chứa nội dung của thư mục.

+ Nếu là tệp chương trình thì thực hiện chương trình này.

+ Nếu là tệp ứng dụng được tạo ra từ các chương trình hệ thống như Microsoft Excel, Microsoft Word ... thì khởi động chương trình này đồng thời mở tệp được chỉ định.

*\* Xem thông tin của tệp hoặc thư mục*

Chọn tệp hoặc thư mục cần xem thông tin, nhấn chuột phải. Khi này, một hộp thoại xuất hiện. Chọn Properties.

*\* Tìm kiếm thông tin*

Tìm kiếm thông tin trong Windows 98 và Windows 2000 có khác nhau.

**- Tìm kiếm trong Windows 98:**

Để tìm kiếm thông tin, tiến hành các bước như sau:

+ Vào bảng chọn Tools, chọn mục Find. Khi này, một bảng chọn phụ xuất hiện.

+ Chọn mục Find Files and Folder. Khi này hộp thoại Find xuất hiện.

+ Chọn mục Name & Location: Trong khung Named, chọn tên tệp cần tìm. Bạn có thể sử dụng nhiều ký hiệu dấu sao (\*) cho một nhóm ký tự bất kỳ và dấu hỏi (?) cho một ký tự bất kỳ.

+ Trong khung Containing text, đưa vào đoạn văn bản có chứa trong tệp cần tìm.

+ Trong khung Look in, đưa tên ổ đĩa hoặc thư mục cần tìm kiếm.

+ Nhấn nút Find Now khi đã đưa các thông tin trên. Bạn có thể tìm theo một chỉ tiêu về tên tệp hoặc đoạn văn bản chứa trong tệp (khi này, chỉ tiêu kia để trống) hoặc kết hợp hai chỉ tiêu tìm kiếm nêu trên.

+ Trong hộp hội thoại Find, nhấn chuột vào khung Date để chọn điều kiện tìm kiếm theo ngày tháng. Một hộp thoại xuất hiện.

+ Chọn nút Find all files. Nhấn chuột vào mũi tên xuống trong khung này để chọn chế độ tìm kiếm theo thời gian sửa đổi hay thời gian tạo tệp.

Bạn chỉ có thể chọn một trong ba nút sau:

+ between ... and ... : Giữa một khoảng thời gian.

+ during the previous ... month(s): Trước một số tháng.

+ during the previous ... day(s): Trước một số ngày.

Sau khi khai báo xong các tham số trên, nhấn chuột lên nút Find Now.

+ Trong hộp thoại Find, nhấn chuột lên khung Advanced để chọn chế độ tìm kiếm theo kích thước tệp.

**- Tìm kiếm trong Windows 2000:**

Để tìm kiếm thông tin trên đĩa, các bạn tiến hành các bước sau:

+ Nhấn chuột vào biểu tượng Search trên thanh công cụ. Khi này hộp thoại Search xuất hiện.

+ Đưa tên tệp hoặc thư mục cần tìm kiếm vào khung Search for files or folders named.

+ Đưa đoạn văn bản có trong tệp tìm kiếm

+ Chọn thư mục hoặc ổ đĩa cần tìm trong khung Look in.

+ Khi đã đưa các tham số cần thiết, nhấn chuột vào nút Search now để bắt đầu tìm kiếm.

Trong khung Search Results xuất hiện kết quả các thư mục hoặc các tệp thỏa mãn điều kiện tìm kiếm cùng các tham số của chúng.

\* *Format đĩa:*

Giả sử cần format đĩa mềm (ổ A), đưa vệt sáng đến ổ A trên cửa sổ bên trái, nhấn nút chuột phải. Khi này, một hộp thoại xuất hiện. Chọn mục Format.

### **Câu hỏi ôn tập chương III**

1. Nêu khái niệm hệ điều hành, các chức năng cơ bản của hệ điều hành.
2. Nêu sự khác nhau giữa hệ điều hành máy tính cá nhân và hệ điều hành mạng. Nêu một số hệ điều hành đang được sử dụng rộng rãi hiện nay ở nước ta.
3. Nêu cách khởi động, mở trình ứng dụng, khởi động lại, tắt máy trong hệ điều hành Windows 2000.
4. Trình bày về màn hình chính của Windows 2000, thanh ứng dụng TASKBAR, nút START.
5. Chương trình Windows Explorer để làm gì ?. Nêu một số ứng dụng của Windows Explorer.

## CHƯƠNG IV: HỆ SOẠN THẢO VĂN BẢN MICROSOFT WORD

Một trong các ứng dụng thông dụng nhất của máy vi tính là để soạn thảo văn bản. Hệ soạn thảo văn bản là bộ chương trình giúp cho người sử dụng soạn thảo văn bản trên máy vi tính. Hệ soạn thảo văn bản được sử dụng rộng rãi nhất hiện nay là hệ soạn thảo văn bản Microsoft Word. Trong chương này giới thiệu về hệ soạn thảo văn bản Microsoft Word. Nội dung chính đề cập tới các vấn đề: Khởi động Word, các thành phần trên màn hình word, soạn thảo văn bản, định dạng dữ liệu, các lệnh về tệp, kẻ bảng biểu, các thao tác thường gặp trong Word.

### 1 - Giới thiệu chương trình Ms-Word

#### 1.1 - Giới thiệu chung

MS-Word là trình soạn thảo văn bản do hãng MicroSoft thiết kế. MS-Word hoạt động trong môi trường Window. Hệ điều hành WINDOWS có giao diện đồ họa giữa người sử dụng và máy vi tính, mọi lệnh của WINDOWS được thể hiện qua các biểu tượng đồ họa trên màn hình và việc ra lệnh cho máy tính chỉ là chọn các biểu tượng hoặc thực đơn phù hợp thông qua bàn phím hoặc thiết bị chuột.

Các máy tính ở nước ta hiện nay phổ biến dùng hệ điều hành Window 98 hoặc Window 2000.

Chương trình MS-WORD là hệ soạn thảo văn bản nằm trong bộ chương trình Tin học văn phòng (MicroSoft Office), chúng ta có thể gặp các phiên bản khác nhau của Word như Word6.0 trong Office 4.3 hoặc Word 97; Word 2000. Những phiên bản mới này được bổ sung thêm một số tính năng về đồ họa, bảng biểu mà các phiên bản trước không có.

Trong môi trường WINDOWS có thể tìm thấy bảng tính điện tử Excel, hệ quản trị dữ liệu Access, công cụ vẽ Powerpoint và một số trình ứng dụng khác. Trong phần này chúng ta chỉ đề cập đến MS-WORD.

Bộ chương trình Windows được cài đặt trong ổ cứng của máy vi tính. Với phiên bản 3.1 (là phiên bản làm việc trong môi trường DOS) sau khi khởi động từ đầu nhắc hệ thống C:\> ta chỉ cần gõ lệnh WIN và bấm tiếp phím Enter là chương trình được khởi động. Các phiên bản Windows 95 trở đi việc khởi động được hoàn toàn tự động.

#### 1.2 - Cách sử dụng chuột

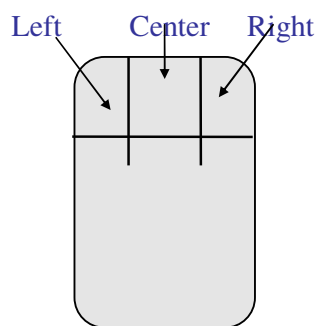
Trước khi làm việc với MS-WORD ta cần phải biết cách sử dụng thiết bị chuột (Mouse). Mouse có thể có 2 hoặc 3 phím (Hình 1.4). Khi di chuyển mouse trên bàn làm việc con trỏ mouse trên màn hình sẽ di chuyển theo, tùy thuộc vào vị trí của Mouse trên màn hình mà hình dạng của nó sẽ thay đổi. Phần lớn các thao tác của mouse chỉ sử dụng phím trái (Left) do vậy trong tài liệu này nếu nói bấm mouse thì có nghĩa là bấm phím trái, những trường hợp bấm phím phải sẽ có ghi chú riêng.

Có 3 thao tác bấm phím của mouse:

- **Bấm đơn:** đưa mouse đến biểu tượng hoặc vị trí cần thiết rồi bấm phím trái
- **Bấm kép:** đưa mouse đến biểu tượng hoặc vị trí cần thiết rồi bấm phím trái hai lần liên tiếp
- **Bấm rê:** đưa mouse đến vị trí cần thiết bấm và giữ phím trái sau đó di mouse trên bàn, thao tác bấm rê nhằm bôi đen một vùng văn bản hoặc di chuyển một đối tượng từ chỗ này sang chỗ khác.

Hình dạng con trỏ chuột phụ thuộc vào vị trí của nó trên màn hình, mỗi hình dạng có một công dụng khác nhau, chúng ta sẽ thấy trong từng ứng dụng cụ thể.





Hình 1.4

Một số chuột hiện nay phím giữa được thay bằng một bánh xe. Quay bánh xe này màn hình sẽ được cuộn lên hay xuống.

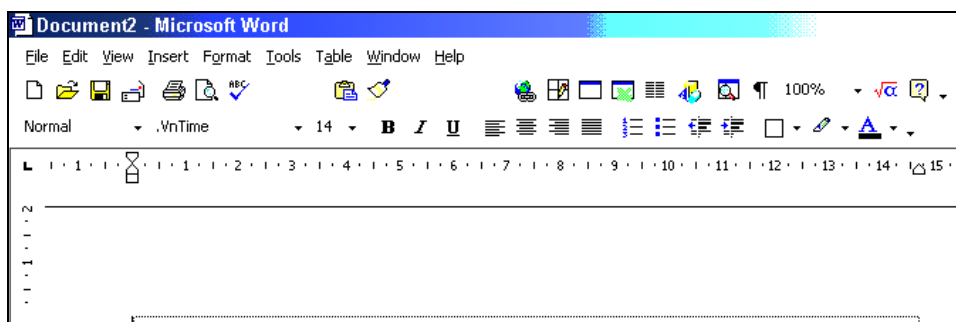
### 1.3 - Khởi động và ra khỏi Word

#### a) Khởi động Word

Có thể khởi động Word theo 3 cách sau:

- Cách 1: Kích đúp chuột vào biểu tượng **Word** trên Desktop
- Cách 2: Chọn nút **Start** / chọn mục **Program**/ chọn **Microsoft Word**
- Cách 3: Chọn nút **Start** / chọn mục **Run** / chọn nút **Browse** / chọn thư mục và chọn chương trình ứng dụng.

Sau khi khởi động bằng một trong ba cách trên, cửa sổ làm việc của Microsoft Word có dạng như hình sau (Hình 2.4)



Hình 2.4

Theo thứ tự từ trên xuống dưới thì màn hình bao gồm các phần sau:

- Thanh tiêu đề: Thanh này cho biết tên cửa sổ đang làm việc (Microsoft Word) và tên văn bản đang soạn thảo.
- Thanh thực đơn (Menu): Thanh này trình bày các thực đơn ngang, mỗi mục chọn trong thực đơn ngang sẽ cho tiếp một thực đơn dọc.
- Thanh công cụ (Toolbars): Trên thanh công cụ là các nút (Button), các nút này là công cụ giao tiếp thay cho việc chọn các thực đơn con trong thanh Menu.

- Thanh định dạng (Format): Thanh này gồm các nút phục vụ cho việc định dạng văn bản, các chức năng định dạng văn bản thực ra cũng đã có trong thực đơn Format, tuy nhiên định dạng bằng thanh Format sẽ nhanh hơn và tiện dụng hơn.
  - Thước (Ruler): Chỉ ra lề trái, lề phải, vị trí các cột trong bảng.
  - Phần hiện ra văn bản: Gồm nhiều dòng, để hiện ra văn bản mà ta gõ vào
  - Cuối cùng là thanh trạng thái: Chỉ ra các thông tin hiện thời về tệp văn bản đang soạn thảo: Trang hiện thời (Page), đoạn (Sec), dòng (Ln), cột (Col), ...
- Có thể dùng chức năng Tools - Customize để thay đổi các nút trong các thanh công cụ hoặc dùng View-Toolbars... để thêm bớt các thanh.

## b) Ra khỏi Word

Chọn mục FILE/ chọn mục EXIT hoặc nhấn đồng thời hai phím Alt và F4

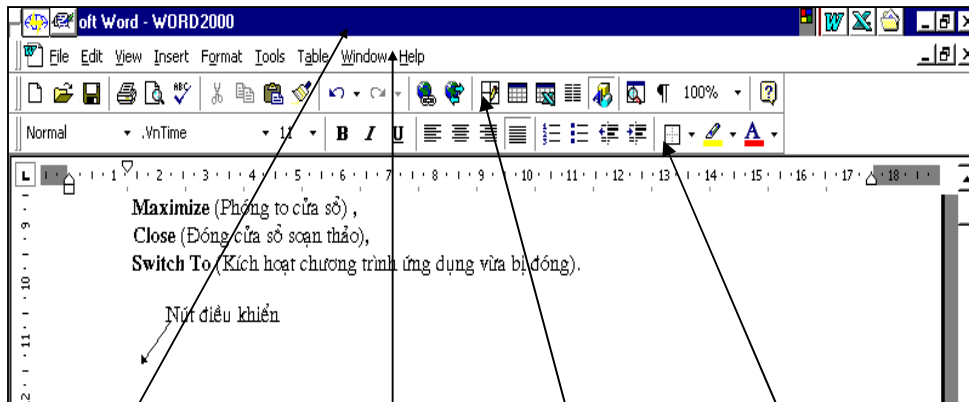
## 1.4 - Mô tả các thành phần trong màn hình Word

### a) Thanh tiêu đề (Title Bar)

Phần giữa thanh tiêu đề là tên của cửa sổ đang mở ( Microsoft Word) và tên tệp văn bản đang soạn thảo, khi bắt đầu làm việc với MS-WORD 6.0 tên tệp văn bản được đặt ngầm định là Document1, nếu chúng ta ghi văn bản vào đĩa với tên khác thì tên đó sẽ thay thế cho Document1. Bên trái thanh tiêu đề là nút điều khiển ( Ctrl menu box). Khi bấm mouse vào nút này ta thấy hiện lên một menu dọc gồm các Menu con:

- Restore** (Khôi phục cửa sổ về trạng thái trước),
- Move** (Dịch chuyển cửa sổ đến vị trí mới),
- Size** (Thay đổi kích thước cửa sổ),
- Minimize** (Thu cửa sổ thành biểu tượng),
- Maximize** (Phóng to cửa sổ),
- Close** (Đóng cửa sổ soạn thảo),
- Switch To** (Kích hoạt chương trình ứng dụng vừa bị đóng).

Nút điều khiển



Thanh tiêu đề    Thanh thực đơn    Thanh công cụ    Thanh định dạng

**Hình 3.4**

Phía bên phải thanh tiêu đề có ba nút

- Thu nhỏ màn hình làm việc thành một dòng

Thu nhỏ hoặc   phóng to cửa sổ

- × Kết thúc làm việc, đóng cửa sổ lại

## b) Thanh thực đơn (Menu Bar)

Thanh thực đơn gồm một danh mục các thực đơn chính xếp theo hàng ngang. Để chọn các ứng dụng trong một Menu ngang ta bấm đơn phím trái mouse vào menu đó hoặc bấm tổ hợp phím nóng phím ( đề phím ALT rồi gõ tiếp chữ cái gạch chân của menu ). Mỗi menu ngang lại gồm nhiều mục dọc (Menu popup), để chọn một mục dọc ta làm hoàn toàn tương tự như chọn menu ngang. Các menu ngang và mục dọc tương ứng được liệt kê dưới đây:

\* **File** (các lệnh về xử lí tệp):

**New** (mở cửa sổ mới để soạn thảo văn bản )  
**Open** (mở văn bản đang lưu trong đĩa ra màn hình)  
**Save** (cất văn bản đang soạn thảo vào tệp trên đĩa)  
**Save as** (cất văn bản vào tệp với tên khác)  
**Close** (đóng văn bản đang soạn thảo)  
**Find file** (tìm tệp trên đĩa)  
**Summary Information** (tạo thông tin tóm tắt về văn bản)  
**Templates** (Chọn khuôn mẫu trình bày văn bản)  
**Page Setup** (định dạng trang văn bản , chọn cỡ giấy, lề in ...)  
**Print Preview** ( xem toàn cảnh văn bản trước khi in )  
**Print** (in văn bản).

\* **Edit** (các lệnh phục vụ việc soạn thảo):

**Undo** (bỏ thao tác vừa làm)  
**Repeat** (lặp lại thao tác vừa làm)  
**Cut** (xoá đối tượng đã chọn hoặc đoạn văn bản đã bôi đen)  
**Copy** (chép đối tượng đã chọn hoặc đoạn văn bản đã bôi đen vào vùng đệm clipboard)  
**Paste** (dán đối tượng đã có trong vùng đệm vào vị trí mới)  
**Paste Special** (dán đối tượng theo một phương pháp đặc biệt, chuyển cột thành hàng hoặc hàng thành cột, nâng cao mật độ khi in)  
**Clear** (xoá đoạn văn bản đã chọn)  
**Find** (tìm kiếm từ ngữ hoặc đoạn văn trong văn bản)  
**Replace** (tìm và thay thế)  
**Goto** (nhảy tới trang số..)  
**AutoText** (Tạo một đoạn văn bản mẫu để có thể gọi ra bất kỳ lúc nào)  
**Bookmark** (đánh dấu định vị trong văn bản).

\* **View** (chọn chế độ hiển thị văn bản trên màn hình):

**Normal** (bình thường)  
**Layout** (hiện lề ngoài văn bản hoặc hiện văn bản kèm theo các hình vẽ)  
**PageLayout** (hiện lề ngoài trang văn bản)  
**Master Document** (tạo văn bản chính với các văn bản con)  
**Fulscreen** (mở cửa sổ văn bản rộng kín toàn màn hình)  
**Toolbars..**(Cho hiện hoặc không hiện các thanh công cụ, thanh định dạng, thanh đường viền...)  
**Ruler** (cho hiện hoặc không hiện thước kẻ trên đầu cửa sổ văn bản)  
**Header and Footer** (cho hiện tiêu đề đầu và cuối trang)  
**Footnote** (cho hiện chú giải cuối trang)

**Annotation** (Cho hiện chú thích trong văn bản)  
**Zoom** (Thay đổi kích thước cửa sổ soạn thảo)

\* **Insert** ( chèn các đối tượng khác nhau vào văn bản):

**Break** ( chèn dấu ngắt trang)

**Page Number** (đánh số trang)

**Annotation** (thêm lời chú thích)

**Date and Time** (Ngày và giờ)

**Field** (chèn mã trường có chứa thông tin xác định vào văn bản),

**Symbol** (chèn các kí tự đặc biệt như  $\alpha, \beta$ ... hoặc các dấu hoa văn )

**Form Field** (biểu mẫu định sẵn)

**Footnote** (chú giải cuối trang)

**Caption** (chèn thêm chú giải cho nội dung đã chọn)

**Cross reference** (chèn các đối tượng qua tham khảo chéo trong hộp liệt kê)

**Index and Table** (chèn bảng mẫu)

**File** (chèn tệp)

**Frame** (chèn khung để đóng gói văn bản hoặc một bức tranh)

**Picture** (chèn thêm bức tranh đã có sẵn trong thư viện vào văn bản hoặc vào khung)

**Objects** (chèn các đối tượng khác như bảng tính, các dấu toán học ..)

**Database** (chèn cơ sở dữ liệu)

\* **Format** ( chức năng định dạng văn bản):

**Font** :định dạng chữ bao gồm kiểu chữ: (Font); kích thước: (Font size); màu sắc: (Color) ; dáng chữ: (Style)

**Paragraph** (định dạng đoạn văn bản: khoảng cách dòng, độ thụt dòng, lề, khoảng cách giữa các đoạn văn bản)

**Tabs** (định dạng bước nhảy cột khi gõ phím Tab)

**Borders and Shading** (định dạng đường bao và bóng của khung, bảng)

**Columns** (phân chia đoạn văn bản hoặc trang giấy thành các cột ...)

**Change-Case** (biến chữ to thành nhỏ và ngược lại, định dạng nhóm từ kiểu tên riêng...)

**Drop cap** ...(định dạng chữ cái đầu tiên của một đoạn văn bản)

**Bullets and Numbering** (định dạng nét gạch và đánh số đầu mục)

**Heading Numbering** (đánh số các tiêu đề)

**Autoformat** (thiết lập chế độ định dạng tự động)

**Style gallery** (định dạng theo mẫu đã được thiết kế của Windows )

**Styles** (lựa chọn hoặc tổ chức kiểu trình bày có sẵn cho đoạn văn hoặc cho kí tự)

**Frame** (định dạng khung bao)

**Picture** (định dạng lại các hình ảnh đã chèn vào văn bản)

**Drawing object** (định dạng các đối tượng vẽ )

\* **Tools** (Các công cụ trợ giúp)

**Spelling** (kiểm tra lỗi chính tả tiếng anh)

**Grammar** (chỉnh lí văn phạm)

**Thesaurus** (tìm từ đồng nghĩa)

**Hyphenation** (đặt gạch nối các từ một cách tự động hay bằng tay)

**Language** (Chọn ngôn ngữ viết văn bản)

**Word count** (đếm số chữ , số từ , số dòng , số đoạn có trong đoạn văn đã lựa chọn)

**Autocorrect** (thiết lập tên cho một cụm từ hoặc tiêu đề mà nội dung của nó được tự động chỉnh lỗi chính tả)

**Mail Merge** (trộn dữ liệu vào văn bản chính, dữ liệu có thể tạo ra trong Word hoặc các phần mềm khác như Foxpro, Lotus, Foxbase..., nhằm mục đích tạo ra một văn bản mới)

**Envelopes and labels** (tạo bao thư cho văn bản)

**Protect Document** (bảo vệ tài liệu)

**Revisions** (hiệu đính)

**Macro** (tạo lập các vĩ lệnh, có thể hiểu Macro là một tập hợp tất cả các lệnh được gõ từ bàn phím trừ các thao tác chuột)

**Customize** (tạo các nút công cụ hoặc menu riêng)

**Options** (các phương án lựa chọn tổng hợp trình bày màn hình hoặc văn bản)

\* **Table** (Thực đơn về tạo lập bảng):

**Insert table** (chèn thêm một bảng vào vị trí con trỏ)

**Delete cells** (xoá các ô đã bôi đen)

**Merge cells** (liên thông các ô đứng gần nhau theo hàng ngang thành một ô)

**Split cells** (phân chia một nhóm ô thành nhiều ô)

**Select row** (chọn hàng)

**Select column** (chọn cột)

**Select table** (chọn bảng)

**Table Autoformat** (tự động định dạng bảng theo khuôn có sẵn)

**Cell Height and Width** (định kích thước của ô hoặc hàng, cột)

**Headings** (nhập tiêu đề cho cột)

**Convert Text to Table** (chuyển văn bản dưới dạng bảng thành bảng dữ liệu. Điều kiện cần là các dữ liệu phải ngăn cách nhau bằng dấu phẩy .

**Sort Text** (Xếp bảng dữ liệu theo cột số hoặc cột chữ)

**Formula** (tính toán với các số liệu có trong bảng)

**Split table** (phân chia bảng thành 2 bảng)

**Gridlines** (cho hiện hoặc không cho hiện lưới kẻ bảng)

\* **Window** (các lệnh xử lý của sổ văn bản):

**New** (mở cửa sổ văn bản mới)

**Arrange all** (cho hiện đồng thời trên màn hình cửa sổ của các văn bản đang mở)

**Split** chia cửa sổ hiện hành thành hai phần

\* **Help** (xem hướng dẫn các lệnh)

#### c) Thanh công cụ (Tools Bar) (hình 4.4)



Hình 4.4

Các nút trên thanh công cụ tính từ trái qua phải gồm:

**New:** mở một màn hình mới che lên màn hình đang làm việc

**Open:** mở một văn bản đang lưu trữ trong đĩa của máy

**Save:** ghi văn bản hiện có trên màn hình vào đĩa

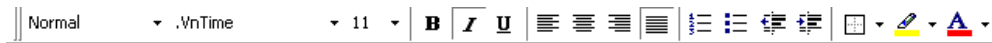
**Print:** In văn bản ra giấy

**Print Preview:** quan sát toàn cảnh văn bản trước khi in  
**Spelling and Grammar:** dò lỗi chính tả và ngữ pháp tiếng Anh  
**Cut:** xoá phần văn bản đang được bôi đen trên màn hình  
**Copy:** sao chép phần đang bôi đen vào Clipad (vùng đệm bộ nhớ)  
**Paste:** dán những gì đang có trong Clipad vào vị trí hiện thời trên màn hình  
**Format Painter:** sao chép định dạng của một đoạn văn bản  
**Undo Typing:** bỏ thao tác vừa thực hiện (khôi phục lại tình trạng trước đó)  
**Redo cut:** khôi phục lại những gì vừa bị xoá  
**Insert Hyperlink:** thêm siêu liên kết  
**Web Toolbar:** hiện thanh công cụ kết nối Internet  
**Tables and Border:** tạo bảng biểu bằng bút vẽ  
**Insert Table:** tạo bảng biểu  
**Insert Microsoft Excel Worksheet:** chèn vào văn bản một trang tính Excel  
**Columns:** chia văn bản thành nhiều cột  
**Drawing:** hiện thanh công cụ vẽ  
**Document Map:** hiện sơ đồ các đề mục của văn bản  
**Show/Hide:** hiện hoặc tắt ký hiệu xuống dòng  
**Zoom:** Thay đổi tỷ lệ kích thước màn hình  
**Office Assistant:** trợ giúp

Thanh công cụ chứa các biểu tượng dưới dạng các nút giúp lựa chọn nhanh một chức năng nào đó. Thanh công cụ chuẩn của Office 97 hoặc Office 2000 (Standard Tools bar) gồm các nút như trong hình 3.

Các nút trong thanh công cụ được sử dụng bằng cách đưa chuột đến nút rồi bấm đơn, thao tác này cũng tương tự như chọn một chức năng trong thanh thực đơn rồi chọn chức năng con có tên tương ứng. Ví dụ để ghi văn bản đang có trên màn hình vào đĩa ta chỉ việc bấm chuột vào nút Save thay vì phải chọn thực đơn File rồi thực đơn con Save trong Menu File.

#### d) Thanh định dạng (Formating Bar) ( Hình 5.4 )



**Hình 5.4**

Thanh này gồm có các nút dùng để định dạng văn bản, tính từ trái qua phải bao gồm:

**Style** (Kiểu trình bày)  
**Font** (chọn kiểu chữ)  
**Font size** ( chọn cỡ chữ)  
**Bold** (chữ đậm)  
**Italic** (chữ nghiêng)  
**Under Line** (gạch chân)  
**Align Left** (dóng thẳng lề trái)  
**Align Right** (dóng thẳng lề phải)  
**Justify** (dãn đều 2 lề )  
**Numbering** (đánh số thứ tự các đoạn văn bản)  
**Bullets** (Dùng các biểu tượng đánh dấu đoạn văn bản)  
**Decrease Indent** (dịch chuyển cả đoạn văn bản sang trái)  
**Increase Indent** (dịch chuyển cả đoạn văn bản sang phải)  
**Border** (định dạng đường viền khung)  
**Highlight** (chọn màu nền văn bản)  
**Font Color** (chọn màu chữ văn bản)

## 2 - Các phím gõ tắt (ShortCut-Key )

Các phím gõ tắt thường là một tổ hợp phím nhằm định của Windows hoặc các phím do người sử dụng cài đặt để thực hiện nhanh một công việc nào đó thông qua việc gõ bằng bàn phím . Các phím gõ tắt bao giờ cũng bắt đầu bằng một trong các phím chức năng phụ Ctrl, Shift, Alt kết hợp với một hoặc hai phím khác. Dưới đây là một bảng liệt kê các phím gõ tắt có sẵn của Windows và chức năng tương ứng của nó. Cách bấm phím gõ tắt như sau: Đè phím thứ nhất rồi bấm tiếp phím thứ hai hoặc đè phím thứ nhất, đè tiếp phím thứ hai rồi tiếp phím thứ 3 sau đó buông tất cả ra.

Ví dụ: khi ghi tổ hợp phím Alt + F thì có nghĩa là đè phím Alt sau đó gõ phím F, còn nếu ghi Ctrl+Shift+W thì có nghĩa là đè hai phím Ctrl và Shift sau đó gõ phím W rồi buông tay ra.

### 2.1 - Các phím gõ tắt bắt đầu bằng phím Alt

|       |                                      |
|-------|--------------------------------------|
| Alt+F | Chọn thực đơn File trên thanh Menu   |
| Alt+E | Chọn thực đơn Edit trên thanh Menu   |
| Alt+V | Chọn thực đơn View trên thanh Menu   |
| Alt+I | Chọn thực đơn Insert trên thanh Menu |
| Alt+O | Chọn thực đơn Format trên thanh Menu |
| Alt+T | Chọn thực đơn Tools trên thanh Menu  |
| Alt+A | Chọn thực đơn Table trên thanh Menu  |
| Alt+W | Chọn thực đơn Window trên thanh Menu |
| Alt+H | Chọn thực đơn Help trên thanh Menu   |
| Alt+N | Chọn thực đơn Font trên thanh Menu   |

### 2.2 - Các phím gõ tắt bắt đầu bằng phím Ctrl

- Bôi đen đoạn văn bản sau đó bấm:

|               |                                                       |
|---------------|-------------------------------------------------------|
| Ctrl+C        | Sao chép đoạn văn bản đã bôi đen vào Clipboard        |
| Ctrl+V        | Dán đoạn văn bản đã có trong Clipboard vào vị trí mới |
| Ctrl+B        | In đậm, chữ béo                                       |
| Ctrl+I        | In nghiêng                                            |
| Ctrl+U        | Chữ có gạch chân                                      |
| Ctrl+Shift+W  | Gạch chân từng từ                                     |
| Ctrl+Shift+D  | Gạch chân bằng nét kép                                |
| Ctrl+=        | Viết chỉ số dưới ( $x_1, m_2, \dots$ )                |
| Ctrl+Shift+=  | Viết chỉ số trên, số mũ ( $a^2, b^{\sin x}, \dots$ )  |
| Ctrl+ Shift+K | Chữ in hoa nhỏ                                        |
| Ctrl+ Shift+A | Tắt cả chữ in hoa                                     |
| Ctrl+ Shift+H | Cho ẩn văn bản                                        |
| Ctrl+ Shift+C | Copy định dạng                                        |
| Ctrl+ Shift+V | Dán kiểu định dạng                                    |
| Ctrl+Spacebar | Loại bỏ định dạng                                     |

### 2.3 - Các phím gõ tắt bắt đầu bằng phím Shift

|              |                                                        |
|--------------|--------------------------------------------------------|
| Shift+End    | Bôi đen đoạn văn bản từ vị trí hiện thời đến cuối dòng |
| Shift+Home   | Bôi đen đoạn văn bản từ vị trí hiện thời đến đầu dòng  |
| Shift+ ↓     | Bôi đen một dòng văn bản                               |
| Shift+ ← (→) | Bôi đen một ký tự bên trái (hoặc bên phải con trỏ)     |

### 3 - Các thao tác định dạng

#### 3.1 - Phương pháp viết tiếng Việt

Hiện có hai phương pháp viết tiếng Việt đang được sử dụng là phương pháp viết kiểu TELEX và phương pháp viết kiểu Đánh máy. Trong phạm vi chương trình chúng ta chỉ học phương pháp TELEX. Bộ phong chữ tiếng Việt sử dụng trong chương trình này là bộ phong VietKey, hiện nay ở nước ta còn tồn tại nhiều phong chữ như ABC, Freecode, Vietwear, Vni,... Với các máy kết nối Internet thì phải dùng bộ phong Unicode.

##### Cách gõ chữ Việt theo kiểu Telex:

Kiểu Telex là cách gõ tiếng Việt thuận tiện và dễ nhớ, đảm bảo có được tiếng Việt đúng chính tả, đồng thời bỏ đi những cách đặt dấu sai đã tồn tại thành thói quen.

Cách gõ các ký tự đặc biệt của tiếng Việt như sau:

| Gõ vào | Nhận được | Gõ vào | Nhận được |
|--------|-----------|--------|-----------|
| aw     | ã         | Aw     | Ă         |
| ow     | ơ         | Ow     | Ơ         |
| uw     | ư         | Uw     | Ư         |
| aa     | â         | AA     | Â         |
| oo     | ô         | OO     | Ô         |
| ee     | ê         | EE     | Ê         |
| dd     | đ         | DD     | Đ         |

Gõ dấu: **f: huyền ; s: sắc ; r: hỏi ; x: ngã ; j: nặng**

Ví dụ : để có dòng chữ " Trường Đại học Nông nghiệp I" ta phải gõ như sau:

**Truwowngf DDaij hocj Noong nghiiepj I**

\* **Lưu ý :** - Dấu phải gõ vào cuối chữ , nghĩa là viết xong chữ rồi mới đánh dấu.

Ví dụ: Truwowngf -- Trường, DDieenj -- Điện, baos -- báo

- Phím xoá dấu : khi muốn xoá dấu đã đánh trên một từ ta đưa con trỏ đến sát ký tự cuối cùng của từ và gõ phím z.

- Chữ ư và ơ hay đi liền nhau nên chúng còn được bố trí ở phím ] và [, đây là hai phím liền nhau trên bàn phím kiểu QWERTY, vì vậy gõ sẽ nhanh hơn.

] -- ư      [ -- ơ      } -- Ư      { -- Ơ

Thí dụ : tr][ngf -- Trường    dd][cj -- được

- Gõ ooo nhận được oo như coongs -- cóng

- Nếu bạn muốn gõ dấu [ và ] thì bạn chỉ việc gõ 2 lần phím đó.

Khi gõ Telex, Vietkey sẽ tự động bỏ dấu đúng chính tả tiếng Việt. Nếu bạn gõ nhầm dấu, bạn cứ việc gõ dấu mới vào mà không phải mất công xoá dấu cũ đi.

Cũng cần lưu ý thêm rằng khi gõ sai dấu trên một từ và chúng ta đã chuyển sang gõ từ khác thì không quay về sửa dấu được mà phải xoá hết nguyên âm của từ sai rồi mới gõ lại từ đó. Để tránh điều phiền toái này bạn có thể chọn tính năng sửa dấu nhanh của Vietkey. Cách thức chọn như sau: Bấm đơn phím phải chuột vào biểu tượng Vietkey, trên màn hình xuất hiện hộp thoại (hình 6.4). Chọn chức năng "Sửa dấu nhanh" bằng cách bấm đơn vào chức năng này. Kể từ nay khi quay lại từ đánh dấu sai ta chỉ việc gõ lại dấu đúng.



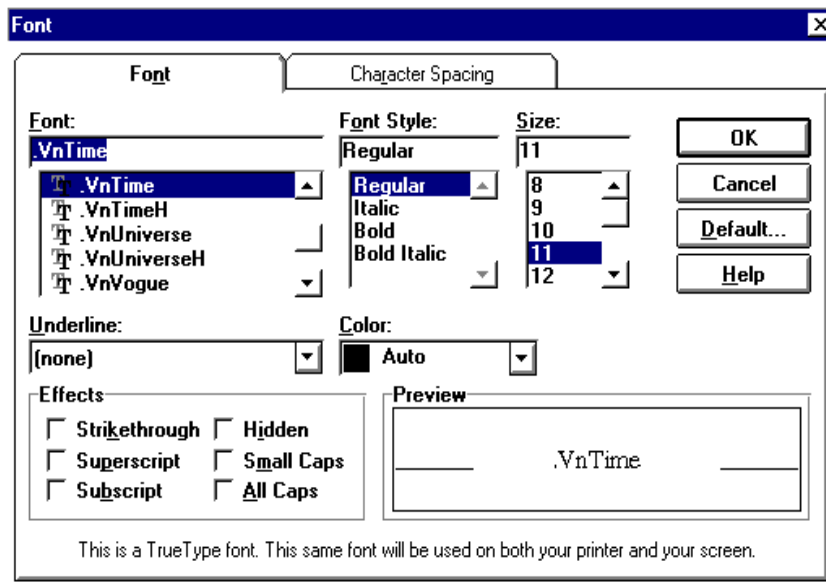


Hình 6.4

### 3.2 - Định dạng ký tự

#### a) Sử dụng thanh thực đơn

- Bôi đen đoạn văn bản muốn định dạng
- Chọn chức năng **Format**
- Trong Format chọn **Font**, hộp thoại Font xuất hiện ( Hình 7.4). Trong hộp này có hai mục chọn là **Font** (kiểu chữ) và **Character Spacing** ( vị trí chữ ). Trong mục chọn Font có các nút định dạng như sau:
  - + **Font**: kiểu chữ, ngay phía dưới là các kiểu chữ có thể lựa chọn. Với bộ phông chữ tiếng Việt chuẩn Vietkey (phông bắt đầu bằng hai ký tự Vn) nếu tận cùng là chữ H thì là phông chữ in.
  - + **Font Style**: dáng chữ, có các dáng Regular, Bold, Italic, Bold Italic. Chọn một trong các dáng này và quan sát khung Preview để xem hiệu quả.
  - + **Size**: kích thước chữ, có thể chọn kích thước từ 8 đến 72, ngầm định là 12.
  - + **Underline**: chữ có gạch chân
  - + **Color**: chọn màu cho chữ





Hình 7.4

- + **Effect**: Một số thao tác định dạng khác:

- **Strikethrough**: tạo nét gạch xuyên qua từ
  - **Superscript**: đưa chữ lên cao, tạo số mũ
  - **Subscript**: đưa chữ xuống dưới, tạo chỉ số dưới
  - **Hidden**: cho ẩn chữ, muốn cho hiện lại bấm tổ hợp phím Ctrl+Z
  - **Small Caps**: Biến chữ hoa thành chữ thường
  - **All Caps**: Biến tất cả thành chữ hoa
- + **Preview**: quan sát hiệu quả định dạng
- Trong mục chọn Character Spacing có các nút định dạng như sau:
- + **Spacing**: cách thức hiện chữ trên dòng, có 3 khả năng lựa chọn:
- **Normal**: hiện bình thường
  - **Expanded**: các ký tự viết giãn cách nhau
  - **Condensed**: các ký tự viết sát vào nhau
- + **Position**: vị trí của ký tự trên dòng
- **Normal**: hiện bình thường
  - **Raised**: đưa chữ lên cao hơn
  - **Lowered**: đưa chữ xuống thấp hơn


Sau khi lựa chọn xong các đặc tính cần thiết cho chữ bấm chuột vào OK hoặc bấm phím Enter để quay về màn hình soạn thảo văn bản.

### b) Sử dụng thanh Format

Trong trường hợp chỉ cần định dạng kiểu chữ, kích thước chữ, và các dáng đậm, nghiêng, gạch chân ta có thể dùng các nút trên thanh định dạng. Phương pháp tiến hành như sau: bôi đen đoạn văn bản cần định dạng, sau đó muốn có chữ đậm chỉ việc bấm chuột vào nút **B**, cần chữ nghiêng bấm nút **I**, cần chữ gạch chân bấm nút **U**, muốn đưa đoạn văn bản vào giữa trang giấy bấm tiếp nút . còn muốn văn bản căn đều hai bên mép giấy thì bấm nút ...

#### Lưu ý:

\* Việc chọn màu cho ký tự trong trường hợp a chỉ có ý nghĩa khi quan sát trên màn hình, chứ không thể in ra giấy (trừ trường hợp chúng ta có máy in màu).

\* Văn bản hiện có trên màn hình và văn bản khi in ra giấy có thể không giống nhau. Muốn chắc chắn văn bản in ra ta phải chọn chức năng quan sát toàn cảnh Print Preview tức là chọn nút . Những gì quan sát được qua Print Preview thì khi in ra giấy sẽ giữ nguyên 100%.



### c) Sao chép định dạng

Sau khi đã làm các thủ tục định dạng một đoạn văn bản ta có thể sao chép sự định dạng này sang các đoạn văn bản khác, thao tác như sau:

- Bôi đen đoạn văn bản đã định dạng

- Chọn nút Format Painter , di chuột vào phần màn hình chứa văn bản khi đó chuột sẽ có thêm một chiếc chổi bên trái.

- Bấm và rê chuột trên đoạn văn bản cần định dạng sau đó buông tay ra, đoạn văn bản sẽ được định dạng theo cách của đoạn ban đầu.

Nếu muốn sao chép định dạng cho nhiều đoạn văn bản cùng một lúc, ta bấm kép vào nút Format Painter  rồi lặp lại thao tác như trong bước trên, khi nào sao chép xong bấm đơn lần nữa vào nút Format Painter .

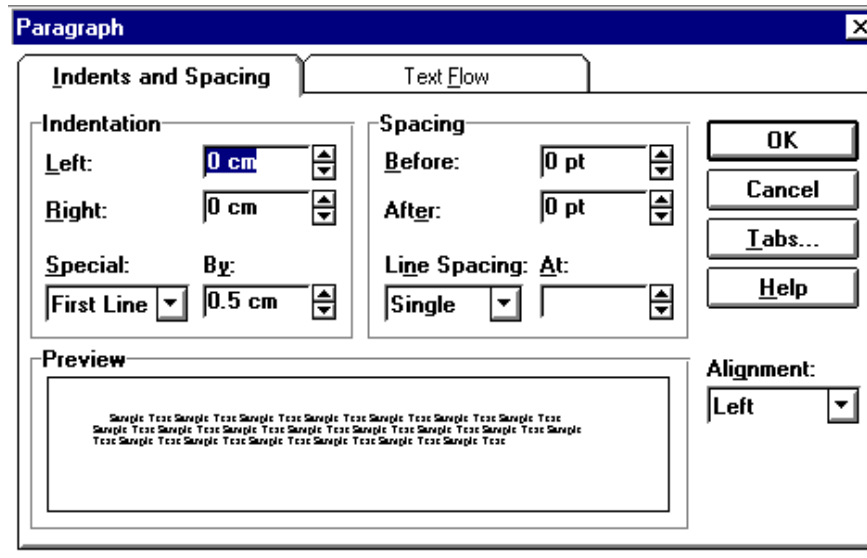
### 3.3 Định dạng đoạn văn bản

Đoạn văn bản (Paragraph) được hiểu là toàn bộ những từ nằm giữa hai lần bấm phím Enter.

Thao tác định dạng đoạn văn bản có thể làm trước hoặc sau khi viết văn bản. Nếu ta đã viết xong văn bản rồi mới định dạng thì phải bấm Ctrl+A để bôi đen toàn bộ văn bản.

Các bước tiến hành:

- \* Bôi đen văn bản hoặc một số đoạn muốn định dạng
- \* Chọn nút Format - chọn tiếp Paragraph, hộp thoại Paragraph xuất hiện ( Hình 8.4)



Hình 8.4

+ **Indents and Spacing**: vị trí đoạn văn bản

+ **Text Flow**: sự liên kết các đoạn

Trong mục Indents and Spacing có các thông số lựa chọn sau:

**Left**: Khoảng cách từ đoạn văn bản đến lề trái

**Right**: Khoảng cách từ đoạn văn bản đến lề phải

**Before**: Đoạn hiện thời cách đoạn trên bao nhiêu

**After**: Đoạn hiện thời cách đoạn dưới bao nhiêu

**Special**: trong Special có hai chức năng

**First line**: Vị trí của dòng đầu tiên viết lùi vào bao nhiêu

**Hanging**: Các dòng còn lại viết lùi vào bao nhiêu

**Line Spacing**: Khoảng cách giữa các dòng, bao gồm các khả năng

**Single**: cách nhau bình thường theo kích thước chuẩn

**1.5 Line**: các dòng cách nhau gấp rưỡi

**Double**: các dòng cách nhau gấp đôi

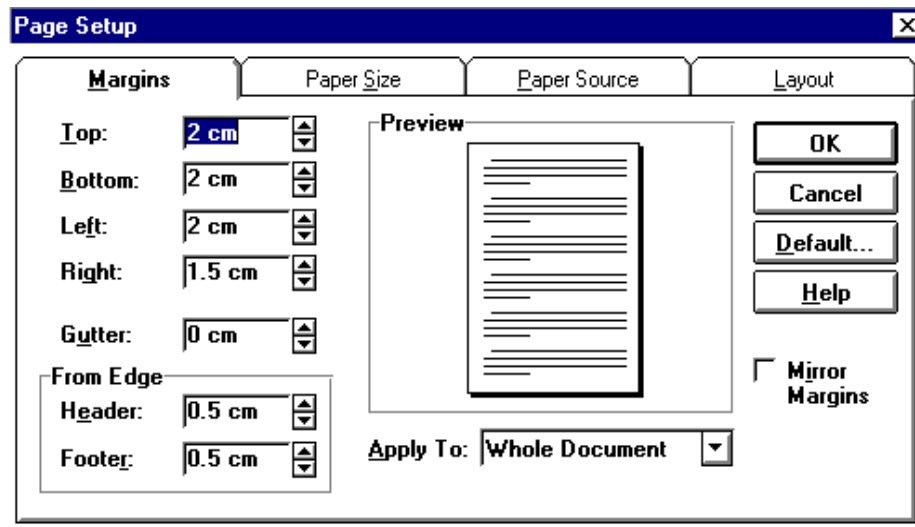
**At least** hoặc **Exactly**: các dòng cách nhau theo lựa chọn của người sử dụng

**Multiple**: Cách nhau gấp 3 ( chữ thường bị mất dấu )

### 3.4 - Định dạng trang giấy

Định dạng trang giấy có thể làm trước hoặc sau khi viết văn bản, nói chung nên làm trước. Thao tác định dạng trang giấy bao gồm:

Chọn chức năng File chọn tiếp Page Setup. Hộp thoại Page Setup hiện lên ( Hình 9.4) với các khai báo:



Hình 9.4

**Margins:** lề trang giấy, trong Margins có:

**Top:** khoảng cách từ mép giấy phía trên đến vị trí dòng đầu tiên

**Bottom:** khoảng cách từ mép giấy phía dưới đến vị trí dòng cuối cùng

**Left:** lề bên trái

**Right:** lề bên phải

**Gutter:** khoảng cách giữa các cột nếu trang giấy chia thành nhiều cột

**From Edge:** khoảng cách tính từ mép giấy, bao gồm:

**Header:** khoảng cách từ mép giấy phía trên đến vị trí viết tiêu đề hoặc số trang

**Footer:** khoảng cách từ mép giấy phía dưới đến vị trí viết dòng chú thích ở đáy trang

**Paper Size :** kích thước trang giấy

Trong mục chọn này có thể chọn các cỡ giấy chuẩn : A4, A3, Letter .. hoặc tự định kích thước trong các mục

**Width:** chiều rộng

**Height:** chiều cao trang giấy

**Orientation:** định hướng in :

**Portrait:** in văn bản theo chiều dọc trang giấy

**Landscape:** in văn bản theo chiều ngang trang giấy

**Paper Source:** Nguồn lắp giấy, thông thường nên chọn giấy rời lắp bằng tay nghĩa là chọn Manual Feed.

## 4 - Kẻ bảng biểu

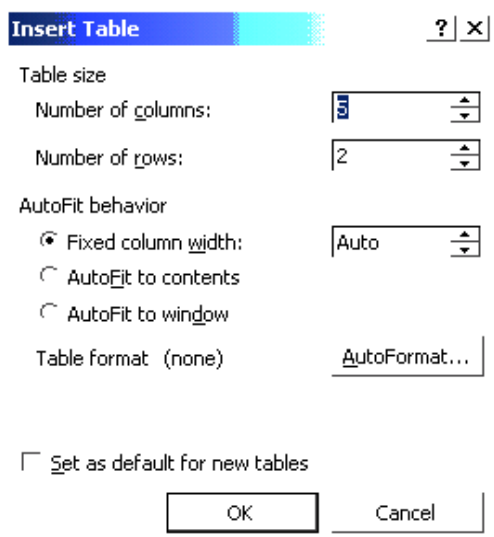
### 4.1 - Chèn bảng biểu vào văn bản

Có hai phương pháp đưa vào văn bản một bảng biểu, hoặc là dùng chức năng **Table** trên thanh thực đơn hoặc là dùng nút **Insert Table** trên thanh công cụ.

Đưa con trỏ đến vị trí cần chèn bảng biểu, bấm đơn vào chức năng Table rồi chọn tiếp Insert - Table hộp thoại Insert Table hiện lên như hình 10.4

Mục **Number of Columns** cho phép chọn số cột của bảng biểu, bấm vào các mũi tên phía bên phải con số ngầm định (số 2) để tăng hoặc giảm số cột.

Mục **Number of Rows** cho phép chọn số hàng của bảng biểu, bấm vào các mũi tên phía bên phải con số ngầm định (số 2) để tăng hoặc giảm số hàng.



Hình 10.4

Mục **Fixed Columns Widths** cho phép chọn độ rộng của các cột trong bảng biểu tính theo Inch hoặc cm, nếu để ngầm định Auto thì MS-WORD sẽ tự động chọn các cột có độ rộng bằng nhau và bằng chiều rộng trang giấy chia cho số cột đã chọn.

Mục **AutoFormat** dùng để chọn dạng bảng biểu đã thiết kế sẵn trong Word. MS-WORD 2000 đã thiết kế 42 kiểu bảng biểu gọi là 42 style. Các style này chủ yếu khác nhau về các dòng tiêu đề phía đỉnh của bảng biểu. Ví dụ bảng biểu trong hình 11.4 là dạng chuẩn có tên là Grid 5.

|  |  |  |  |  |
|--|--|--|--|--|
|  |  |  |  |  |
|  |  |  |  |  |

Hình 11.4

Bảng biểu đã lựa chọn trong cửa sổ Insert Table sẽ hiện lên tại vị trí con trỏ hiện thời trong văn bản, đó mới là bảng biểu tượng trưng, khi in ra giấy sẽ không có các đường kẻ. Muốn hiện các đường kẻ ta phải bôi đen toàn bộ bảng và chọn chức năng **Format - Borders and Shading**. Hộp thoại Table Borders and shading hiện lên có các phần sau:

Phần **Borders** (đường bao) có một số lựa chọn:

**None:** không tạo các đường kẻ

**Box:** chỉ tạo đường bao xung quanh bảng

**Shadow:** tạo bóng đen phía dưới và bên phải bảng

**3-D:** hiện bảng dưới dạng hộp không gian ba chiều

**Custom:** tự chọn các nét vẽ bảng

Để chọn một nét vẽ nào đó cho bảng, ta chọn nét trước ở mục Style, màu sắc nét vẽ chọn ở mục Color, độ dày mảnh của nét vẽ chọn ở mục Width, sau khi đã chọn xong thì bấm chuột vào nét vẽ tương trưng ở mục Preview rồi chọn OK.

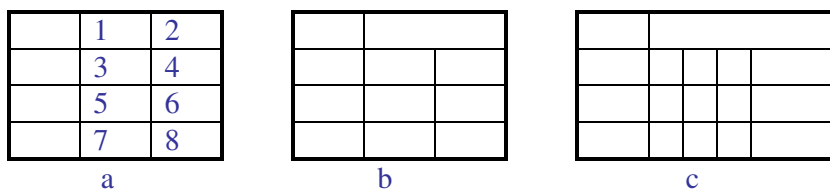
**Page Border:** Tạo đường viền cho trang văn bản

**Shading:** chức năng shading dùng để tạo hoa văn nền cho vùng ô đã bôi đen trong bảng biểu, đồng thời với việc chọn hoa văn còn có thể chọn màu cho hoa văn song với các máy in thông dụng hiện nay ta chỉ có thể in đen trắng. Trong cửa sổ Table Borders and shading ta thấy có nút **Show Toolbar**. Nút này dùng để hiện lên thanh công cụ Borders. Sử dụng các nút trên thanh Borders ta cũng có thể tạo nên các đường viền theo ý muốn.

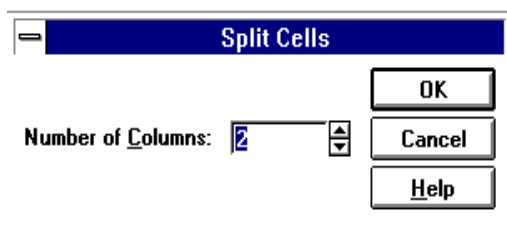
#### 4.2 - Một số thao tác sửa chữa bảng biểu

Trong một số trường hợp cần tạo nên các bảng biểu phức tạp theo yêu cầu ví dụ chia một số ô thành các ô nhỏ hơn hoặc nối thông một số ô với nhau. Ta có thể làm việc này thông qua các chức năng có trong thực đơn Table.

Hình 12.4. a là bảng biểu ban đầu. Để có hình 12.4.b ta bôi đen hai ô 1 và 2 sau đó chọn chức năng **Table - Merge cells**. Để có hình 12.4.c ta bôi đen các ô 3,4,5, sau đó chọn **Table - Split cells** hộp thoại split cells xuất hiện (Hình 13.4). Mục chọn **Number of Columns** cho phép chia các ô đã bôi đen thành số ô tùy ý. Bấm vào mũi tên lên hoặc xuống để chọn số ô mà ta định chia ra, cụ thể ở đây là chia 3.



Hình 12.4



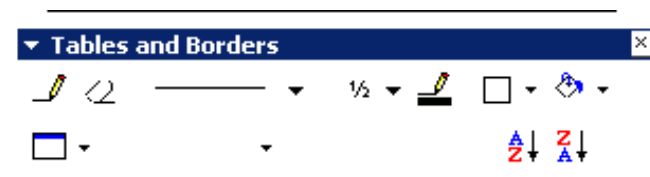
Hình 13.4

Với hệ soạn thảo Word 2000 trên thanh công cụ có thể thấy một nút vẽ bảng tên nút này là Tables and Borders 

Bấm đơn để chọn nút này trên màn hình xuất hiện một thanh công cụ vẽ bảng và chuột có hình dạng bút chì, dùng bút chì chúng ta có thể vẽ một khung đồng thời vẽ thêm các nét ngang dọc nghĩa là thêm vào trong bảng các cột hoặc hàng.

Trường hợp vẽ sai bấm đơn vào viên tẩy chuột sẽ biến thành viên tẩy. Bấm rê chuột trên một nét vẽ nào đó chúng ta sẽ xóa được nét vẽ đó.

Chú ý: Những nét vẽ tạo nên đường bao của bảng khi xóa sẽ để lại nét mờ, khi in ra sẽ không có.

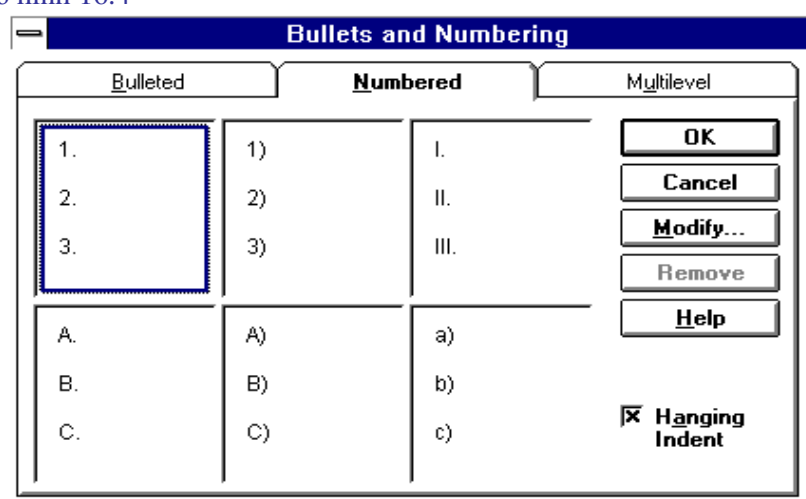


Hình 14.4

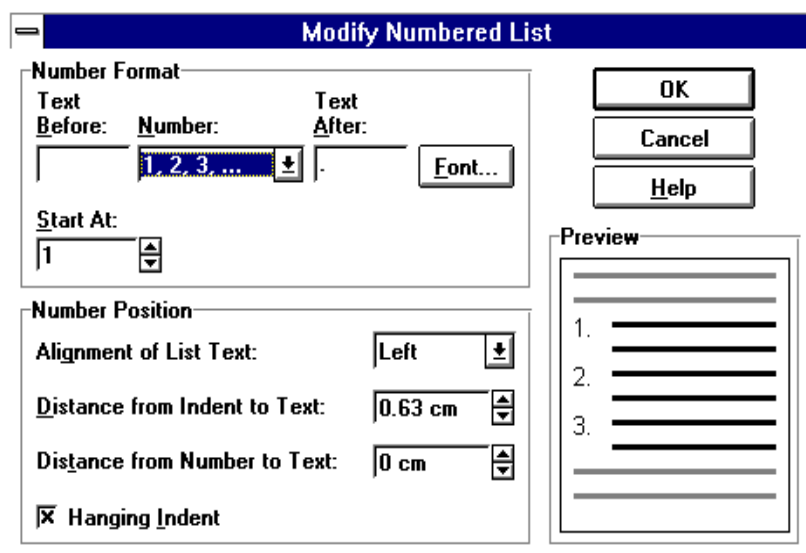
### 4.3 - Đánh số thứ tự trong một cột

Giả sử cần đánh số thứ tự từ 1 đến 50 trong 50 ô của một cột nào đó

- Bôi đen số ô muốn đánh số thứ tự
- Chọn Format - chọn Bullets and Numbering xuất hiện cửa sổ hình 15.4
- Chọn Numbered trong cửa sổ, chọn tiếp một dạng số mà ta muốn điền vào trong cột, ví dụ dạng đầu bên phải các số sẽ có dấu chấm. Để bỏ dấu chấm đó hãy chọn Modify ta có cửa sổ hình 16.4



Hình 15.4



Hình 16.4

Dưới mục Text After máy để ngầm định dấu chấm, hãy xoá dấu chấm đó đi rồi chọn OK.

#### 4.4 - Tạo bảng bằng bút chì

Sử dụng nút Tables and Borders hoặc chọn Table - Draw Table trên thực đơn ta có một thanh công cụ vẽ bảng, hình 14.4.

Lúc này chuột biến thành một chiếc bút chì, bằng cách bấm rê chuột ta có thể vẽ nên một bảng tùy ý, nếu nét vẽ sai có thể dùng chiếc tẩy trên thanh công cụ này để tẩy nét vẽ đi, cách thức tiến hành là:

Bấm đơn chuột vào viên tẩy, chuột sẽ có hình dạng tẩy, bấm rê chuột trên nét vẽ sai rồi buông tay ra nét vẽ sẽ biến mất.

### 5 - các thao tác thường gặp

#### 5.1 - Viết chữ cái đầu tiên

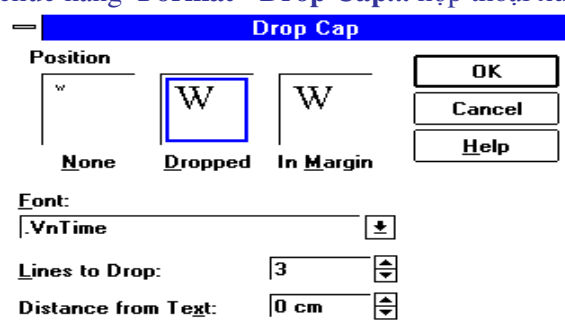
Chữ cái đầu tiên của một đoạn có thể viết to hơn các chữ khác , thậm chí có thể viết trên hai, ba dòng

Ví dụ:

Bông lau trắng giữa rừng xanh  
Mong manh trước gió , mong manh trước đời  
Mùa lá đổ, mùa mưa rơi  
Phất phơ lau trắng ven đồi sương buông

Các thao tác:

- \* Đặt con trỏ màn hình vào đầu dòng muốn điều chỉnh chữ
- \* Chọn chức năng **Format - Drop Cap...** hộp thoại xuất hiện như hình 17.4



Hình 17.4

Các khả năng lựa chọn:

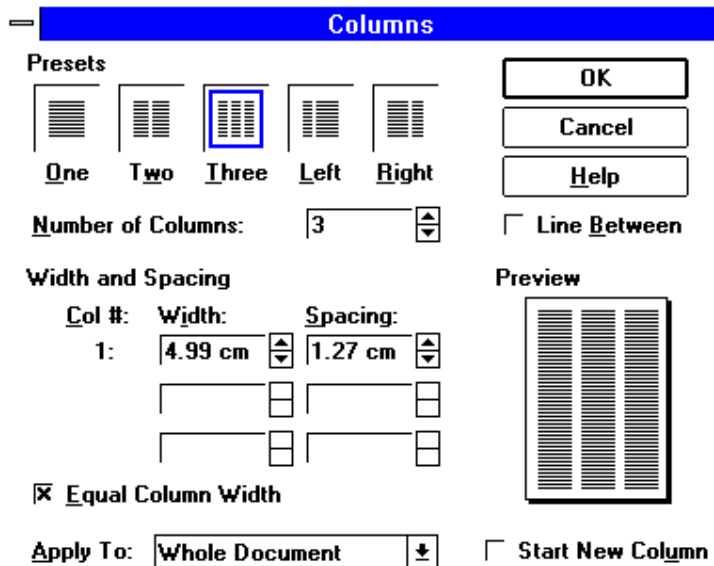
- None:** Chữ viết bình thường , không viết to
- Dropped:** Viết chữ to nằm gọn trong khuôn khổ văn bản
- In Margin:** chữ viết to nằm ngoài lề văn bản
- Font:** Chọn phong chữ cho chữ viết to
- Lines to Drop:** Chữ viết trên bao nhiêu dòng
- Distance from Text:** khoảng cách giữa chữ và văn bản

#### 5.2 - Chia trang hoặc đoạn văn bản thành nhiều cột

Chọn **Format - Columns...** Hộp thoại xuất hiện như trong hình 18.4 các mục chọn:

- One:** Trang giấy chỉ có 1 cột
- Two:** Trang giấy chia thành 2 cột
- Three:** Trang giấy chia thành 3 cột
- Left:** Trang giấy chia thành 2 cột , cột bên trái có kích thước nhỏ hơn cột phải





Hình 18.4

**Right:** Trang giấy chia thành 2 cột , cột bên trái có kích thước lớn hơn cột phải

**Numbers of Columns:** Số lượng cột cần có

**Width and Spacing:**

**Width:** Độ rộng của một cột

**Spacing:** Khoảng cách giữa các cột

Cần chú ý rằng sau khi chọn kích thước giấy và lề trang giấy thì MS-WORD sẽ tự động xác định độ rộng của các cột tùy thuộc vào số lượng cột và khoảng cách giữa các cột mà ta đã chọn.

### 5.3 - Tạo thuật ngữ viết tắt

Nhiều thuật ngữ đặc biệt là tiếng La tinh dùng trong y học hoặc những tên riêng chúng ta hay gặp khi soạn thảo có thể viết tắt theo phương pháp:

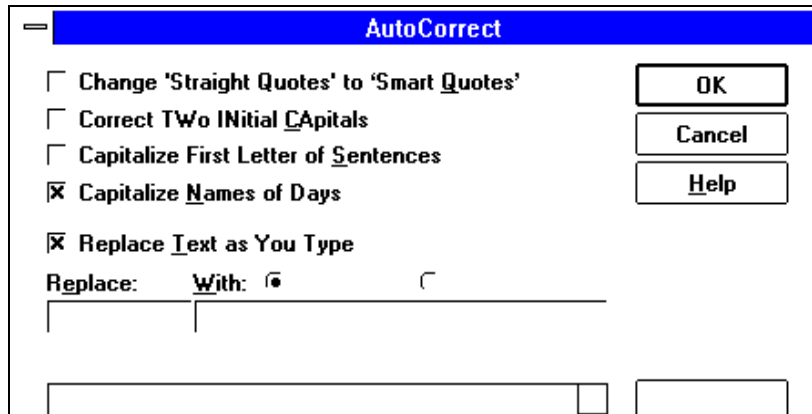
Chọn **Tools - AutoCorrect**. Hộp thoại AutoCorrect xuất hiện như hình 19.4. Giả sử chúng ta muốn viết tắt dòng chữ " Cộng hoà xã hội chủ nghĩa Việt nam " bằng cách bấm hai chữ cái vn

Dưới mục chọn Replace ta viết chữ "vn"

Dưới mục chọn With ta viết dòng chữ Cộng hoà xã hội chủ nghĩa Việt nam sau đó bấm chuột vào nút Add rồi chọn OK.

Trong văn bản từ nay trở đi khi ta gõ vn và bấm phím khoảng cách thì toàn bộ dòng chữ Cộng hoà xã hội chủ nghĩa Việt nam sẽ xuất hiện thay cho chữ vn.

Các từ viết tắt tạo ra như trên chỉ có thể viết được một dòng còn muốn tạo ra nhiều dòng thì phải dùng phương pháp khác.



Hình 19.4

#### 5.4 - Tạo đoạn văn bản mẫu

Nếu hàng này chúng ta phải thường xuyên gửi công văn đi các nơi thì phần tiêu đề của công văn có thể tạo sẵn và lưu trong máy sau đó chỉ việc gọi ra và điền nội dung vào

Giả sử cần tạo ra một tiêu đề mẫu sau đây:

**Cộng hoà xã hội chủ nghĩa việt nam**  
***Độc lập - Tự do - Hạnh phúc***

**Công văn**

*Hà nội, ngày ..... tháng ..... năm 199...*

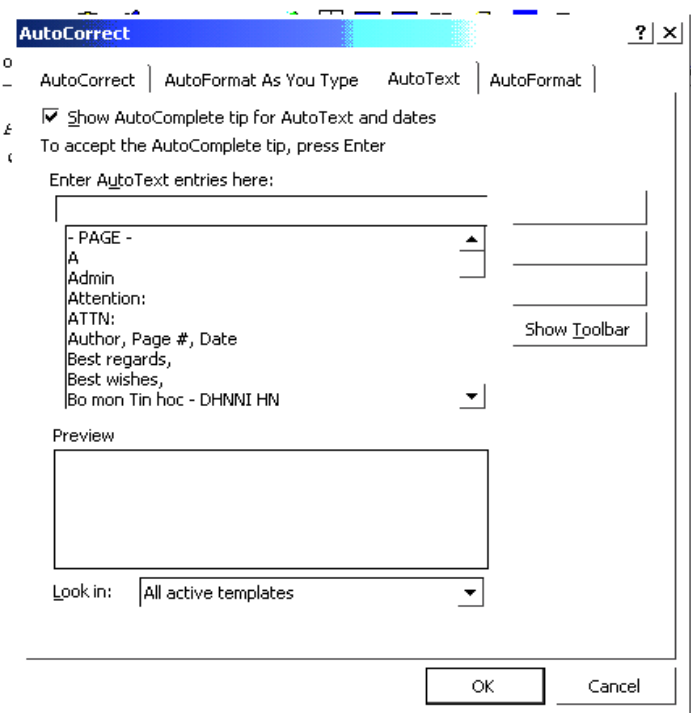
Các bước thực hiện:

- \* Viết đoạn tiêu đề công văn
- \* Trình bày kiểu chữ theo ý muốn
- \* Bôi đen đoạn tiêu đề
- \* Chọn Tools - Autocorrect - AutoText ...
- \* Trong hộp thoại Autocorrect (Hình 20.4) dưới mục Enter Autotext entries here ta viết một tên cho đoạn văn bản này. Tên cần chọn sao cho dễ nhớ và chỉ được viết bằng tiếng Việt không dấu. Tiếp đó bấm đơn vào nút Add

Khi cần điền đoạn tiêu đề trên vào trang văn bản ta thực hiện các bước:

Chọn Insert - AutoText, tìm trong khung lựa chọn tên của đoạn văn bản cần thiết, bấm đơn vào tên đó cho nó nằm trong nền xanh tiếp đó chọn tiếp nút Insert.

Do hạn chế thời gian của chương trình học tập trong tài liệu này chỉ giới thiệu những gì mà sinh viên có thể tiếp thu trong khoảng 6 tiết học. MS-WORD 2000 là một hệ soạn thảo đa năng với rất nhiều chức năng phong phú muốn khai thác hết cần có thời gian học lý thuyết dài hơn và đặc biệt là cần được thao tác trên máy nhiều hơn.



Hình 20.4

### 5.5 - Viết số mũ và chỉ số

\* Viết số mũ: ví dụ  $x^3$

Đầu tiên viết chữ x sau đó bấm tổ hợp phím Ctrl+ Shift + =, con trỏ lúc này sẽ chuyển lên vị trí cao để ta gõ số 3, gõ lại một lần nữa tổ hợp phím trên con trỏ sẽ trở lại bình thường

\* Viết chỉ số: ví dụ  $x_1$

Gõ chữ x sau đó bấm tổ hợp phím Ctrl + =, con trỏ dịch xuống vị trí thấp ta bấm tiếp số 1 rồi bấm lại tổ hợp phím trên một lần nữa.

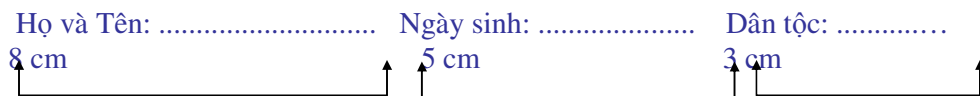
### 5.6 - Chèn vào văn bản một ký tự đặc biệt

Chọn Insert - Symbol ta sẽ có một bảng các ký tự đặc biệt, các ký tự này thường rất nhỏ khó nhận dạng, muốn nhìn rõ bấm đôn vào ký tự đó. Muốn chèn ký tự này vào văn bản bấm kép vào ký tự hoặc bấm vào nút Insert ở phía dưới. Xong việc bấm Close.

## 6 - Tự động hoá quá trình soạn thảo

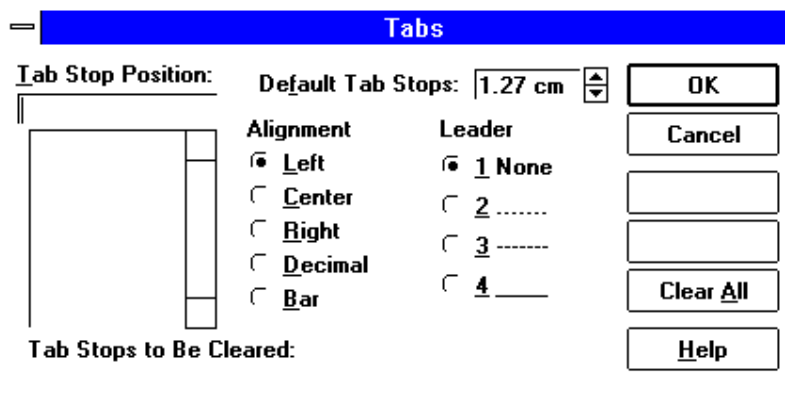
### 6.1 - Tạo dòng Tab

Khi bấm phím Tab trên bàn phím con trỏ sẽ tự động dịch chuyển một đoạn, Word ngầm định đoạn này là 1,27 Cm (1/2 Inch). Khi dịch chuyển con trỏ có thể để lại trên dòng các dấu chấm, các nét gạch ngang. Giả sử ta cần tạo ra dòng văn bản sau:



Chữ Họ và tên viết cách mép trái 1 cm. Phần dành cho Họ và tên là 8 cm, phần dành cho ngày sinh là 5 cm và phần dành cho dân tộc là 3 cm. Cách thức tạo như sau:

Chọn Format - Tab ta có cửa sổ (Hình 21.4)



Hình 21.4

Tab Stop Psition : gõ 1  
 Alignment : chọn Left  
 Leader : chọn None  
 Chọn Set

Tiếp đó lặp lại các thao tác  
 Tab Stop Psition : gõ 8  
 Alignment : chọn Left  
 Leader : chọn số 2  
 Chọn Set

Tab Stop Psition : gõ 5  
 Alignment : chọn Left  
 Leader : chọn số 2  
 Chọn Set

Tab Stop Psition : gõ 3  
 Alignment : chọn Left  
 Leader : chọn số 2  
 Chọn Set

Chọn OK để trở về cửa sổ soạn thảo, khi đó trên dòng thước kẻ (Rule) xuất hiện một số ký hiệu thể hiện vị trí Tab.

Dòng Tab vừa tạo ra sẽ có tác dụng cho đến khi ta tạo ra dòng Tab mới

## 6.2 - Tạo các STYLE

Style được hiểu là tập hợp những thao tác định dạng cho một đoạn văn bản do người sử dụng lựa chọn, chúng được lưu trữ dưới một tên gọi và có thể sử dụng bất kỳ lúc nào. Khi một Style được gọi toàn bộ phần văn bản viết sau đó sẽ được định dạng theo Style này. Phần định dạng đưa vào một Style có thể là kiểu chữ, dạng chữ, kích thước, màu sắc chữ, khoảng cách giữa các dòng, giữa các cột, giữa các đoạn ....

Giả sử cần tạo ra các Style có tên là St1, St2, St3

Trong St1 sẽ đưa vào các định dạng:

Kiểu chữ: VnTimeH, kích thước: 14, màu đỏ, chữ nghiêng

Trong St2 sẽ đưa vào các định dạng:

Kiểu chữ: Vnaritote , kích thước: 16, màu xanh  
 Trong St1 sẽ đưa vào các định dạng:  
 Kiểu chữ: VnGothich , kích thước: 16, màu tím, chữ nghiêng và đậm

**a) Tạo Style mới**

Chọn Format - Style khi đó xuất hiện cửa sổ (Hình 22.4)  
 Chọn tiếp New nếu muốn tạo Style mới, nếu sửa Style cũ thì chọn Modyfi, giả sử ta chọn New, khi đó xuất hiện cửa sổ hình 23.4

Dưới mục **Name** ( tên của Style mới ) ta gõ St1.

Mục **Style Type** có hai lựa chọn:

**Character** để định dạng chữ, **Paragraph** để định dạng đoạn văn bản, ngầm định là Paragraph

Giả sử ta chọn Paragraph, sau đó chọn nút Format khi đó sẽ xuất hiện hộp thoại bao gồm:

**Font:** định dạng chữ

**Paragraph:** định dạng đoạn

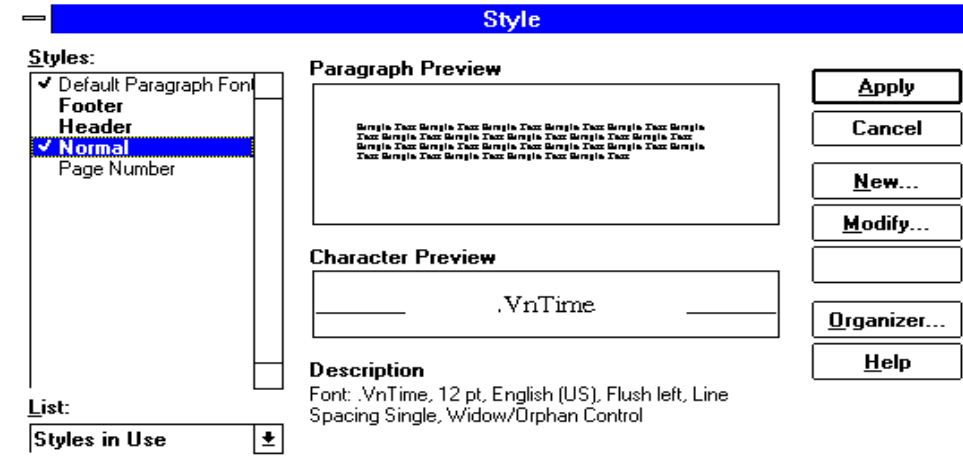
**Tab:** định dạng bước nhảy khi bấm phím Tab

**Border:** định dạng đường bao ( bảng biểu)

**Language:** chọn ngôn ngữ

**Frame:** định dạng khung

**Numbering:** định dạng kiểu số



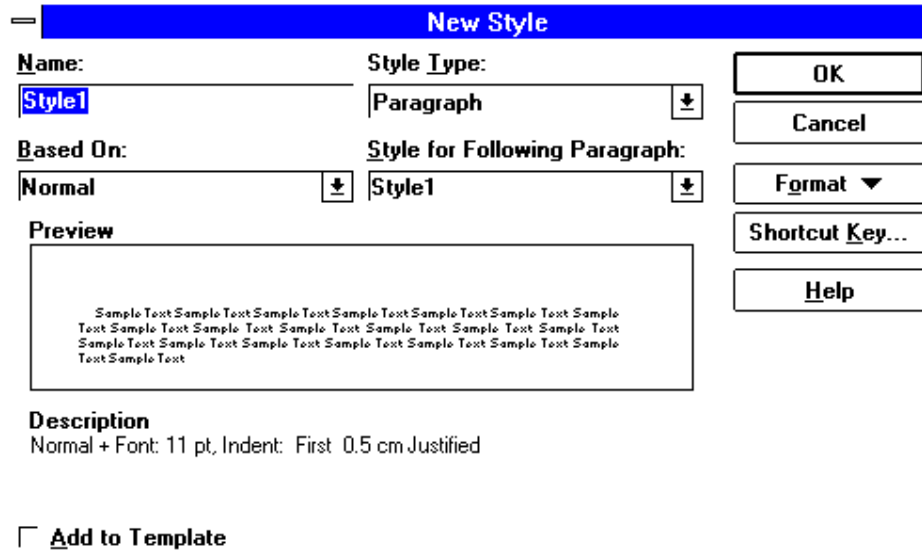
Hình 22.4

Chọn một trong các chức năng của hộp thoại ví dụ chọn Font ta sẽ quay về cửa sổ quen thuộc để làm các thao tác định dạng, cụ thể ta sẽ chọn: Kiểu chữ: VnTimeH, kích thước:14, màu đỏ, chữ nghiêng

Sau khi đã lựa chọn xong chọn OK để quay về các bước trước đó, cuối cùng chọn Close để trở về màn hình văn bản hiện thời. Nếu muốn ứng dụng ngay Style vừa tạo ra thì chọn Apply.

Tên của Style vừa tạo ra được để trong thanh định dạng. Nếu muốn ứng dụng Style nào ta chỉ việc kích chuột vào mũi tên bên phải nút Style tên các Style sẽ hiện lên trong hộp lựa chọn.

Các Style St2, St3 cũng thực hiện tương tự.



Hình 23.4

### 6.3 - Tạo Macro

Macro tạm hiểu là một vĩ lệnh, nó giống như một máy ghi, ghi lại toàn bộ các thao tác mà người sử dụng đã tiến hành. Khi gọi tên Macro các thao tác này sẽ được lặp lại từ đầu cho đến thao tác cuối cùng.

Macro có thể được lưu trữ trên thanh công cụ, trên bàn phím (với các phiên bản trước window2000 có thể để Macro trong Menu.)

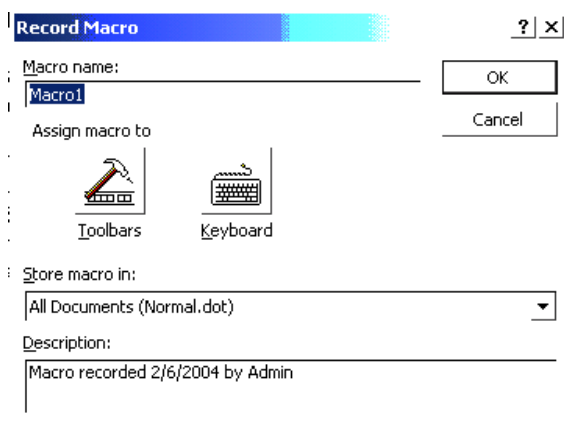
#### a) Tạo Macro trên ToolBar

Các bước tiến hành:

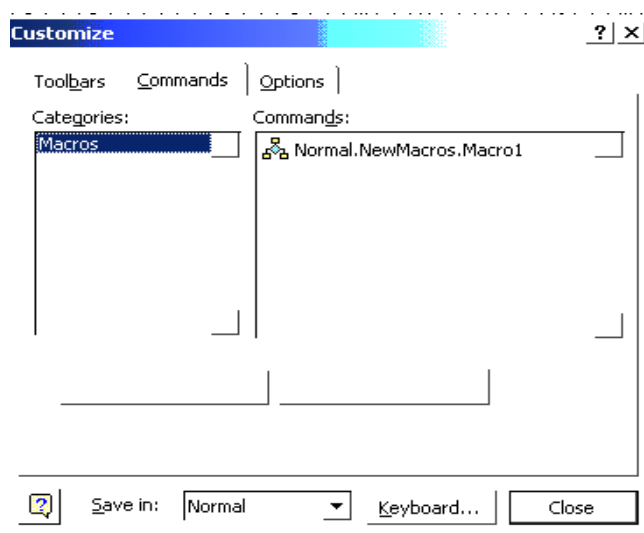
- Chọn Tools
- Chọn Macro - Record new macro, xuất hiện cửa sổ (Hình 24.4 )
- Trong mục Macro Name ta ghi một tên nào đó ví dụ M1
- Trong mục Assign Macro to (đặt macro vào) chọn Toolbars có cửa sổ tiếp theo, cách chọn này có nghĩa là Macro sẽ lưu thành một nút trên thanh công cụ, nếu chọn KeyBoard thì Macro được lưu trên bàn phím.
- Dưới mục Commands bấm và di cả dòng chữ Normal.NewMacros.Macro1 lên thanh công cụ sẽ tạo nên một nút mới, bên trong nút này chưa có biểu tượng mà chỉ có dòng chữ đã nói. Để thu bé biểu tượng chọn tiếp Modify Selection sẽ xuất hiện cửa sổ hình 25.4

Bên phải mục Name trong hộp thoại hãy xóa bớt số ký tự đã có và điền vào đó tên mà ta lựa chọn. Cần lưu ý rằng không thể xóa bỏ tất cả mọi ký tự mà phải điền vào ít nhất một ký tự.

Mục Change Button Image cho phép lựa chọn một hình vẽ cho biểu tượng macro , chỉ cần bấm đơn vào biểu tượng là nó sẽ tự động được đưa và vị trí nút mà ta đã chọn.



Hình 24.4



Hình 25.4

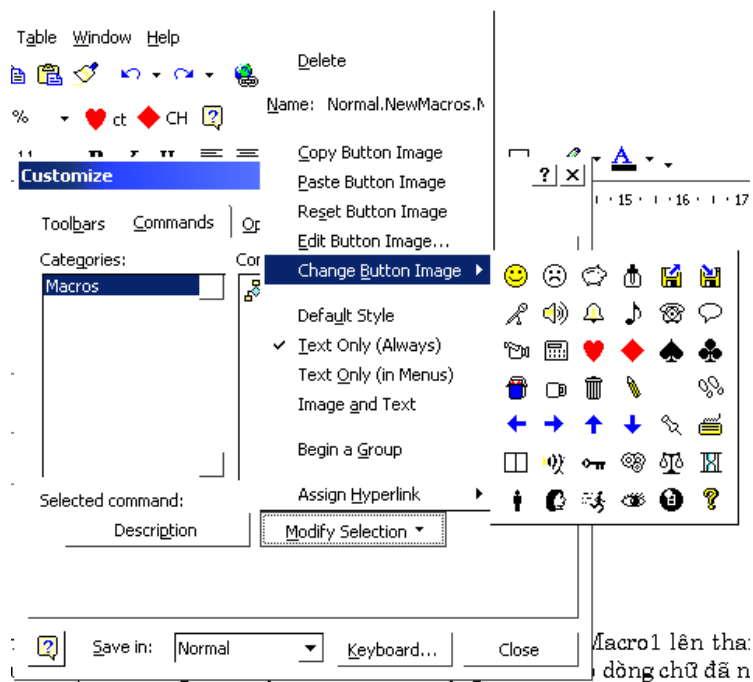
Chọn Close để quay về cửa sổ soạn thảo, lúc này trên màn hình xuất hiện một thanh công cụ gồm hai nút:

Stop: dùng để dừng Macro

Pause: dùng để tạm ngừng ghi Macro

Con trỏ chuột trên màn hình xuất hiện thêm một biểu tượng giống như băng cát xet. Kể từ thời điểm này mọi thao tác trên bàn phím sẽ được ghi lại và được gửi vào biểu tượng vừa tạo ra trên thanh công cụ.

Kết thúc công việc tạo Macro bấm vào nút Stop.



Hình 26.4

**Chú ý :**

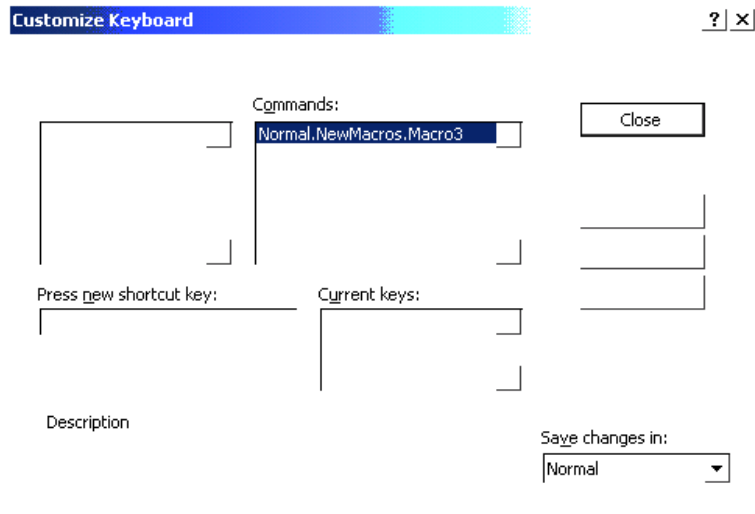
- \* Khi thực hiện các thao tác tạo Macro, ta không thể bôi đen văn bản bằng thiết bị chuột mà phải dùng phím Shift và các mũi tên dịch chuyển.
- \* Để gọi Macro ta chỉ việc bấm vào biểu tượng đã tạo ra trên thanh công cụ.
- \* Để xoá một Macro đã có trên thanh công cụ ta đề phím Alt đồng thời bấm rê kéo biểu tượng Macro vào vùng soạn thảo, biểu tượng sẽ biến mất.
- \* Để xoá tên Macro chọn: Tools - Macro - chọn tên Macro rồi chọn Delete

**b) Tạo Macro trên bàn phím (Keyboard)**

Các bước tiến hành:

- Lặp lại các bước 1a - 1c
  - Chọn Keyboard sẽ xuất hiện cửa sổ mới (Hình 27.4)
  - Trong mục chọn Press New Shortcut Key ta bấm một tổ hợp phím ví dụ Ctrl - Alt - A,
  - Chọn Assign, tổ hợp phím trên được ghi vào vùng Current Key
  - Chọn Close để trở về màn hình soạn thảo
- Các thao tác tiếp theo sẽ lặp lại như mục 1





Hình 27.4

### c) Gọi Macro

Để gọi một Macro tiến hành các thao tác sau:

- \* Đặt con trỏ ở đầu một dòng trống
- \* Chọn Tools - Macro - Macro xuất hiện cửa sổ mới
- \* Chọn tên Macro trong danh mục sau đó chọn Run

## 7 - Đồ họa

### 7.1 - Các thao tác vẽ cơ bản

Đồ họa là phần công cụ phụ trợ trong Winword, ở đây chỉ bao gồm những chức năng cơ bản, nếu muốn sử dụng công cụ đa năng hơn ta có thể dùng PowerPoint là một ứng dụng độc lập với Winword và cũng được cài đặt trong Windows.

Chọn nút Drawing trên thanh công cụ ta sẽ có thêm một thanh công cụ mới, thanh này thường được bố trí dưới đáy màn hình (Hình 28.4). Nếu máy cài đặt bộ Office 97 hoặc office 4.3 thì thanh công cụ vẽ có thể khác đi đôi chút vì đó là các phiên bản cũ.



Hình 28.4

Thanh công cụ đồ họa bao gồm các nút có tên như sau (theo thứ tự từ trái sang phải):

- |                 |                                             |
|-----------------|---------------------------------------------|
| Draw            | : Kẻ lưới, quay hình, chọn thứ tự hình .... |
| Select Object   | : Chọn đối tượng vẽ                         |
| Free Rotate     | : Quay hình tự do                           |
| AutoShapes      | : Chọn các hình mẫu có sẵn                  |
| Line            | : Vẽ đường thẳng                            |
| Arrow           | : Vẽ mũi tên                                |
| Rectangle       | : Vẽ hình vuông hoặc chữ nhật               |
| Oval            | : Vẽ hình tròn hoặc Ellipse                 |
| Text Box        | : Tạo một hộp chứa văn bản                  |
| Insert WordArt  | : Chèn chữ nghệ thuật                       |
| Insert Clip Art | : Chèn ảnh                                  |

|            |                                        |
|------------|----------------------------------------|
| Fill Color | : Chọn màu nền cho những hình khép kín |
| Line Color | : Chọn màu nét vẽ                      |
| Line Style | : Chọn dạng nét vẽ                     |
| Font Color | : Chọn màu cho chữ                     |
| Dash Style | : chọn kiểu nét vẽ                     |
| Arow Style | : Chọn kiểu mũi tên                    |
| Shadow     | : Chọn hình có bóng                    |
| 3-D        | : Chọn hình không gian 3 chiều         |

Trong mục chọn Draw chúng ta có thể tìm được các chức năng sau (có thể một vài chức năng đã bị xoá):

|                        |                                                            |
|------------------------|------------------------------------------------------------|
| Bring to Front         | : Chuyển một hình vẽ ở dưới lên phía trên                  |
| Send to Back           | : Chuyển hình vẽ ở trên xuống dưới                         |
| Bring in Front of Text | : Đưa hình che lên trên văn bản                            |
| Send Behind Text       | : Cho văn bản hiện lên trên hình                           |
| Group                  | : Gộp một số hình thành một nhóm                           |
| Ungroup                | : Tách một nhóm hình thành các phần riêng rẽ               |
| Flip Horizontal        | : Chuyển hình bên trái sang bên phải                       |
| Flip Vertical          | : Chuyển hình phía trên xuống phía dưới                    |
| Rotate Right           | : Quay hình 90° theo chiều kim đồng hồ                     |
| Reshape                | : Sửa hoặc bẻ gãy nét vẽ tự do (Freedom)                   |
| Snap to Grid           | : Kẻ lưới trên đối tượng vẽ                                |
| Align Drawing Objects  | : Canh lề các hình vẽ (dóng thẳng lề theo một hình nào đó) |
| Create Picture         | : Tạo (sửa chữa một bức vẽ đã có)                          |
| Insert Frame           | : Chèn thêm một khung (chứa văn bản vào trong bức vẽ)      |

- Chọn một trong năm dạng nét vẽ ( Line, Rectangle, Ellipse....) bằng cách bấm đơn vào biểu tượng nét vẽ đó, chuyển chuột vào vùng soạn thảo, chuột sẽ biến thành hình dấu "+".

- Bấm di chuột để tạo hình có kích thước mong muốn.

- Đánh dấu hình: Đưa chuột đến gần hình chuột sẽ xuất hiện thêm biểu tượng hai mũi tên vuông góc. Bấm đơn để đánh dấu hình (xuất hiện các khối vuông đen bao quanh hình)

- Tô màu:

\* Nếu là nét vẽ không khép kín thì chỉ có thể tô màu nét vẽ, chọn Line Color - chọn một màu nào đó

\* Nếu là nét vẽ khép kín ( hình vuông , tròn, tam giác....) việc chọn màu nét vẽ như mục 3, để chọn màu nền chọn Fill Color - rồi chọn màu

- Chọn dạng nét vẽ: Đánh dấu hình rồi chọn nút Line Style - sau đó chọn một nét thích hợp.

- Muốn đưa một dòng chú thích và mũi tên chỉ vào một vị trí nào đó:

+ Chọn AutoShapes

+ Chọn Callout

+ Đặt chuột ( dấu "+") vào vị trí cần chỉ mũi tên

+ Bấm di chuột sẽ tạo ra một hình để ghi chú thích bên trong

**Một số điều cần lưu ý:**

\* **Chọn nhiều hình cùng một lúc**

Đè phím Shift rồi đánh dấu lần lượt từng hình

\* **Chuyển vị trí hình**

Khi vẽ hình, hình vẽ sau sẽ che lên trên hình vẽ trước. Muốn thấy hình vẽ trước ta thực hiện:

- Đánh dấu hình vẽ sau

- Chọn Draw - Order
- Chọn nút Send to Back

Ngược lại hình vẽ sau có thể đưa lên trước bằng cách đánh dấu hình rồi chọn nút Bring to Front

\* Nếu có hai hình nằm theo hàng ngang và cả hai hình đang được đánh dấu ta có thể: quay hình đi 90° ( nút Rotate Right), chuyển hình trái sang phải (Flip Horizontal)

\* Nếu có hai hình nằm theo hàng dọc và cả hai hình đang được đánh dấu ta có thể: quay hình đi 90° ( nút Rotate Right), chuyển hình trên xuống dưới (Flip Vertical)

**\* Viết chữ trong hình:**

- Chọn nút Text Box rồi vẽ trong hình một khung, trong khung sẽ có con trỏ để viết văn bản

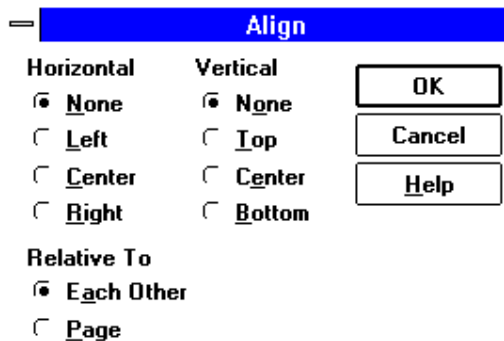
- Nếu không có Text Box ta có thể viết trực tiếp trong hình, chữ viết ra không nhìn thấy vì nằm bên dưới hình, muốn chữ hiện lên:

- Đánh dấu hình
- Chọn nút: Send Behind text

**Chú ý:** Để có thể đưa con trỏ vào trong hình vẽ, trước khi vẽ cần đưa con trỏ xuống đáy màn hình. Việc dịch chuyển con trỏ sang ngang phải dùng phím Space (phím khoảng cách)

**\* Dóng vị trí ( canh lề) các hình:**

- Vẽ hai hình ở vị trí bất kỳ trên màn hình,
- đánh dấu cả hai hình
- chọn Align Drawing Objects, xuất hiện cửa sổ (Hình 29.4)



Hình 29.4

Horizontal: (theo chiều ngang)

- None : Không canh lề
- Left : Dóng thẳng theo cạnh trái của hình
- Center : Đặt hình vào giữa theo chiều ngang
- Right : Dóng thẳng theo cạnh phải của hình

Vertical: (theo chiều dọc)

- None : Không canh lề theo chiều dọc
- Top : Dóng thẳng theo mép trên của hình
- Center : Dóng thẳng theo đường xuyên tâm ngang
- Bottom : Dóng thẳng theo mép dưới của hình

### **Bẻ gãy nét vẽ:**

Chọn AutoShapes - Lines - freedom vẽ 1 nét bất kỳ bao gồm một số đoạn thẳng và đoạn cong (Hình 30.4), đánh dấu nét vẽ

\*Chọn Draw - Edit Points nét vẽ sẽ được đánh dấu tại các điểm giao nhau (Hình 31.4)



Hình 30.4



Hình 31.4

Trên nét vẽ các đoạn thẳng sẽ được đánh dấu ở hai đầu, còn các đoạn cong sẽ được đánh dấu liên tục bởi các ký hiệu đánh dấu. Đưa chuột vào một trong các ký hiệu đánh dấu bấm rê chuột ta sẽ uốn được nét vẽ theo ý muốn. Nếu muốn bẻ gãy nét vẽ tại một vị trí của đoạn thẳng (trên đó không có ký hiệu đánh dấu) ta có thể làm như sau:

- Đưa chuột đến vị trí cần bẻ gãy
- Bấm rê chuột, nét vẽ sẽ bị bẻ và tạo cho ta vị trí gãy như một cánh sao.

### **7.2 - Chữ nghệ thuật**

Chữ nghệ thuật được hiểu là các kiểu chữ phi tiêu chuẩn đã được thiết kế sẵn trong Word. Chữ nghệ thuật được sử dụng để viết các khẩu hiệu, trang trí trên các hình vẽ v.v... Thao tác viết chữ nghệ thuật được thực hiện qua các bước sau:

Bấm đơn vào nút đồ họa Drawing trên thanh công cụ để xuất hiện thanh công cụ đồ họa

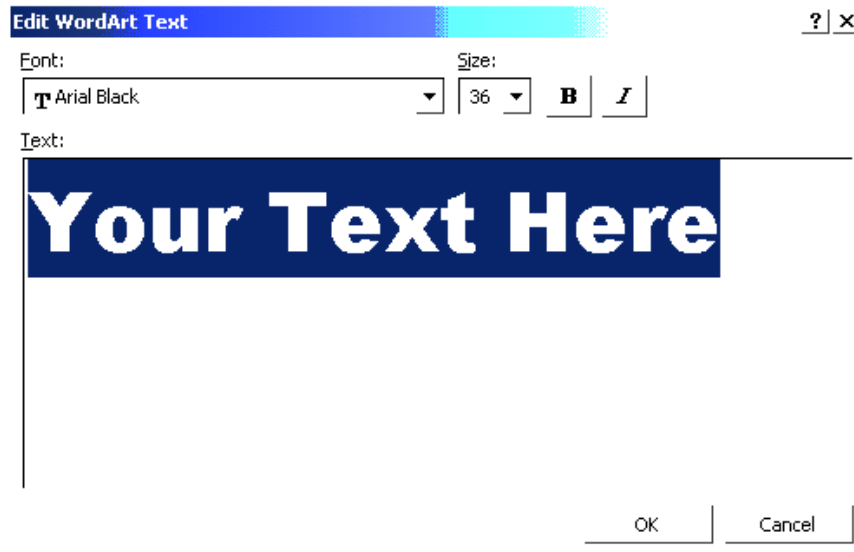
Bấm đơn vào nút Insert WordArt sẽ xuất hiện cửa sổ WordArt Gallery (Hình 32.4)

Chọn một kiểu chữ nghệ thuật mà bạn muốn rồi bấm OK sẽ xuất hiện cửa sổ Edit WordArt Text (Hình 33.4)

Font chữ hiện thời là font tiếng anh. Để chọn Font tiếng Việt trong mục Font kích chuột vào mũi tên bên phải sau đó chọn một Font tiếng Việt, ví dụ chọn VnAristote



Hình 32.4



Hình 33.4

Trong mục Size có thể chọn kích thước chữ, ngoài ra bạn có thể chọn kiểu chữ đậm (B), nghiêng (I).

Xoá dòng chữ Your Text Here và gõ dòng chữ cần trình bày ví dụ :

Chúc mừng sinh nhật

Chọn OK

Dòng chữ sẽ hiện trên màn hình như hình 34.4 đồng thời có một thanh công cụ hiện kèm theo. Sử dụng thanh công cụ này bạn có thể thay đổi dòng chữ theo ý muốn.

# Chúc mừng sinh nhật

Hình 34.4

Các nút trên thanh công cụ này là:

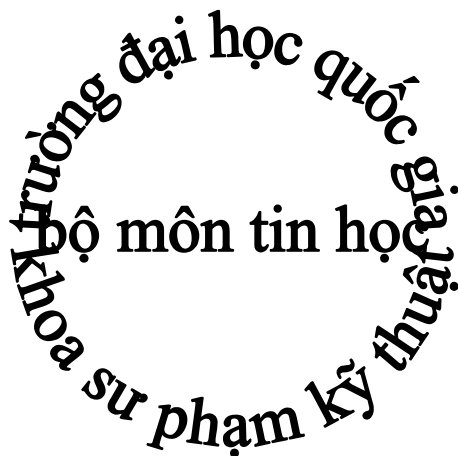
- \* Insert WordArt: chọn một kiểu chữ mới
- \* Edit Text: sửa lại chữ cũ
- \* WordArt Gallery: thư viện các kiểu chữ nghệ thuật
- \* Format WordArt: Định dạng nét và màu chữ
- \* WordArt Shape: thay đổi kiểu trình bày
- \* Free Rotate: quay tự do
- \* Text Wrapping: cách thức hiện chữ và hình
- \* WordArt Same letter Hights: biến chữ hoa thành chữ thường và ngược lại
- \* WordArt Vertical text: chuyển chữ nằm ngang thành thẳng đứng và ngược lại
- \* WordArt Agliment: canh lề chữ
- \* WordArt Character Spacing: chọn chữ gầy hay béo

Hình 35.4 cho ta một dạng của chữ nghệ thuật, muốn có chữ kiểu con dấu, ta phải gõ vào vùng your text here ba dòng khác nhau. Ví dụ:

Trên cùng là dòng Trường Đại học Quốc gia

Tiếp đó là dòng Bộ môn Tin học

Dưới cùng là dòng Khoa Sư phạm kỹ thuật



Hình 35.4

## Chú ý:

\* Việc chọn Font chữ hoặc chọn màu chỉ có thể thực hiện trong cửa sổ của Word Art mà không thể chọn theo cách thông thường.

\* Muốn sửa lại một WordArt đã có, bấm kép vào vùng chữ đã được tạo ra ta sẽ trở lại cửa sổ WordArt.

\* Với một WordArt đã có ta có thể mở rộng hoặc thu hẹp kích thước bằng cách bấm đơn vào vị trí bất kỳ thuộc Word Art đó, lúc này toàn bộ Word Art sẽ nằm trong một hình chữ nhật. Các cạnh của hình chữ nhật được đánh dấu bởi các hình vuông đen. Đặt chuột vào một trong các hình vuông đen đó, bấm rê chuột sẽ thay đổi được kích thước của WordArt.

## 8 - Trộn tài liệu tạo thành văn bản mới

### Nội dung:

Trộn dữ liệu trong một bảng biểu đã có sẵn ( đã lưu trong đĩa cứng với một tên nào đó ) với một văn bản mẫu để tạo thành một văn bản mới. Ví dụ đã có sẵn một mẫu giấy mời và một danh sách khách mời ta sẽ trộn chúng với nhau để có một tập giấy mời hoàn chỉnh.

### Các bước tiến hành:

- \* Tạo sẵn mẫu giấy mời và ghi vào đĩa với tên là GM.
- \* Tạo một bảng biểu chứa danh sách những người sẽ mời
- \* Trộn lẫn hai văn bản này thành một giấy mời hoàn chỉnh

### 8.1 - Tạo mẫu giấy mời

|                                                                                                                                                                                                                                                                                                                                      |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>cộng hoà xã hội chủ nghĩa việt nam</b><br/><b>độc lập tự do hạnh phúc</b></p> <p>-----</p> <p>Giấy mời</p> <p>Trân trọng kính mời Ông/Bà:</p> <p>Đến dự :</p> <p>Thời gian :</p> <p>Địa điểm :</p> <p style="text-align: center;">Rất hân hạnh được đón tiếp</p> <p style="text-align: center;"><i>Ngày tháng năm 1996</i></p> |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

- \* Ghi giấy mời vào đĩa với tên GM

### 8.2 - Tạo danh sách khách mời

- \* Chọn nút NEW để mở một cửa sổ mới che lên trên cửa sổ chứa giấy mời
- \* tạo một bảng chứa danh sách khách mời, ví dụ bảng có bảng 4 cột và 6 dòng theo mẫu dưới đây

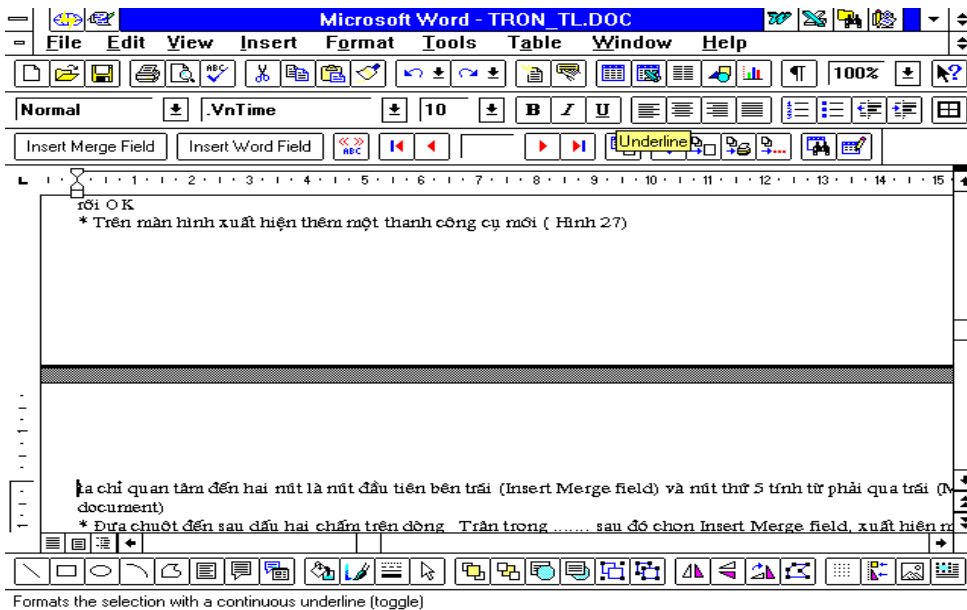
| Hoten           | noidung               | thoigian | diadiem            |
|-----------------|-----------------------|----------|--------------------|
| Nguyễn Văn Tâm  | Dự họp tổng kết       | 8h30     | Hội trường 204     |
| Trần Thanh Bình | Dự liên hoan chia tay | 11h30    | Nhà hàng Hương sen |
| Cao Thị Vân     | Học nghị quyết        | 7h30     | Phòng 12 - A1      |
| Đào Hoàng Lâm   | Họp thi đua           | 7h30     | Phòng hội thảo     |
| Phạm Minh Hồng  | Dự cưới               | 10h30    | Nhà riêng          |

### Lưu ý:

*Dòng đầu tiên của bảng biểu là Tiêu đề do đó không viết tiếng việt có dấu, không được viết các chữ cách nhau - Không trang trí đường viền của bảng biểu*

- \* Ghi bảng biểu này vào với tên DS (danh sách)
- \* Chọn FILE - CLOSE đóng cửa sổ bảng biểu lại, lúc này cửa sổ chứa giấy mời lại hiện ra.
- \* Chọn TOOLS - MAIL MERGE - CREAT - FORM LETTERS...- ACTIVE WINDOS

- \* Chọn tiếp GET DATA - OPEN DATA SOURCE Khi đó hiện lên tên các tệp lưu trong đĩa cứng của máy, chọn tên tệp DS rồi OK
- \* Trên màn hình xuất hiện thêm một thanh công cụ mới ( Hình 36.4)



Hình 36.4

ta chỉ quan tâm đến hai nút là nút đầu tiên bên trái (Insert Merge field) và nút thứ 5 tính từ phải qua trái (Merge to New document)

- \* Đưa chuột đến sau dấu hai chấm trên dòng Trộn trọng ..... sau đó chọn Insert Merge field, xuất hiện một khung nhỏ chứa tiêu đề các cột bảng biểu - nháy chuột vào Hoten. Lúc này bên cạnh dấu hai chấm sẽ xuất hiện một nhóm ký tự lạ
- \* Tiếp tục thao tác với các dòng Thời gian, Địa điểm ...
- \* Cuối cùng nháy chuột vào nút Merge to New document ta sẽ có được số giấy mời cần thiết
- \* Ghi văn bản này vào đĩa hoặc đóng cửa sổ lại không ghi.

**Chú ý:**

Nếu muốn trình bày kiểu chữ cho văn bản sẽ trộn ta chỉ việc bôi đen cụm ký tự vừa điền vào giấy mời sau đó chọn Format - Font và làm các thao tác như đã trình bày ở phần đầu.

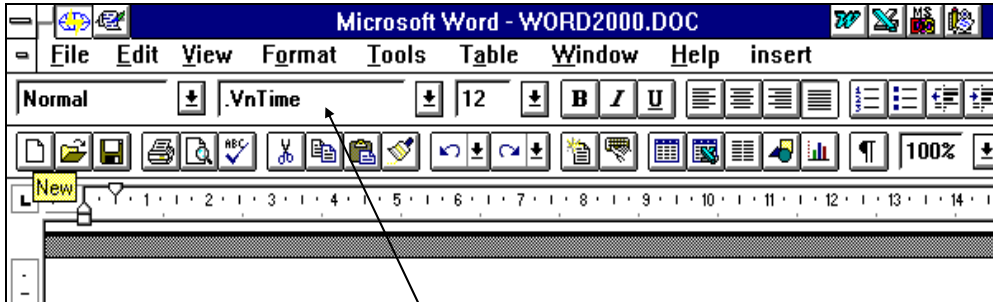
Sau khi trộn có thể mỗi giấy mời sẽ nằm trên một trang giấy. Muốn ghép một số giấy mời vào một trang ta cần để con trỏ ở dưới giấy mời ( cách một , hai dòng ) sau đó bấm hai lần phím Delete , giấy mời phía dưới sẽ được kéo lên trang hiện thời.

**9 - Các thao tác hỗ trợ**

**9.1 - Khôi phục các thanh công cụ**

Với những người mới làm quen với Word rất dễ xảy ra tình trạng làm dịch chuyển hoặc mất thanh công cụ, thanh định dạng. Phổ biến nhất là thanh định dạng nằm phía trên, thanh công cụ nằm dưới (Hình 37.4).





Vị trí bấm và di chuột

Hình 37.4

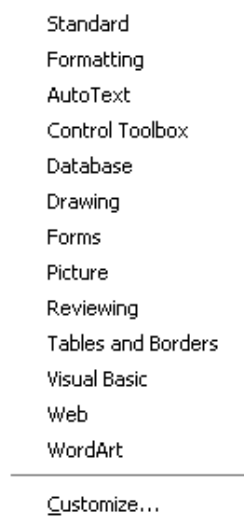
Để đưa thanh định dạng xuống dưới ta chỉ việc đưa chuột vào vị trí trống giữa các nút hoặc nhóm nút sau đó đè phím trái chuột, lúc này đường viền thanh sẽ biến thành nét đứt. Bấm và di thanh xuống phía dưới. Khi thanh định dạng nằm trùng lên thanh công cụ thì buông nút trái chuột.

## 9.2 - Hiện hoặc giấu các thanh công cụ

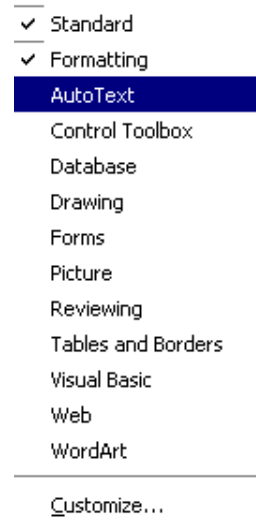
Ngầm định trên màn hình chỉ có 2 thanh công cụ là thanh Standard và thanh Formatting tức là thanh công cụ chuẩn và thanh định dạng. Một số nút trên thanh Standard như nút Drawing (đồ họa), nút Table and Border (vẽ bảng) khi chọn lại cho hiện lên thanh công cụ tương ứng. Việc cho hiện hoặc giấu các thanh công cụ có thể thực hiện theo các bước sau đây:

- Khi một nút ứng với một thanh nào đó đang có màu trắng (nghĩa là nó đang được chọn) thì thanh công cụ tương ứng sẽ xuất hiện trên màn hình. Bấm đơn vào nút này thanh công cụ tương ứng sẽ biến mất và nút trở thành màu xám. Bấm tiếp vào nút thanh công cụ sẽ lại hiện ra.

- Nếu trên màn hình không có một thanh công cụ nào, hoặc thiếu thanh công cụ cần thiết, đưa chuột vào một vị trí bất kỳ trên thanh thực đơn, bấm phím phải chuột sẽ xuất hiện một hộp thoại (Hình 38.4)



Hình 38.4



Hình 39.4

Bấm đon vào tên thanh công cụ muốn chọn thanh này sẽ xuất hiện trên màn hình, thao tác tương tự với các thanh khác.

**Chú ý:**

- Nếu phía bên trái tên thanh công cụ đã có ký hiệu đánh dấu (Hình 39.4 ) mà trên màn hình vẫn không thấy thanh công cụ đó thì cần tìm xem nó nằm đâu đó trên màn hình, có khi chỉ nhìn thấy một phần rất nhỏ của thanh công cụ, phần còn lại nằm khuất ngoài phạm vi màn hình. Chỉ cần đưa chuột vào phần nhỏ đó bấm rê nó vào trong màn hình là được.

- Không nên cho hiện lên quá nhiều thanh công cụ trên màn hình vì như vậy sẽ không còn không gian để làm việc. Khi dùng xong một thanh công cụ nào nếu không cần thiết thì nên cất nó đi.

**9.3 - Thêm bớt chức năng trên thanh công cụ**

Mỗi thanh công cụ của Word có một số biểu tượng (nút) thể hiện một số chức năng, chúng ta có thể tùy ý thêm bớt các chức năng vào các thanh này hoặc chuyển các nút từ thanh này sang thanh khác.

**a) Xoá hoặc chuyển vị trí một nút**

Đè phím Alt trên bàn phím sau đó bấm rê nút muốn xoá vào vùng soạn thảo, buông hai tay ra nút sẽ bị xoá.

Nếu chúng ta lôi nút đến một thanh công cụ khác rồi buông tay ra thì nút sẽ nằm trên thanh công cụ đó.

**b) Thêm một nút mới vào thanh công cụ**

Giả sử bạn luôn phải định dạng chữ, hãy thêm vào thanh Standard các nút thể hiện Font chữ hay dùng. Cách thức tiến hành như sau:

Chọn Tools - Customize (hoặc đưa chuột lên thanh công cụ rồi bấm phím phải sau đó chọn Customize) xuất hiện cửa sổ.

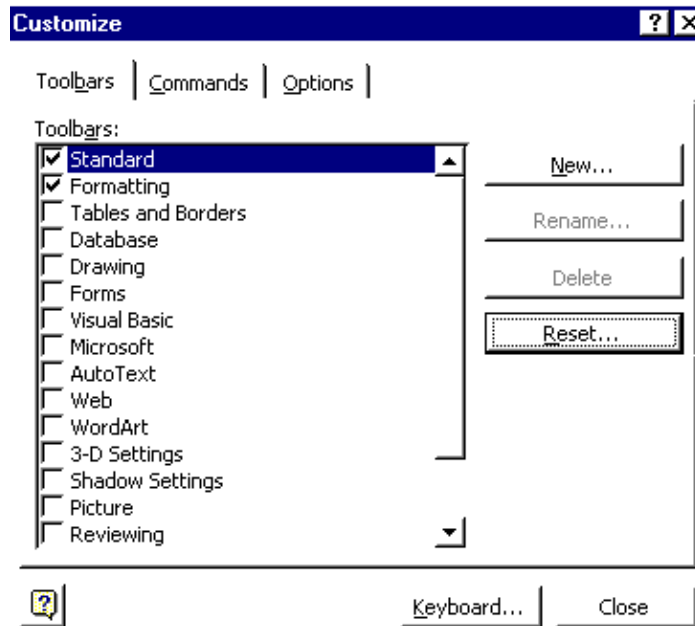
Trong cửa sổ Customize chọn chức năng Command, kéo thanh cuộn phía dưới để tìm chức năng Fonts và bấm đon vào đó phía bên phải sẽ xuất hiện toàn bộ các font chữ có trong Word. Bấm rê một font nào đó lên thanh công cụ, tiếp đó bấm vào Modify Selection để thay đổi chữ hiện trong nút và chọn biểu tượng cho nút (xem lại phần Macro).

Khi một nút thể hiện một font chữ đã có trên thanh công cụ việc định dạng chữ sẽ trở nên rất đơn giản, chỉ cần bôi đen chữ rồi bấm vào nút font đã có.

**c) Hồi phục trạng thái ngầm định của một thanh công cụ**

Đôi khi vì tò mò hay vô tình bạn phá hỏng một thanh công cụ, để khôi phục lại dạng ngầm định của thanh đó chúng ta có thể làm như sau:

Chọn Tools - Customize (hoặc đưa chuột lên thanh công cụ rồi bấm phím phải sau đó chọn Customize) xuất hiện cửa sổ hình 40.4



Hình 40.4

Chọn chức năng Toolbars khi đó tên các thanh công cụ xuất hiện phía dưới. Bấm đơn vào dòng chữ tên thanh công cụ để nó nằm trên nền màu xanh. (Nhớ là không bấm vào hình vuông bên trái) sau đó bấm đơn vào nút Reset phía bên phải cửa sổ, lúc này xuất hiện cửa sổ nhỏ. Trong cửa sổ Reset Toolbars chọn OK, thanh công cụ đã chọn sẽ được khôi phục lại nguyên dạng ngầm định mà Word đã thiết kế.

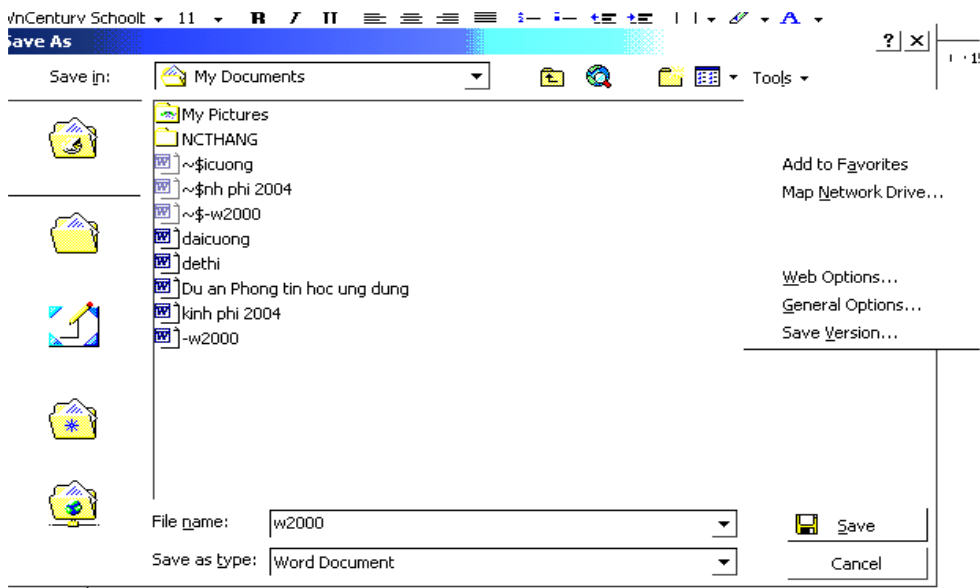
#### 9.4 - Bảo mật văn bản

Những văn bản quan trọng không thể để cho người khác tự động mở ra cần được bảo vệ bằng mật khẩu. Bảo mật văn bản có hai cấp:

- \* Cho phép đọc nhưng không được sửa chữa
- \* Không cho phép đọc

Thao tác tiến hành:

- Chọn File - Save As xuất hiện cửa sổ hình 41.4



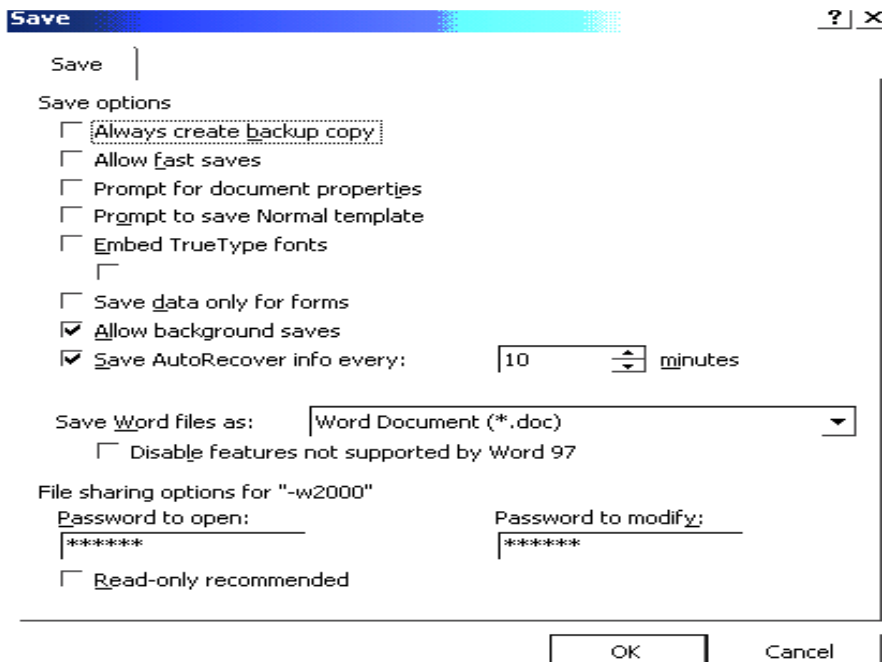
Hình 41.4

- Trong cửa sổ Save As chọn Tools - chọn tiếp General Options, sẽ xuất hiện cửa sổ hình 42.4

Mục Password to Open: Mật khẩu cho phép mở văn bản, chúng ta tùy chọn một mật khẩu nào đó. Nên chọn khoảng 5-6 ký tự và cần lưu mật khẩu này vào sổ để đề phòng sau một thời gian có thể bị quên. Với mật khẩu này chúng ta có thể mở văn bản ra xem nhưng không thể thêm bớt gì vào văn bản.

Mục Password to Modify: Mật khẩu cho phép sửa chữa văn bản, có thể chọn mật khẩu giống như Mục Password to Open hoặc chọn khác đi. Chọn xong bấm OK máy sẽ xuất hiện cửa sổ mới yêu cầu nhắc lại mật khẩu một lần nữa. Gõ lại đúng các mật khẩu đã chọn. Bấm OK thoát ra.

Sau khi đã chọn mật khẩu xong chọn Save để ghi lại văn bản kèm theo mật khẩu. Kể từ nay muốn mở văn bản ra xem chúng ta phải khai báo mật khẩu.



Hình 42.4

## Bài tập Chương IV

**Bài 1:** Soạn thảo văn bản và kẻ bảng dưới đây:

**CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM**  
*Độc lập - Tự do - Hạnh phúc*

---

### GIẤY CAM ĐOAN BẢO VỆ MÔI TRƯỜNG ĐIỆN NƯỚC

*Kính gửi :* .....

*Tên tôi là :* .....

*Công tác tại :* .....

*Nơi ở :* .....

Tôi làm giấy này xin cam đoan trong quá trình cải tạo, nâng cấp căn hộ của gia đình, không làm điều gì ảnh hưởng tới hệ thống điện, nước chung của cả khu vực.

Tôi xin cam đoan sẽ đảm bảo vệ sinh môi trường, không để vật liệu ảnh hưởng đến đường đi và quang cảnh chung của cả khu. Trong quá trình xây dựng không làm ảnh hưởng đến móng, tường nhà của 2 hộ bên cạnh, ***nếu xảy ra điều gì tôi hoàn toàn chịu trách nhiệm***

*Hà nội, ngày ... tháng ... năm .....*

*Người viết đơn*

## BẢNG ĐIỂM THI HỌC KỲ II NĂM HỌC 2003 - 2004

*Lớp 12 B trường PTTH Trần Hưng Đạo*

| Stt | Họ và tên    | Ngày sinh | Điểm thi        |                  |                | Hạng kiểm  |
|-----|--------------|-----------|-----------------|------------------|----------------|------------|
|     |              |           | <i>Điểm Tin</i> | <i>Điểm Toán</i> | <i>Điểm Lý</i> |            |
| 1   | Lê Thu An    | 20-10-83  | 7               | 5                | 5              | <b>Tốt</b> |
| 2   | Trần Thị Hà  | 15-01-82  | 8               | 6                | 7              | <b>Tốt</b> |
| 3   | Đỗ Văn Minh  | 22-11-84  | 9               | 7                | 8              | <b>Tốt</b> |
| 4   | Ngô Thị Thái | 18-03-83  | 5               | 6                | 9              | <b>Khá</b> |
| 5   | Lê Văn Thu   | 01-10-84  | 6               | 8                | 6              | <b>Tốt</b> |

**Bài 2:** Soạn thảo văn bản và kê bảng dưới đây:

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM  
*Độc lập - Tự do - Hạnh phúc*  
----- o O o -----

## ĐỀ CƯƠNG NGHIỆM CỨU KHOA HỌC CẤP TRƯỜNG

**Tên đề tài :** Một số chương trình quản lý hệ thống thông tin đất

**Chủ nhiệm đề tài:** .....

**Những người tham gia:** .....

### 1- Mục đích:

Xây dựng chương trình để quản lý các thông tin đất một cách tự động là *nhệm vụ quan trọng của một kỹ sư ngành Quản lý đất đai trong thời kỳ hiện nay*. Đối với sinh viên ngành Quản lý đất đai nắm được cách tổ chức các cơ sở dữ liệu và cách tổ chức các chương trình về thông tin đất có hệ thống là việc làm vô cùng cần thiết. Điều đó củng cố các kiến thức về Tin học và Chuyên môn, gắn chặt lý thuyết đã học với yêu cầu thực tế trong lĩnh vực chuyên môn của mình.

Xây dựng các chương trình quản lý hệ thống thông tin đất cũng là đáp ứng được yêu cầu đòi hỏi của thực tế trong chuyên ngành Quản lý đất đai hiện nay.

Xuất phát từ các vấn đề trên chúng tôi tiến hành đề tài : “ *Một số chương trình quản lý hệ thống thông tin đất* “.

## BẢNG DANH SÁCH LÊN LƯƠNG NĂM 2000

*Trường Đại học Nông nghiệp I – Hà nội*

| STT | HỌ VÀ TÊN     | ĐƠN VỊ   | BẠC LƯƠNG |            | GHI CHÚ             |
|-----|---------------|----------|-----------|------------|---------------------|
|     |               |          | <i>Cũ</i> | <i>Mới</i> |                     |
| 1   | Trần Ngọc Anh | Phòng HC | 3.37      | 3.62       |                     |
| 2   | Lê Văn Khanh  | Phòng ĐT | 2.95      | 3.37       |                     |
| 3   | Bùi Ngọc Bình | Phòng ĐT | 4.12      | 4.42       | <i>Phó phòng</i>    |
| 4   | Ngô Bình Minh | Phòng QT | 5.05      | 5.37       | <i>Trưởng phòng</i> |
| 5   | Đỗ Minh Thanh | Phòng HC | 2.90      | 3.24       |                     |

**Bài 3:** Soạn thảo văn bản và kê bảng dưới đây:

**TRƯỜNG ĐẠI HỌC NÔNG NGHIỆP I HÀ NỘI**      **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  
Phòng HCKH-VT      **Độc lập - Tự do - Hạnh phúc**  
Số :      /CV-HC  
V/v làm danh bạ điện thoại

*Hà nội, ngày 25 tháng 11 năm 2000*

**Kính gửi : Các đơn vị trong toàn trường**

Trong hai năm qua cuốn danh bạ điện thoại của trường đã góp phần không nhỏ vào sự phát triển thông tin liên lạc của Nhà trường. Hiện nay cuốn danh bạ đó không cung cấp đầy đủ số máy của các cá nhân và tập thể trong trường.

Vì vậy, phòng HCKH-VT sẽ phát hành cuốn "**Danh bạ điện thoại của Trường Đại học Nông nghiệp I - năm 2001**". Để công việc hoàn thành trước tết dương lịch, kính đề nghị các cá nhân, tập thể cung cấp đầy đủ và chính xác số điện thoại mới của mình cho phòng HCKH-VT trước ngày 05/12/1999.

Chúng tôi mong nhận được sự cộng tác chặt chẽ của các đơn vị và cá nhân trong toàn trường.

*Xin trân trọng cảm ơn.*

**Trưởng phòng  
Hành Chính kế hoạch - Vật tư**

### **BẢNG THỐNG KÊ THUÊ NĂM 2000**

| STT | Tên hợp tác xã | Quý 1    |          |          | Quý 2    |          |          |
|-----|----------------|----------|----------|----------|----------|----------|----------|
|     |                | Tháng 1  | Tháng 2  | Tháng 3  | Tháng 4  | Tháng 5  | Tháng 6  |
| 1   | Bát tràng      | 50000000 | 60000000 | 50000000 | 50000000 | 40000000 | 50000000 |
| 2   | Cổ bi          | 35000000 | 55000000 | 35000000 | 35000000 | 55000000 | 75000000 |
| 3   | Dương xá       | 22000000 | 42000000 | 22000000 | 22000000 | 32000000 | 25000000 |
| 4   | Đa tớn         | 20000000 | 50000000 | 20000000 | 20000000 | 50000000 | 60000000 |
| 5   | Trâu quỳ       | 40000000 | 80000000 | 40000000 | 40000000 | 20000000 | 50000000 |



**Bài 4:** Soạn thảo văn bản và kê bảng dưới đây:

**CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM**  
*Độc lập - Tự do - Hạnh phúc*

**ĐƠN XIN HỢP ĐỒNG**

**Kính gửi :** .....

Tên tôi là: ..... Ngày sinh: .....

Chỗ ở hiện nay: .....

Trình độ văn hoá: .....

Hiện nay Bộ môn do thiếu cán bộ kỹ thuật nên đề nghị khoa và phòng Tổ chức cán bộ cho tôi được ký hợp đồng làm việc tại bộ môn từ 1-1-2000 đến 1-1-2001 .

Nếu được thu nhận *tôi xin cam đoan tuyệt đối chấp hành nội qui học tập và lao động*. Tuyệt đối phục tùng sự phân công học tập, bố trí của tổ chức đơn vị sử dụng. Nếu vi phạm nội qui, kỷ luật hoặc tự ý thôi, bỏ học, bỏ việc phải bồi thường hoàn toàn học phí, những tổn thất đã gây ra theo quy định của Nhà nước và của nhà trường đơn vị công tác.

**Kính mong các cấp giải quyết.**

**Hà Nội, ngày tháng năm**

*Người làm đơn*

**Thời khoá biểu năm học 2000-2001**

| THỨ 2 |       | THỨ 3 |       | THỨ 4 |       | THỨ 5 |       | THỨ 6 |       |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Sáng  | Chiều | Sáng  | Chiều | Sáng  | Chiều | Sáng  | Chiều | Sáng  | Chiều |
| Toán  | Anh   | Triết | Sinh  | Tin   | Anh   | Triết | Luật  | Toán  | Anh   |
| Toán  | Anh   | Triết | Sinh  | Tin   | Anh   | Triết | Luật  | Toán  | Anh   |
| Toán  | Anh   | Triết | Sinh  | Tin   | Anh   | Triết | Luật  | Toán  | Anh   |
| Tin   |       | Luật  |       | Toán  |       | Sinh  |       | Tin   |       |
| Tin   |       | Luật  |       | Toán  |       | Sinh  |       | Tin   |       |
| Tin   |       | Luật  |       | Toán  |       | Sinh  |       | Tin   |       |

**Bài 5:** Soạn thảo văn bản và kê bảng dưới đây:

**ĐỀ THI CHẤT LƯỢNG MÔN TOÁN**  
(Thời gian 90 phút)

**Câu 1:** Cho hàm số :  $y = x^3 + 3x^2 + (m+1)x + 4m$  (1)

1/ **Khảo sát và vẽ đồ thị của (1) khi  $m = -1$**

2/ Với giá trị nào của  $m$  hàm (1) đồng biến trên khoảng  $(-1, 1)$

**Câu 2:** Trong mặt phẳng tọa độ Oxy cho 2 điểm  $F_1(-3,0)$ ;  $F_2(3,0)$

1/ Viết phương trình chính tắc của elíp nhận  $F_1$ ;  $F_2$  là các tiêu điểm và trục lớn có độ dài bằng 10

2/ Tìm trên elíp nói trên những điểm  $M(x_0, y_0)$  sao cho khoảng cách từ đó tới đường thẳng cho bởi phương trình :  **$x + y = 6$  là nhỏ nhất.**

*Hà nội, ngày 20 tháng 4 năm 2000*

**Chủ tịch hội đồng thi**

**BẢNG THỐNG KÊ HÀNG NHẬP**

| Số tt | Tên hàng | Kho A     |          |         | Kho B     |          |         |
|-------|----------|-----------|----------|---------|-----------|----------|---------|
|       |          | Ngày nhập | Số lượng | Đơn giá | Ngày nhập | Số lượng | Đơn giá |
| 1     | Vải hoa  | 1/1/98    | 100      | 10000   | 12/1/98   | 200      | 10000   |
| 2     | Vải lụa  | 1/1/98    | 200      | 15000   | 11/1/98   | 300      | 14000   |
| 3     | Khăn     | 2/1/98    | 50       | 7000    | 21/1/98   | 150      | 8000    |
| 4     | Bột giặt | 3/1/98    | 100      | 6500    | 30/1/98   | 400      | 6000    |
| 5     | Bánh     | 5/1/98    | 120      | 6000    | 11/2/98   | 220      | 5000    |

## CHƯƠNG V: INTERNET VÀ CÁCH SỬ DỤNG INTERNET

Internet là mạng thông tin toàn cầu, nó đóng vai trò quan trọng trong mọi lĩnh vực của xã hội. Hiện nay Internet được sử dụng rộng rãi trên thế giới cũng như ở nước ta. Mọi người cần phải hiểu và biết sử dụng Internet để phục vụ cho chuyên môn của mình.

Chương này cung cấp các kiến thức cơ bản về mạng Internet và thư điện tử. Nội dung chính bao gồm các vấn đề : Thế nào là mạng Internet, Internet có các chức năng cơ bản nào, cài đặt Internet, trình duyệt Internet Explorer, E-mail và các mô hình hoạt động của E-mail, cách tạo gửi và nhận thư với Yahoo và với Outlook Express.

### 1 - Giới thiệu chung

#### 1.1 - Mạng máy tính

\* Mạng máy tính: là một nhóm các máy tính được nối kết với nhau theo một cách nào đó. Về nhận thức, nó giống như kiểu mạng truyền hình hoặc radio nối kết một nhóm các trạm truyền hình hoặc radio lại với nhau sao cho chúng có thể cùng chia sẻ chương trình. Các mạng truyền hình gửi thông tin tương tự đến mọi trạm vào cùng lúc, trong mạng máy tính, mỗi thông điệp thường được dẫn đến một máy tính cụ thể nào đó. Khác với các mạng truyền hình, các mạng máy tính luôn hai chiều sao cho khi máy tính A gửi thông điệp tới máy tính B thì B có thể trả lời lại cho A.

Một số mạng máy tính bao gồm một máy tính trung tâm và một nhóm các trạm từ xa có thể báo cáo về máy tính trung tâm, ví dụ một máy tính dịch vụ giữ chỗ hàng không trung tâm có hàng ngàn kênh tại các sân bay và đại lý du lịch.

\* Mạng Internet: Là mạng máy tính toàn cầu, hoặc chính xác hơn là mạng của các mạng.

Để các máy tính nối lại và có thể giao tiếp được với nhau, người ta dùng một thiết bị gọi là modem. Modem và chương trình ứng dụng dùng để chuyển đổi từ tín hiệu tương tự (analog) trên đường dây điện thoại máy tính thành tín hiệu số (digital) để đưa vào máy tính. Ngược lại, modem còn chuyển tín hiệu số của máy tính thành tín hiệu tương tự để truyền dẫn trên đường dây điện thoại. Như vậy, một máy tính muốn kết nối và truy nhập thông tin thì phải có một modem nối vào một cổng của máy tính. Về vị trí lắp đặt, có thể chia modem thành hai loại: loại lắp trong máy tính (internal) và lắp ngoài máy tính (external).

\* Xuất xứ của Internet

Tổ tiên của Internet là ARPANET, một dự án do bộ Quốc phòng Hoa Kỳ khởi đầu năm 1969 vừa là một thực nghiệm trong việc kết mạng một cách đáng tin cậy, vừa là một kết nối giữa bộ Quốc phòng và các nhà thần nghiên cứu khoa học và quân sự lại với nhau, bao gồm một số lớn các trường đại học tiến hành các nghiên cứu quân sự được tài trợ. (ARPA là viết tắt của chữ Advanced Research Projects Agency: Cơ quan Các Dự án Nghiên cứu Cao cấp, một bộ phận thuộc bộ Quốc phòng phụ trách việc cấp phát tài trợ.

Kế tiếp là Hội đồng Khoa học Quốc gia Hoa Kỳ (National Science Foundation - NSF) quyết định thiết lập 5 trung tâm siêu máy tính nhằm mục đích nghiên cứu (Một siêu máy tính là một máy tính cực kỳ nhanh với giá rất đắt, khoảng 10 triệu USD mỗi máy). NSF tính toán rằng nên tài trợ cho một ít máy tính, để cho các nhà nghiên cứu trên khắp đất nước sử dụng ARPANET gửi các chương trình của họ đến để được "siêu tính toán" và rồi gửi kết quả trở lại. Qua một thời gian khi các siêu máy tính rõ ràng đã trở nên lỗi thời, NSF đã xây dựng NSFNET, một mạng riêng và nhanh hơn nhiều của mình để nối với các trung tâm siêu tính toán. Sau đó, NSF dần xếp để thiết lập một chuỗi các mạng khu vực nhằm liên kết những

người sử dụng trong từng khu vực với NSFNET nối mọi mạng khu vực. NSFNET hoạt động hiệu quả, từ năm 1990 hệ thống mạng quốc tế nối vào NSFNET, NSF đã được xác lập vững chắc trên Internet.

Các mạng quốc tế liên kết với nhau nhờ vào một chương trình kỹ thuật được gọi là IP (Internet Protocol: Giao thức Internet). NSFNET chỉ cho phép những lưu thông liên quan đến nghiên cứu và giáo dục, do đó những dịch vụ mạng IP độc lập cho phép lưu thông được dẫn từ mạng này sang mạng khác khi cần. Mọi mạng nối bởi IP đều sử dụng IP để giao tiếp nên chúng đều có thể trao đổi các thông điệp với nhau.

Các mạng thương mại nối với các mạng khu vực theo cách tương tự như kết nối của NSFNET và cung ứng kết nối trực tiếp đến khách hàng.

Bên ngoài Hoa Kỳ, các mạng IP xuất hiện tại nhiều nước, do các công ty điện thoại địa phương tài trợ hoặc do một nhà cung ứng quốc gia hoặc khu vực độc lập. Hầu hết trong số họ đều được kết nối trực tiếp hoặc gián tiếp với một mạng nào đó của Hoa Kỳ, có nghĩa là tất cả đều có thể trao đổi lưu thông với nhau.

\* Mạng Internet các chức năng chính sau đây:

- **Thư điện tử:** Đây chắc chắn là dịch vụ được sử dụng rộng rãi nhất - bạn có thể trao đổi thư điện tử với hàng triệu người trên khắp thế giới. Các danh sách thư điện tử cho phép bạn tham gia vào thảo luận nhóm và gặp những người khác trên Net. Các mail server (những chương trình đáp lại các thông điệp thư điện tử) cho phép bạn truy tìm mọi loại thông tin.

- **Truy tìm thông tin:** Internet giúp bạn tìm kiếm thông tin trên mạng. Nhiều máy tính có các file chứa thông tin cho phép lấy miễn phí. Các file đó bao gồm các file văn bản, các file hình ảnh được số hóa và một số lượng lớn các phần mềm từ trò chơi cho đến các hệ thống điều hành. Hàng núi thông tin sẵn có trên Internet và bạn sẽ mục mục "Navigate" (Tìm đường) chỉ đến những phần giúp bạn lưu thông trên mạng.

- **Bảng thông báo điện tử (Bulletin Board):** Một hệ thống tên là USENET là một bảng thông báo điện tử vĩ đại với 40 triệu ký tự các thông điệp bao gồm 2.000 nhóm chủ đề khác nhau được tiếp nối hàng ngày. Các chủ đề bao gồm từ các vấn đề máy tính học búa cho đến những trò giải trí như đua xe đạp, từ những ý kiến tranh luận chính trị không bao giờ kết thúc cho đến những vấn đề bình thường nhất. Nhóm USENET được đọc rộng rãi nhất là các mẫu chuyện vui được chọn lọc, một số trong những chuyện này thực sự buồn cười.

- **Trò chơi và tán gẫu:** Một trò chơi được gọi là MUD (Multi-User Dungeon) có thể dễ dàng thu hút toàn bộ thời gian của bạn - trong đó bạn có thể thi tài với những người chơi khác ở bất cứ nơi nào trên thế giới.

Internet Relay Chat (IRC) là một đường dây theo nhóm, qua đó bạn có thể có những cuộc đàm thoại thú vị với những người sử dụng khác ở khắp mọi nơi. IRC dường như được sử dụng nhiều nhất bởi những học sinh, sinh viên, bạn sẽ không bao giờ biết được ai là người đang nói chuyện với bạn.

## 1.2 - Nhà cung cấp dịch vụ Internet

Để một máy tính cá nhân nối vào được mạng Internet, bạn phải đăng ký thuê bao ở một nhà cung cấp dịch vụ Internet (Internet Service Provider - ISP) và khi này ISP sẽ là vị trí trung gian chuyên tiếp để máy tính của bạn nối với các trang Web hoặc các địa chỉ trên mạng.

Hiện nay có nhiều nhà cung cấp dịch vụ Internet như VNN, FPT, VCD, Phươngnam, Saigonnet, Netnam .... Mỗi một ISP có những khả năng khác nhau. Khi chọn ISP, bạn nên quan tâm đến một số vấn đề sau:

- Khả năng đảm bảo đường truyền

- Giá đăng ký và thuê bao. Hiện nay giá cả của các nhà cung cấp dịch vụ không giống nhau.

- Vị trí của ISP. Đây là một vấn đề cần hết sức quan tâm khi chọn nhà cung cấp dịch vụ Internet. Ví dụ, nếu bạn ở Hà Nội thì nên chọn VNN hoặc FPT vì lúc này khi trả tiền điện thoại, bạn chỉ phải thanh toán theo giá điện thoại nội hạt. Nếu bạn chọn một ISP khác và không có dịch vụ kết nối mạng thì tiền điện thoại sẽ phải thanh toán từ vị trí gọi đến địa điểm đặt máy chủ của mạng. Bạn cũng cần lưu ý rằng, khi sử dụng Internet có ba loại tiền cần thanh toán:

\* Tiền điện thoại: Trong thời gian sử dụng Internet, tiền điện thoại được tính như khi gọi điện bình thường. .

\* Tiền truy cập mạng: Số tiền này được tính theo thời gian truy cập và thời điểm truy cập. Hầu hết các ISP đều tính giá tiền cho thời gian truy cập vào ban đêm và các ngày nghỉ thấp hơn ban ngày. Số tiền này bạn phải thanh toán cho ISP. Các nhà cung cấp dịch vụ Internet khác nhau số tiền này cũng khác nhau.

\* Tiền lấy các thông tin trên mạng: Hầu hết các trang Web trong nước và nhiều trang Web nước ngoài cung cấp các thông tin miễn phí, tức là bạn không phải trả tiền cho các thông tin hoặc các ứng dụng trên mạng. Tuy nhiên, cũng có một số bạn phải trả tiền.

### 1.3 - Kết nối Internet

\* Điều kiện gia nhập mạng Internet: Để gia nhập mạng Internet bạn phải có các các mục sau:

- Máy vi tính
- Đường điện thoại
- Modem

- Tài khoản đăng ký truy nhập với nhà cung cấp ( ISP): Để có thể truy cập Internet, bạn cần ACcount hay tài khoản Internet, bao gồm:

+ User name là tên đăng ký với ISP, tên này không thể thay đổi được. Ví dụ User name máy của bộ môn Tin học là: bmthnn1

+ Password là mật khẩu , một lần kết nối ta phải gõ vào mật khẩu, mật khẩu có thể thay đổi được.

\* Tạo biểu tượng kết nối Internet:

Từ màn hình chính của Windows chọn các mục sau:

My Computer / Control panel / Network and Dial Up Networking / Make New Connection

Sau đó sẽ xuất hiện hộp thoại ta sẽ khai báo các mục sau:

- Gõ tên của biểu tượng vào hộp Type a name for the computer you are dialing
- Chọn **Next**
- Gõ mã vùng vào hộp **Area code**
- Gõ số điện thoại của nhà cung cấp vào **Telephone Number**
- Chọn tên nước trong hộp **Country code**
- Chọn **Next**
- Chọn **Finish**

\* Cách kết nối

Nháy đúp chuột vào biểu tượng kết nối. Khi đó sẽ xuất hiện cửa sổ Connect to, ta khai báo các mục sau:

- **User name:** Gõ tên đăng ký với ISP
- **Password:** Gõ mật khẩu
- **Phone number:** Số điện thoại của ISP

- Chọn **connect** thực hiện kết nối.

Sau khi kết nối được, biểu tượng kết nối có 2 máy tính liên kết nằm ở góc phải, phía dưới màn hình, trên thanh Taskbar của Windows.

\* Huỷ kết nối

- Nháy chuột vào biểu tượng kết nối trên thanh **Taskbar**

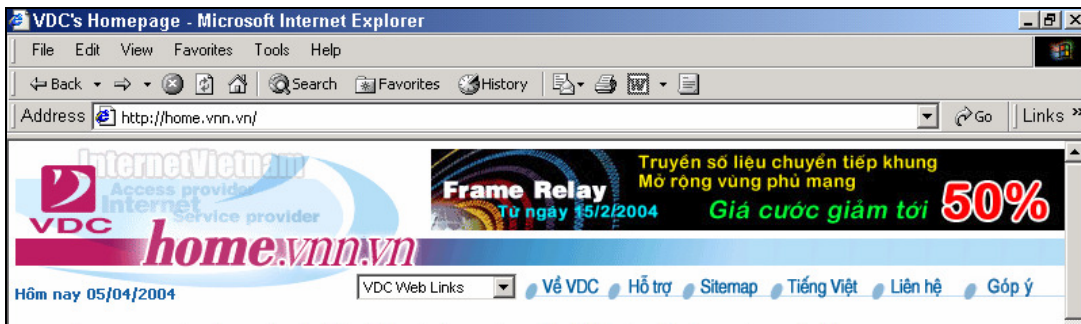
- Chọn **Disconnect**

## 2. Sử dụng trình duyệt Internet Explorer (IE)

### 2.1 - Khởi động

- Kích chuột vào START/ chọn PROGRAM/ chọn INTERNET EXPLORER

- Màn hình IE xuất hiện như sau:



Hình 1.6

Lần lượt từ trên xuống dưới là các phần sau:

+ Dòng đầu tiên là thanh tiêu đề (Title bar): Hiện ra tên trang Web, tên trình duyệt Web.

+ Dòng 2 là thanh Menu (Menu bar: thanh thực đơn): Bao gồm các mục chính của trình duyệt Web.

+ Dòng 3 là thanh công cụ (Tool bar): Bao gồm các biểu tượng của các lệnh hay dùng, muốn chọn lệnh nào ta chỉ việc kích chuột vào biểu tượng của nó.

+ Dòng 4 là thanh địa (Address): Hiện ra địa chỉ của trang Web đang được truy cập. Ta có thể gõ trực tiếp địa chỉ vào đây hoặc nhấn vào mũi tên bên phải mục này để chọn địa chỉ đã lưu.

+ Phần vùng hiển thị: Là phần hiển thị nội dung trang Web đang truy cập, phần này chiếm nhiều dòng.

+ Thanh trạng thái : Dòng này ở cuối cùng, hiện ra các thông tin hiện thời.

### 2.2 - Ra khỏi Internet Explorer

- Chọn mục **File**

- Chọn **Close**

### 2.3 - Tìm kiếm thông tin

\* Cách tìm kiếm thông tin nếu có địa chỉ chính xác:

- Gõ địa chỉ cần truy nhập vào thanh địa chỉ (thanh Address)

- Enter (Hoặc kích chuột vào GO)

\* Cách tìm kiếm thông tin nếu không có địa chỉ chính xác:

- Sử dụng nút Search trên thanh công cụ

## 2.4 - Cách lưu trữ các địa chỉ và cách tổ chức lại các địa chỉ

\* Cách lưu các địa chỉ ưa thích vào FAVORITES:

Những địa chỉ nào thường xuyên truy cập, mỗi lần truy cập ta phải gõ lại địa chỉ, thay vì việc đó ta lưu lại các địa chỉ để mỗi lần truy cập vào trang web đó chỉ cần kích chuột chọn địa chỉ ở bên trong menu Favorites.

Cách lưu:

- Mở trang WEB cần lưu
- Vào menu Favorites
- Chọn Add to Favorites
- OK

\* Tổ chức lại các địa chỉ:

Mỗi lần lưu lại địa chỉ thì menu Favorites sẽ dài thêm. Do vậy để tiện sử dụng ta tổ chức lại menu Favorites như sau:

- Vào menu Favorites/ chọn Organize Favorites/ màn hình xuất hiện:  
Trong đó:
  - + Create Folder: Tạo một Folder mới
  - + Move to Folder: Di chuyển địa chỉ
  - + Rename: Đổi tên Folder
  - + Delete: Xoá Folder hoặc xoá địa chỉ

## 2.5 - Một số tùy chọn của Internet:

\* Thay đổi địa chỉ của trang khởi tạo:

- Vào menu Tool chọn Internet Option
- Chọn General
- Gõ địa chỉ cần đặt vào hộp Address hoặc có thể chọn một trong các nút trong mục đó là: Lấy địa chỉ của trang hiện tại, lấy địa chỉ mặc định, lấy trang trắng
- Chọn Apply
- Chọn OK

\* Thay đổi số ngày lưu trong trang web:

- Vào menu Tool chọn Internet Option
- Gõ vào số ngày trong hộp Days to keep page in history
- OK

\* Xoá các địa chỉ lưu trong hộp History:

- Xoá toàn bộ các trang:
  - + Vào menu Tool chọn Internet Option
  - + Chọn Clear History
  - + OK
- Xoá từng trang:
  - + Mở History
  - + Nháy chuột phải vào trang cần xoá
  - + Chọn Delete

\* Thay đổi cỡ chữ:

- Vào menu View

- Chọn Text size: Largest: Cỡ chữ lớn nhất, Larger: Cỡ chữ lớn, Medium: Cỡ chữ trung bình, Small: Cỡ chữ nhỏ, Smallest: Cỡ chữ nhỏ nhất

## 2.6 - Lưu trang web vào máy tính

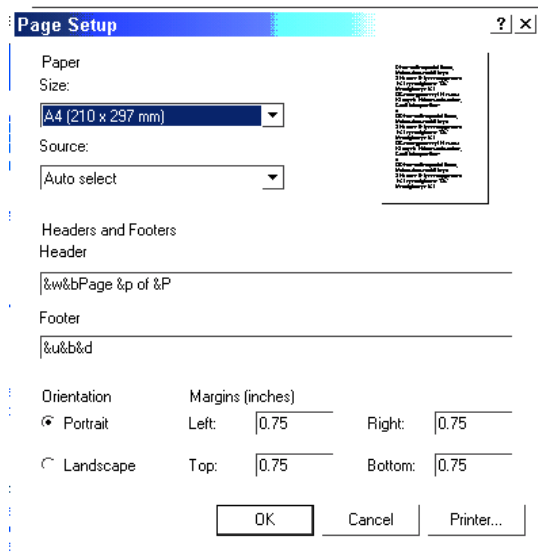
- Vào menu File
- Chọn Save as
- Chọn vị trí cần lưu
- Chọn Save

## 2.7 - Lưu ảnh

- Nháy chuột phải vào ảnh
- Chọn Save Picture As
- Chọn vị trí cần lưu
- Chọn Save

## 2.8 - In trang web

- \* Thiết kế một trang in
  - Chọn File
  - Chọn Page Setup. Màn hình xuất hiện



Hình 2.6

- Size : khổ giấy
- Source: khay giấy
- Header: Tiêu đề đầu trang
- Footer: Tiêu đề cuối trang
- Orientation : Hướng in, Portrait là theo chiều dọc, Landscape là theo chiều ngang.
- Margins : Lê trang in, Left lê trái, Right lê phải, Top là phía trên, Bottom là phía dưới.

- \* Xác định Header and footer

Internet Explore có thể in các tiêu đề đầu trang và tiêu đề chân trang theo các cách sau:

- Left header - Tiêu đề nằm ở đầu trang. Theo định sẵn vùng này chứa Title của tài liệu



- Center header - Tiêu đề nằm ở giữa và trên đầu trang. Theo định sẵn thì vùng này trắng.
- Right header - Tiêu đề nằm ở bên phải ở trên đầu trang. Theo định sẵn thì vùng này chứa số trang của tài liệu
- Left footer - Tiêu đề sẽ nằm về bên trái ở phía dưới của trang. Theo định sẵn thì vùng này chứa ngày tháng của tài liệu.
- Center footer - Tiêu đề sẽ nằm ở giữa và phía dưới của trang. Theo định sẵn thì vùng này trắng.
- Right footer - Tiêu đề sẽ nằm ở chính giữa ở phía dưới trang. Theo định sẵn thì vùng này chứa thời gian in tài liệu.

Bạn có thể sử dụng các mã đặc biệt để lấy thông tin hiện nay từ tài liệu (chẳng hạn như số trang) hay từ hệ thống (chẳng hạn như ngày, giờ). Các mã sau đây sẽ bắt đầu bằng dấu & theo sau là một ký tự. Chú ý rằng chữ hoa hay chữ thường rất quan trọng. Bằng cách sử dụng các mã này thì bạn có thể in các tiêu đề theo ý mình:

| Nhập vào | Sẽ in ra                                                                   |
|----------|----------------------------------------------------------------------------|
| &w       | Tựa đề của tài liệu                                                        |
| &u       | Địa chỉ của trang                                                          |
| &d       | Ngày hiện nay ở dạng tắt xác định trong Regional Setting/ Control Panel    |
| &D       | Ngày hiện nay ở dạng đầy đủ xác định trong Regional Setting/ Control Panel |
| &t       | Giờ hiện nay ở dạng xác định trong Regional Setting/ Control Panel         |
| &T       | Giờ hiện nay theo dạng 24 giờ                                              |
| &p       | Số trang                                                                   |
| &P       | Tổng số trang                                                              |
| &&       | Một dấu &                                                                  |
| &b       | Vị trí xác định                                                            |

Dấu cuối cùng (&b) cho phép bạn xác định mục đã cho sẽ in ở header hay footer. Sau đây là cách thức làm việc của nó:

- Nếu bạn sử dụng mã này chỉ một lần thì bên trái của &b sẽ in ở bên trái còn phần bên phải sẽ in ở bên phải.

Chẳng hạn, &w&b&p sẽ in tựa đề của tài liệu ở bên trái còn số trang sẽ in ở bên phải.

- Nếu bạn sử dụng mã này hai lần thì phần bên trái của &b thứ nhất sẽ in ở bên trái, phần ở giữa hai &b sẽ in ở chính giữa và phần bên phải của &b sẽ in ở bên phải.

Chẳng hạn, &d&b&p&b&t sẽ in ngày hiện nay ở bên trái, trang chính giữa và in thời gian hiện nay ở bên phải.

- Bạn cũng có thể kết hợp văn bản và mã. Chẳng hạn, nếu bạn gõ Page %p of &p thì Internet Explorer sẽ hiện trang hiện nay trên tổng số trang: Page 7 of 10.

\* In tài liệu

- Chọn File, chọn Print hoặc ấn Ctrl + P.

### 3 - Thư điện tử ( E- mail)

#### 3.1 - Giới thiệu E-mail

E-mail - viết tắt của Electronic Mail (thư điện tử) là một dịch vụ được triển khai trên các mạng máy tính cho phép người sử dụng có thể gửi thư cho nhau.

ở đây “thu” là một tệp dạng text, hành động chuyển thư được thực hiện theo phương thức truyền gửi các tệp này trên các đường truyền dẫn của mạng. Trong nhiều trường hợp thư còn có từ đồng nghĩa là thông điệp (message), đúng ra thông điệp cần được hiểu là một thư ngắn không vượt quá một câu.

E- mail là dịch vụ cơ bản nhất và phổ thông nhất trên mạng, là nền tảng để xây dựng một văn phòng không giấy (paperless office).

### 3.2 - Các mô hình hoạt động của dịch vụ E-mail:

- Mô hình thông điệp trực tiếp: Theo mô hình này, các thông điệp được gửi trực tiếp, ngay lập tức tới các máy đang hoạt động trên mạng LAN.

- Mô hình hộp thư lưu: Là mô hình khách – chủ của dịch vụ thư điện tử, theo đó chương trình thực hiện dịch vụ này được viết thành hai phần chương trình chạy độc lập tương đối:

+ Chương trình phía Client (khách) chạy trên các máy trạm làm việc của mạng và phần phía Server (chủ) chạy trên máy chủ của mạng. Người dùng soạn thảo thư, ấn định địa chỉ nhận và “ra lệnh” gửi thông qua chương trình client về máy chủ của mạng.

+ Chương trình Server đón nhận các thư gửi từ Client, lưu trữ vào các hộp thư lưu.

+ Mỗi người dùng cũng nhận và đọc thư của mình bằng cách dùng chương trình phía client duyệt danh sách các thư gửi cho mình và đọc thư quan tâm.

+ Chương trình Server cũng có thể phát tín hiệu báo có thư cho những người đang làm việc trên mạng.

- Mô hình Internet Mail: Là mô hình liên kết các hộp thư lưu trên Internet, mỗi hộp thư lưu cho một vùng. Nếu tất cả các liên kết đang được thực hiện, thư sẽ được chuyển đến hộp thư lưu đích trong vài giây, trường hợp ngược lại nó sẽ được chuyển bước qua từng hộp thư lưu và các đường liên kết đang thực hiện để đến đích cuối cùng.

### 3.3 - Tạo hòm thư miễn phí với Yahoo

Có rất nhiều địa chỉ trang web ta có thể tạo được một địa chỉ thư miễn phí như Yahoo.com, hotmail.com, vietfun.com, ...

#### \* Cách tạo hòm thư miễn phí với Yahoo:

- Gõ địa chỉ [www.yahoo.com](http://www.yahoo.com) (hoặc gõ địa chỉ [mail.yahoo.com](mailto:mail.yahoo.com))

- Chọn Email hoặc Chekmail

- Chọn Sign Up Now

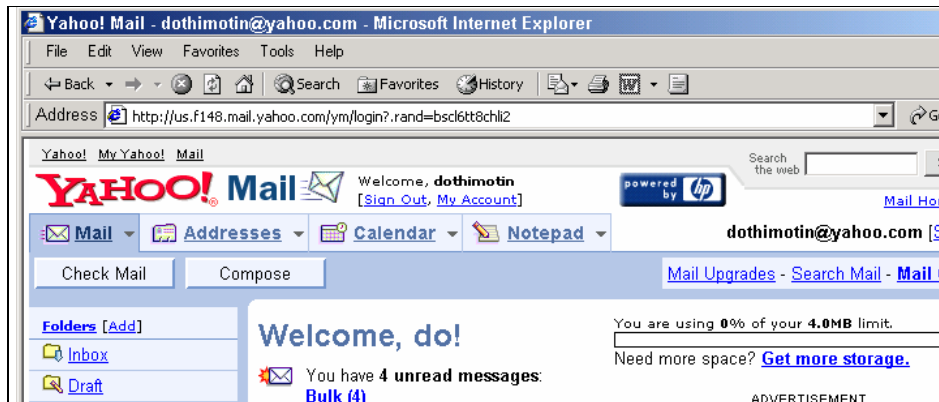
Kết quả xuất hiện cửa sổ ta phải khai báo các mục sau:

- |                       |                                                              |
|-----------------------|--------------------------------------------------------------|
| +YahooID:             | Gõ tên hòm thư cần tạo (Không dấu, không cách)               |
| + Password:           | Gõ mật khẩu (ít nhất 6 ký tự, phải tắt chế độ gõ tiếng việt) |
| - Retype password:    | Gõ lại mật khẩu                                              |
| - Security Question:  | Chọn câu hỏi tùy chọn (chọn bất kỳ)                          |
| - Your answer:        | Gõ vào câu trả lời (ít nhất có 4 ký tự)                      |
| - Birthday:           | Gõ ngày tháng năm sinh                                       |
| - Gender:             | Chọn giới tính                                               |
| - Fist/Last name:     | Gõ họ và tên đệm                                             |
| - Languge:            | Chọn ngôn ngữ                                                |
| - Zip Code:           | Gõ mã nước                                                   |
| - Industry:           | Chọn nghề nghiệp (chọn bất kỳ)                               |
| - Title:              | Chọn chức vụ (chọn bất kỳ)                                   |
| - Enter the word:     | Gõ từ hiển thị ở hộp dưới                                    |
| Chọn Submit this form | Đệ trình Form                                                |

### 3.4 - Gửi và nhận thư với Yahoo

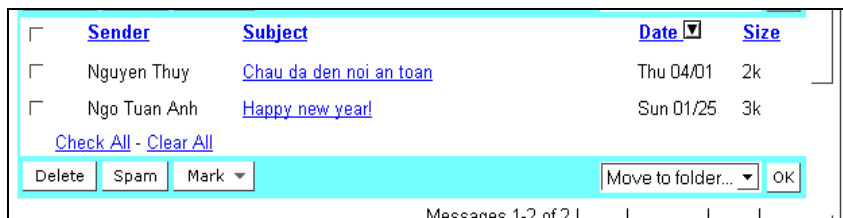
- Gõ địa chỉ www.yahoo.com ( hoặc gõ địa chỉ mail.yahoo.com)
- Chọn Email hoặc Mail
- Gõ địa chỉ thư vào hộp Yahoo ID
- Gõ mật khẩu vào Password
- Chọn Sign In

Kết quả xuất hiện cửa sổ:



Hình 3.6

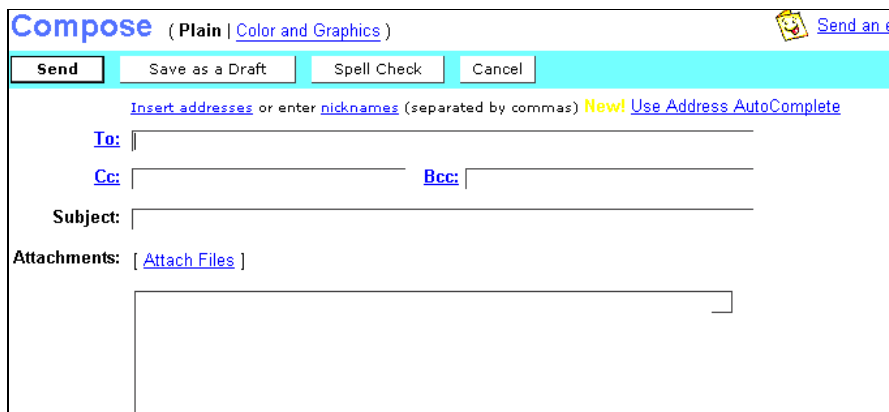
\* Chọn **Check mail**: Để nhận thư. Kết quả có cửa sổ sau:



Hình 4.6

- Sender : Người gửi
- Subject : Tiêu đề thư
- Date : Ngày gửi
- Size : Kích cỡ file
- Muốn đọc thư nào thì nháy chuột vào thư đó.
- Muốn xóa thư thì chọn vào ô vuông bên trái thư, chọn nút Delete

\* **Chọn Compose**: Để gửi thư. Kết quả có cửa sổ sau:



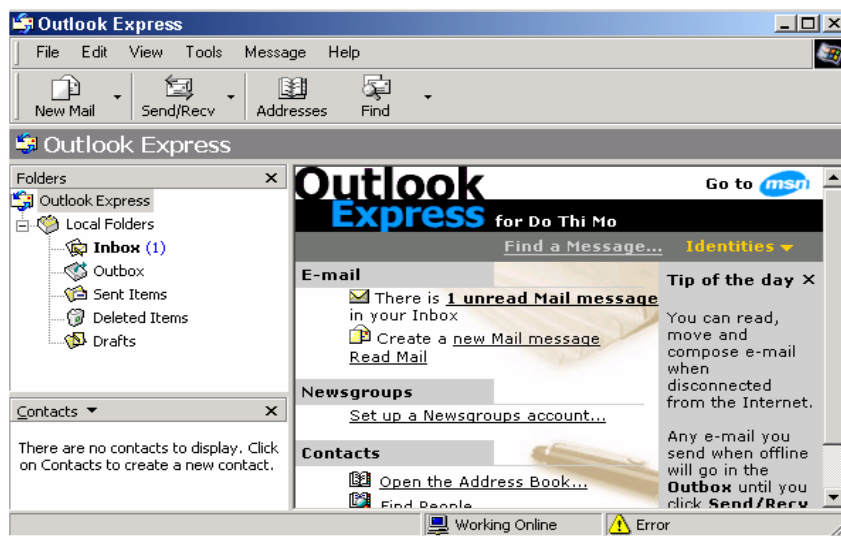
Hình 5.6

- To : Địa chỉ người nhận
- Cc : Địa chỉ đồng gửi tới
- Bcc : Địa chỉ đồng gửi tới
- Subject : Tiêu đề thư
- Attachments [Attachments] : Gửi kèm file
- Phần cửa sổ trắng phía chứa nội dung thư ta gõ vào.
- Chọn nút **Send** để gửi thư đi

### 3.5 - Gửi nhận thư bằng Outlook Express

#### \* **Khởi động Outlook Express**

- Kích chuột vào Start
  - Chọn Program
  - Chọn Outlook Express
- Kết quả có cửa sổ sau:



Hình 6.6

- Inbox : Nơi lưu các thư được tải xuống và nó sẽ ở đó đến khi bạn chuyển nó đi nơi khác.
- Outbox : Nơi lưu tạm thời các thư chưa được chuyển
- Deleted Items : Lưu các thư được xóa ở Inbox (Tương tự như Recycle Bin, nếu muốn xóa hẳn các thư nằm trong thư mục này thì bạn phải: Nháy chuột phải trên biểu tượng của thư mục và chọn Empty Folder)
- Sent Items : Lưu các bản sao của các thư đã gửi đi
- Drafts : Nếu bạn soạn một thư và cần phải thực hiện một số việc khác trước khi hoàn tất bức thư thì chỉ cần đóng bức thư và lưu trữ nó lại. Outlook Express sẽ lưu bức thư này trong thư mục Drafts (bản nháp).

#### \* **Khai báo Accounts**

- Vào menu Tool/ Chọn Accounts/ Chọn Mail/ Chọn Add/ Chọn Mail
- Gõ tên vào hộp Display name
- Chọn Next
- Khai báo máy chủ nhận thư vào hộp Incoming mail :
  - + Thuê bao VNN: mail.hn.vnn.vn

- + Thuê bao FPT : imail.fpt.vn
- + Thẻ VNN : fmail.hn.vnn
- + Thẻ FPT : pop3.cardvn.net

- Khai báo máy chủ gửi thư vào hộp Outgoing mail :

- + Thuê bao VNN: mail.hn.vnn.vn
- + Thuê bao FPT : omail.fpt.vn
- + Thẻ VNN : fmail.hn.vnn
- + Thẻ FPT : smtp.cardvn.net

- Chọn Next
- Gõ tên của Account vào hộp Account name
- Gõ mật khẩu vào Password
- Chọn Next
- Chọn Finish
- Chọn Close

**\* Gửi thư:**

Bạn có thể gửi thư theo ba cách:

- Soạn thư mới
- Hồi đáp thư cũ
- Chuyển một bức thư đã nhận

**\* Soạn một thư mới:**

- Chọn New Mail ( hoặc chọn menu Message/ New Message ). Cửa sổ New Message xuất hiện. Các thao tác tương tự như gửi thư tại yahoo.

**\* Trả lời một bức thư (Hồi đáp thư cũ):**

- Chọn Inbox
- Chọn thư đọc
- Có hai cách trả lời như sau:

+ Reply : Nếu bạn chọn lựa chọn này thì thư hồi đáp của bạn chỉ tới có một địa chỉ e – mail trong dòng From của phần tiêu đề. Không có bản sao nào được gửi tới một người nào trong phần danh sách được xác định ở mục Cc: Của bức thư ban đầu

+ Reply All : Nếu bạn chọn lựa chọn này thì bức thư của bạn tới không chỉ tác giả mà còn tới tất cả các mọi người trong danh sách Cc.

**\* Gửi chuyển tiếp thư đã nhận:**

- Chọn thư cần chuyển
- Chọn Forward Message {Hoặc chọn menu Message (hoặc Compose), chọn Forward

hay ấn Ctrl + F }

- Gõ vào địa chỉ cần gửi đi
- Chọn Send

Hoặc có thể chuyển thư dưới hình đính kèm

**\* Nhận thư đến:**

Chọn Inbox/ Chọn Send and Receive/ Nhập mật khẩu (Password) vào , rồi ấn OK

**\* Đọc thư đến:**

Chọn Inbox: Nếu có thư đến, sẽ thấy hiện thị tên người gửi. Bạn chỉ việc nhấn chuột vào tên thư muốn đọc nội dung của nó ở phía dưới.

**\* Tải thư về (Download) máy tính:**

- Chọn thư muốn tải về
- Chọn File/ Save As
- Chọn kiểu file: Mail (.eml), Text (.txt), web (.html).

- Chọn vị trí muốn lưu file đó
- Đặt tên file rồi bấm OK

### 3.6. Download và một số địa chỉ Internet

#### \* Download :

- Là thực hiện tải một phần mềm hoặc một chương trình trên mạng máy tính về máy tính của mình.
- Có thể Download từ các trang cụ thể (nếu có mục Download) hoặc từ các địa chỉ sau:
  - + Microsoft Free Product Downloads (): Đối với những người sử dụng Win dows thì đây là nơi bạn có thể tìm thấy các fonts, trò chơi, các chương trình hỗ trợ cho Internet Explore và Microsoft Office....

+ Download.com và Shareware.com (www.shareware.com): Đây là những nơi cho phép bạn tìm kiếm trên một cơ sở dữ liệu của hơn 190.000 chương trình miễn phí và chia sẻ.

#### \* Một số địa chỉ Internet:

|                                                                          |                                                                            |
|--------------------------------------------------------------------------|----------------------------------------------------------------------------|
| <a href="http://www.hau1.edu.vn">www.hau1.edu.vn</a>                     | Trang chủ của Trường ĐH Nông nghiệp I                                      |
| <a href="http://www.express.net">www.express.net</a>                     | Trang web của Việt nam Thông tấn xã                                        |
| <a href="http://www.ptcenter.net">www.ptcenter.net</a>                   | Trang web của trung tâm tin học PT                                         |
| <a href="http://www.home.vnn.vn">www.home.vnn.vn</a>                     | Trang web của VDC                                                          |
| <a href="http://www.msn.com">www.msn.com</a>                             | Trang web của hãng Microsoft                                               |
| <a href="http://www.fpt.vn">www.fpt.vn</a>                               | Trang của FPT                                                              |
| <a href="http://www.vnn.vn">www.vnn.vn</a>                               | Trang của VASC                                                             |
| <a href="http://www.quochoi.gov.vn">www.quochoi.gov.vn</a>               | Trang của Quốc hội                                                         |
| <a href="http://www.vtv.org.vn">www.vtv.org.vn</a>                       | Trang của Truyền hình Việt Nam                                             |
| <a href="http://www.nxbgd.com.vn">www.nxbgd.com.vn</a>                   | Trang của Nhà xuất bản giáo dục                                            |
| <a href="http://www.ykhoa.com.vn">www.ykhoa.com.vn</a>                   | Trang Y khoa bằng Tiếng Việt                                               |
| <a href="http://www.laodong.com.vn">www.laodong.com.vn</a>               | Trang báo Lao động                                                         |
| <a href="http://www.nhandan.org.vn">www.nhandan.org.vn</a>               | Trang báo Nhân dân                                                         |
| <a href="http://www.vietnamtourism.com.vn">www.vietnamtourism.com.vn</a> | Trang của Du lịch Việt                                                     |
| <a href="http://www.hut.edu.vn">www.hut.edu.vn</a>                       | Trang chủ của Trường ĐH Bách Khoa HN                                       |
| <a href="http://www.dec.edu.vn">www.dec.edu.vn</a>                       | Trang web của Trường ĐH Cần Thơ                                            |
| <a href="http://www.fao.org">www.fao.org</a>                             | Trang web của tổ chức FAO                                                  |
| <a href="http://www.manguon.com">www.manguon.com</a>                     | Trang web của công ty Mắt bảo về CNTT                                      |
| <a href="http://www.real-english.com">www.real-english.com</a>           | Trang web học tiếng Anh trực tuyến                                         |
| <a href="http://www.caycanhvietnam.com">www.caycanhvietnam.com</a>       | Trang web giới thiệu về cây cảnh Việt nam                                  |
| <a href="http://www.bkav.com.vn">www.bkav.com.vn</a>                     | Trang web của Trung tâm Phần mềm và Giải pháp An ninh mạng ĐH Bách khoa HN |
| <a href="http://www.hoahoctro.com">www.hoahoctro.com</a>                 | Trang báo Hoa học trò                                                      |
| <a href="http://www.loveclub.saigonnet.vn">www.loveclub.saigonnet.vn</a> | Câu lạc bộ tình yêu, hôn nhân và gia đình                                  |
| <a href="http://www.yahoo.com">www.yahoo.com</a>                         | Trang chủ của Yahoo                                                        |
| <a href="http://www.google.com">www.google.com</a>                       | Trang để tìm kiếm                                                          |
| <a href="http://www.download.com">www.download.com</a>                   | Trang chủ để tải các phần mềm                                              |
| <a href="http://www.cnn.com">www.cnn.com</a>                             | Trang chủ của hãng CNN                                                     |
| <a href="http://www.cocacola.com">www.cocacola.com</a>                   | Trang chủ của hãng Cocacola                                                |
| <a href="http://www.healthfinder.gov">www.healthfinder.gov</a>           | Trang cung cấp các thông tin về y tế                                       |
| <a href="http://www.asd.com">www.asd.com</a>                             | Trang thông tin về các trường học ở Mỹ                                     |
| <a href="http://www.amazon.com">www.amazon.com</a>                       | Cửa hàng sách lớn nhất thế giới.                                           |

## **Câu hỏi ôn tập chương V**

1. Mạng Internet là gì? Các chức năng chính của mạng Internet.
2. Để một máy tính các nhân kết nối được vào mạng Internet cần có những gì ?
3. Trình duyệt Internet Explorer có những chức năng gì?
4. E-mail là gì? Các mô hình hoạt động của dịch vụ E-mail.
5. Cách tạo hòm thư miễn phí với Yahoo và với Outlook Express.

## CHƯƠNG VI: GIẢI THUẬT

### 1. Khái niệm giải thuật (Algorithms)

- Khi cần giải quyết một bài toán trong thực tế với sự trợ giúp của máy tính điện tử ta thường phải biết dữ liệu vào của bài toán (Input) là gì? và bài toán yêu cầu dữ liệu ra (Output) là gì?. Bước tiếp theo ta phải thiết lập được các bước thao tác cụ thể để từ Input ta có được Output. Công việc đó trong tin học được gọi là xây dựng giải thuật.

- **Giải thuật của 1 bài toán là một dãy các câu lệnh (Statements) chặt chẽ và rõ ràng xác định một trình tự các thao tác trên một số đối tượng nào đó sao cho sau một số bước hữu hạn thực hiện ta thu được kết quả mong muốn.**

- Với định nghĩa như vậy ta thấy rằng đối với một bài toán cụ thể có thể có nhiều giải thuật khác nhau nhưng tất nhiên là các giải thuật đó phải cho cùng một kết quả theo đúng yêu cầu của bài toán.

- Khi nghiên cứu về giải thuật thường ta phải biết được giải thuật đó tác động lên dữ liệu nào. Việc lựa chọn cấu trúc dữ liệu (Data structures) phù hợp và việc thiết lập được các giải thuật đúng đắn có cấu trúc tốt và hiệu quả là những vấn đề mấu chốt của công việc thiết lập phần mềm. Chính vì vậy mà Niklaus Wirth người sáng lập ra ngôn ngữ lập trình Pascal đã tổng kết: Giải thuật+Cấu trúc dữ liệu= Chương trình

**Ví dụ:** Xây dựng giải thuật tìm UCLN của 2 số nguyên dương a và b, ký hiệu (a,b)

+ Đối với bài toán này ta có

Input: 2 số nguyên dương a, b

Output: (a,b)

+ Giải thuật được xây dựng dựa trên tính chất

Nếu  $a=b$  thì  $(a,b)=a$

Nếu  $a>b$  thì  $(a,b)=(a-b,b)$

Nếu  $a<b$  thì  $(a,b)=(a,b-a)$

+ Cụ thể giải thuật của bài toán như sau:

**Bước 1:** So sánh a và b, Nếu  $a=b$  thì dừng giải thuật và thông báo  $(a,b)=a$ . Nếu  $a \neq b$  thì chuyển sang bước 2

**Bước 2:** Nếu  $a>b$  thì thay thế a bởi  $a-b$ , nếu  $a<b$  thì thay thế b bởi  $b-a$ . Quay lại thực hiện bước 1

Minh hoạ:  $a=20, b=32$

| Bước thực hiện | a  | b  | Kiểm tra điều kiện $a=b$ |
|----------------|----|----|--------------------------|
| Bước 1         | 20 | 32 | Sai                      |
| Bước 2         | 20 | 12 |                          |
| Bước 1         | 20 | 12 | Sai                      |
| Bước 2         | 8  | 12 |                          |
| Bước 1         | 8  | 12 | Sai                      |
| Bước 2         | 8  | 4  |                          |
| Bước 1         | 8  | 4  | Sai                      |
| Bước 2         | 4  | 4  |                          |
| Bước 1         | 4  | 4  | Đúng                     |

Kết quả là:  $(20,32)=4$



## 2. Các yêu cầu với giải thuật

Giải thuật của bài toán phải thoả mãn 3 yêu cầu sau:

**Yêu cầu 1:** Tính dừng

Giải thuật phải dừng sau một số hữu hạn các thao tác, đây là yêu cầu hết sức quan trọng với một giải thuật

**Yêu cầu 2:** Tính đúng đắn

Ta phải đặt câu hỏi "Liệu giải thuật có thể hiện đúng lời giải của bài toán không?" . Thông thường chúng ta cài đặt giải thuật dưới dạng chương trình và cho thực hiện trên máy tính với một số bộ dữ liệu nào đó, sau đó so sánh với những kết quả mà ta đã biết. Nhưng cách thử này chỉ khẳng định được tính sai chứ chưa thể khẳng định được tính đúng đắn của giải thuật. Bằng cách sử dụng các công cụ toán học ta có thể khẳng định được tính đúng đắn của 1 giải thuật nhưng thường thì đây là một công việc phức tạp.

**Yêu cầu 3:** Tính đơn giản và hiệu quả

Ta thường mong muốn xây dựng được một giải thuật đơn giản, dễ hiểu, dễ lập trình. Nhưng đôi khi thuật giải đơn giản lại gây ra sự lãng phí thời gian và bộ nhớ. Do đó mục tiêu là phải xây dựng được các giải thuật có thời gian thực hiện nhanh, hạn chế tối đa dung lượng bộ nhớ dành cho việc lưu trữ những kết quả trung gian.

## 3. Các cách diễn tả giải thuật

### 3.1. Cách 1: Liệt kê từng bước

**Ví dụ:** Có 31 que diêm, người và máy thay nhau bốc. Mỗi lần bốc từ 1 đến 4 que. Ai phải bốc sau cùng là thua. Hãy xây dựng thuật giải sao cho máy bốc trước bao giờ cũng thua.

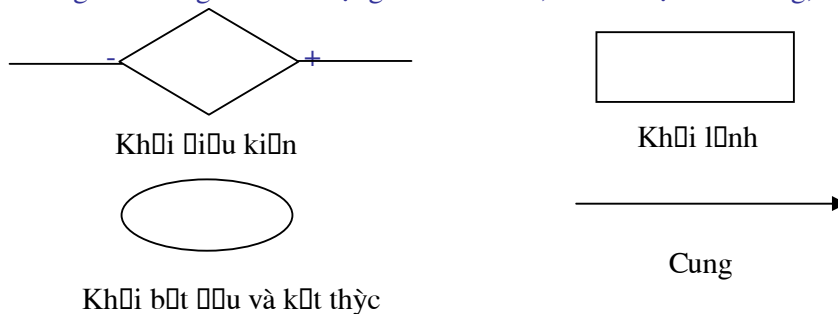
**Bước 1:** Máy bốc ngẫu nhiên  $x$  que diêm ( $1 \leq x \leq 4$ )

**Bước 2:** Người bốc  $(5 - x)$  que, tổng số que diêm giảm đi 5 que. Nếu số que diêm còn lại là 1 que thì chuyển sang bước 3, nếu không thì quay lại thực hiện bước 1

**Bước 3:** Tuyên bố người thắng cuộc

### 3.2. Cách 2: Sử dụng lưu đồ

Sử dụng phương tiện hình học cũng là một cách tốt để minh hoạ giải thuật của 1 bài toán. Trong lưu đồ người ta sử dụng các hình sau, với kí hiệu + là đúng, - là sai.

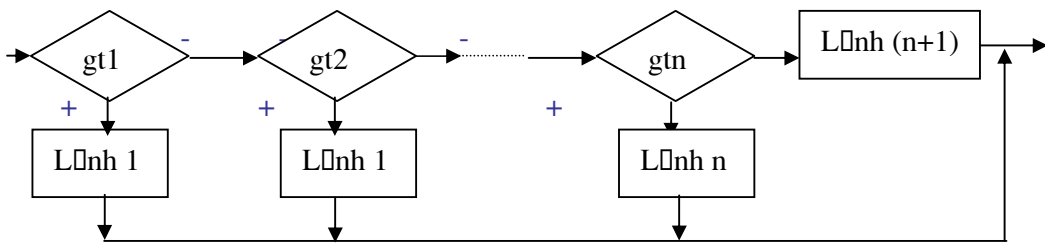
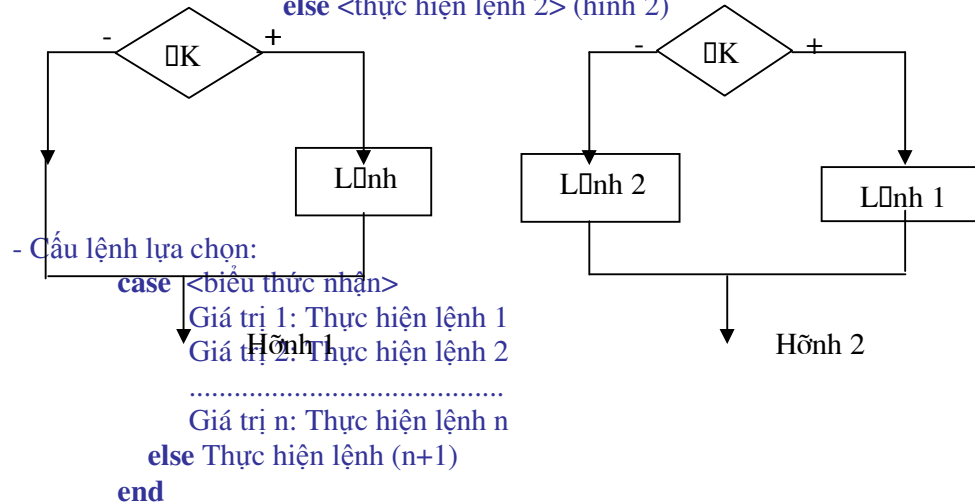


**a. Cấu trúc rẽ nhánh**

- Rẽ nhánh dạng khuyết: **if** <đk thoả mãn> **then** <thực hiện lệnh> (hình 1)

- Rẽ nhánh dạng đủ: **if** <đk thoả mãn> **then** <thực hiện lệnh 1>

**else** <thực hiện lệnh 2> (hình 2)



Hình 3

**b. Cấu trúc lặp**

+ Lặp một số lần định trước:

**Dạng 1: for** i:=m to n **do** <lệnh>

Thực hiện lệnh với i nhận các giá trị nguyên tăng từ m tới n với bước nhảy bằng 1

**Dạng 2: for** i:=n **downto** m **do** <lệnh>

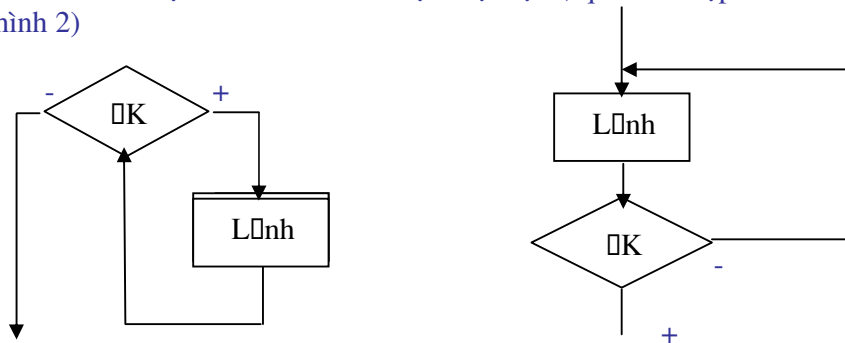
Tương tự như dạng 1 nhưng bước nhảy giảm bằng 1

+ Lặp với điều kiện trước: **while** <điều kiện> **do** <lệnh>

Khi điều kiện còn đúng thì còn thực hiện lệnh, quá trình lặp kết thúc khi điều kiện sai (hình 1)

+ Lặp với điều kiện sau: **repeat** <lệnh> **until** <điều kiện>

Khi điều kiện còn sai thì còn thực hiện lệnh, quá trình lặp kết thúc khi điều kiện đúng (hình 2)



### 3.3 Cách 3: Sử dụng giả ngôn ngữ có cấu trúc tựa ngôn ngữ lập trình bậc cao

Là phương pháp diễn tả giải thuật dựa vào các cấu trúc điều khiển, cùng với các từ khoá của một ngôn ngữ lập trình bậc cao nào đó. Trong giáo trình này ta sẽ sử dụng ngôn ngữ tựa Pascal để diễn tả giải thuật. Cách diễn đạt này đã tiếp cận gần hơn với ngôn ngữ lập trình.

**Ví dụ:** Với thuật giải tìm UCLN ở trên ta có thể diễn đạt như sau

```
while a≠b
 begin
 if a>b then thay a bởi a-b
 else thay b bởi b-a
 end
write ước chung lớn nhất là a
```

## 4. Thiết kế giải thuật

### 4.1. Mô-đun hoá và việc giải quyết bài toán

Những bài toán ta gặp trong thực tế thường là phức tạp, trong trường hợp đó người ta thường chia bài toán thành những bài toán nhỏ và dễ giải quyết hơn. Nghĩa là coi bài toán ban đầu là Mô-đun chính, ta chia nó thành các Mô-đun con, và mỗi Mô-đun con này có thể lại được chia thành các Mô-đun nhỏ hơn... Cách giải quyết bài toán như vậy người ta thường gọi là chiến thuật "Chia để trị" (divide and conquer).

Trong khi lập trình việc chia chương trình chính thành các chương trình con thể hiện tính có cấu trúc của ngôn ngữ lập trình về mặt chương trình

### 4.2. Tinh chỉnh từng bước giải thuật

Tinh chỉnh từng bước là phương pháp thiết kế giải thuật gắn liền với lập trình. Bước đầu giải thuật được minh hoạ bằng ngôn ngữ tự nhiên, càng ở các bước sau ngôn ngữ tự nhiên được thay thế bởi ngôn ngữ tự nhiên pha lẫn ngôn ngữ lập trình mà ta gọi là giả ngôn ngữ.

Ta có sơ đồ sau: Ngôn ngữ tự nhiên → Giả ngôn ngữ → Ngôn ngữ lập trình

### 4.3. Phân tích thuật giải

Phân tích giải thuật phải căn cứ vào 3 tiêu chuẩn đối với một giải thuật: Tính đúng, tính đúng đắn, tính đơn giản và hiệu quả.

Việc kiểm tra giải thuật là một phần hết sức quan trọng, lý tưởng là khi có thể khẳng định một cách hình thức tính đúng đắn của giải thuật. Tuy nhiên trong thực tế thời gian và công sức để viết ra một cách cẩn thận và đầy đủ tất cả những chi tiết chứng minh tính đúng đắn của một giải thuật phức tạp thường là không cho phép. Người lập trình thường áp dụng các biện pháp sau:

- Chứng minh một cách suy diễn rằng những bước trong giải thuật là đúng đắn. Nghĩa là giải thuật bắt đầu bằng một khẳng định (giả thiết) về dữ liệu vào và dùng phương pháp suy luận logic để chỉ ra rằng việc thực hiện giải thuật sẽ cho một khẳng định (kết luận) về đầu ra.

- Thể hiện giải thuật bằng một ngôn ngữ lập trình và thử thực hiện chương trình với các bộ dữ liệu vào mà kết quả ta đã biết trước. Thường thì các lỗi về cú pháp và lỗi lúc thực hiện chương trình thường dễ tìm và sửa chữa còn các lỗi logic thường khó phát hiện hơn nhiều.

- Có đôi lúc việc kiểm tra chương trình phải thực hiện thủ công, kiểm tra từng bước, từng thủ tục của chương trình chính. Kỹ thuật này được gọi là “đi bộ qua chương trình”(Walking through the program)

- Quan tâm đặc biệt tới thời gian thực hiện chương trình, thời gian thực hiện phụ thuộc rất nhiều vào việc tổ chức dữ liệu đưa vào (kích thước dữ liệu).

## 5. Giải thuật sắp xếp (Sorting)

Sắp xếp là một trong số yêu cầu thường xuyên xuất hiện trong quá trình xử lý số liệu. Bản chất của thuật giải sắp xếp là bố trí lại vị trí của số liệu theo thứ tự tăng dần hoặc giảm dần. Có nhiều giải thuật sắp xếp trong tin học, trong giáo trình này chúng ta sẽ đề cập đến một số thuật giải đơn giản đó là sắp xếp lựa chọn (selection sort), sắp xếp chèn (insertion sort) và sắp xếp nổi bọt (bubble sort). Đề đơn giản giả sử yêu cầu của bài toán là: Sắp xếp một dãy số cho trước  $a_1, a_2, \dots, a_n$  theo thứ tự tăng dần.

### 5.1 Sắp xếp lựa chọn (selection sort)

Thuật giải chọn được diễn tả như sau: Tìm phần tử nhỏ nhất trong dãy số và hoán vị nó với phần tử đầu tiên, tìm phần tử nhỏ nhất kế tiếp và hoán vị nó với phần tử thứ hai. Tiếp tục quá trình này đến khi toàn bộ dãy số được sắp xếp.

```
procedure Selection_Sort;
begin
 for i:=1 to n-1 do
 begin
 m:=i
 for j:=i+1 to n do if $a_j < a_m$ then m:=j
 if $m \neq i$ then đổi chỗ a_i và a_m
 end
 end
```

#### Ví dụ

|                  |    |   |    |   |   |
|------------------|----|---|----|---|---|
| Dãy số ban đầu : | 3  | 6 | -2 | 7 | 5 |
| i=1              | -2 | 6 | 3  | 7 | 5 |
| i=2              | -2 | 3 | 6  | 7 | 5 |
| i=3              | -2 | 3 | 5  | 7 | 6 |
| i=4              | -2 | 3 | 5  | 6 | 7 |

### 5.2 Sắp xếp chèn (insertion sort)

Thuật giải chèn được diễn tả như sau: Xét lần lượt từng phần tử và chèn vào vị trí thích hợp của phần tử đó trong số các phần tử đã xét trước đó. Cụ thể giả sử đã có (i-1) phần tử được sắp xếp đúng vị trí, để chèn phần tử thứ i vào đúng vị trí ta so sánh lần lượt với các phần tử thứ (i-1), (i-2),... khi tìm được vị trí đúng thì chèn phần tử thứ i đó vào.

```

procedure Insertion_Sort
begin
 for i:=2 to n do
 begin
 k:=ai
 j:=i
 while aj-1>k do begin aj:=aj-1, j:=j-1 end
 aj:=k
 end
 end

```

**Ví dụ**

|                  |    |   |    |   |   |
|------------------|----|---|----|---|---|
| Dãy số ban đầu : | 3  | 6 | -2 | 7 | 5 |
| i=2              | 3  | 6 | -2 | 7 | 5 |
| i=3              | -2 | 3 | 6  | 7 | 5 |
| i=4              | -2 | 3 | 6  | 7 | 5 |
| i=5              | -2 | 3 | 5  | 6 | 7 |

**5.3 Sắp xếp nổi bọt (bubble sort)**

Thuật giải này còn có tên gọi khác là sắp xếp bằng cách đổi chỗ trực tiếp (exchange sort), thuật giải nổi bọt được diễn tả như sau: Duyệt dãy số theo thứ tự từ phải sang trái nếu hai phần tử kề cận ngược thứ tự thì đổi chỗ cho nhau. Như vậy sau lượt duyệt đầu tiên phần tử đầu tiên sẽ là phần tử nhỏ nhất, sau lượt thứ hai phần tử nhỏ thứ hai được chuyển lên vị trí thứ hai...cứ như vậy dãy số sẽ được sắp xếp tăng dần.

```

procedure Bubble_Sort
begin
 for i:=1 to n-1 do
 for j:=n downto i+1 do
 if aj<aj-1 then đổi chỗ aj và aj-1
 end
 end

```

**Ví dụ**

|                  |    |   |    |   |   |
|------------------|----|---|----|---|---|
| Dãy số ban đầu : | 3  | 6 | -2 | 7 | 5 |
| i=1              | -2 | 3 | 6  | 5 | 7 |
| i=2              | -2 | 3 | 5  | 6 | 7 |
| i=3              | -2 | 3 | 5  | 6 | 7 |
| i=4              | -2 | 3 | 5  | 6 | 7 |

**6. Giải thuật tìm kiếm (Searching)**

Cùng với các thuật giải sắp xếp, các thuật giải tìm kiếm cũng đóng một vai trò quan trọng trong khi xử lý số liệu. Bài toán tìm kiếm đặt ra như sau: Giả sử ta có một dãy số a<sub>1</sub>, a<sub>2</sub>,..., a<sub>n</sub>, ta phải tìm vị trí của phần tử có giá trị bằng giá trị X cho trước. Chúng ta sẽ xét hai thuật giải tìm kiếm đó là tìm kiếm tuần tự (sequential searching) và tìm kiếm nhị phân (binary searching).

## 6.1 Tìm kiếm tuần tự (sequential searching)

Đây là thuật giải tìm kiếm đơn giản nhất, ta sẽ duyệt tuần tự dãy số, thuật giải sẽ kết thúc khi tìm thấy phần tử bằng giá trị X hoặc khi duyệt hết dãy số nhưng không có phần tử nào có giá trị là X

```
procedure Sequential_Searching
begin
i:=1
while $a_i \neq X$ do i:=i+1
if $i=n+1$ then không có phần tử cần tìm
else vị trí phần tử cần tìm là i
end
```

## 6.2 Tìm kiếm nhị phân (binary searching)

Giả sử dãy số đã được sắp xếp tăng dần  $a_1 \leq a_2 \leq \dots \leq a_n$  (trường hợp sắp xếp giảm dần thì tương tự), thuật giải nhị phân gần giống như khi ta tìm một từ trong từ điển. Để tìm phần tử bằng X trước tiên ta so sánh nó với phần tử ở vị trí giữa của dãy số nếu X nhỏ hơn thì X chỉ có thể ở trong một nửa trước của dãy nếu ngược lại thì X chỉ có thể ở trong nửa sau của dãy. Lặp lại quá trình tìm kiếm đó đến khi tìm thấy hoặc dãy số trở nên rỗng (không tìm thấy).

```
procedure Binary_Searching
begin
left:=1
right:=n
repeat
mid:=(left+right)/2 (*Kí hiệu [a] nghĩa là lấy phần nguyên của số thực a*)
if $X < a_{mid}$ then right:=mid-1
else left:=mid+1
until ($X = a_{mid}$) or (left > right)
if $X = a_{mid}$ then vị trí cần tìm là mid
else không có phần tử cần tìm
end
```

**Ví dụ:** Tìm phần tử 28 trong dãy số sau

|           |    |           |      |           |    |    |      |
|-----------|----|-----------|------|-----------|----|----|------|
|           | [4 | 15        | 28   | <u>33</u> | 67 | 99 | 103] |
| Lặp lần 1 | [4 | <u>15</u> | 28]  |           |    |    |      |
| Lặp lần 2 |    |           | [28] |           |    |    |      |

## 7. Giải thuật đệ quy

### 7.1. Khái niệm đệ quy

Một đối tượng được gọi là đệ quy nếu nó bao gồm một phần của chính nó hay được định nghĩa bởi chính nó.

Trong khi thiết kế giải thuật ta thường thiết kế dưới dạng các mô-đun. Khi giải thuật được cài đặt thành chương trình thì các mô-đun sẽ tương ứng với các chương trình con (hàm-function và thủ tục- procedure),

Chương trình con được gọi là đệ quy nếu trong thân của nó có lời gọi trực tiếp hoặc gián tiếp đến chính bản thân nó.

Đệ quy có ý nghĩa đặc biệt quan trọng trong các định nghĩa toán học

**Ví dụ 1:** Định nghĩa số tự nhiên

- + 0 là số tự nhiên
- + Số tiếp theo của một số tự nhiên là một số tự nhiên

**Ví dụ 2:** Định nghĩa n!

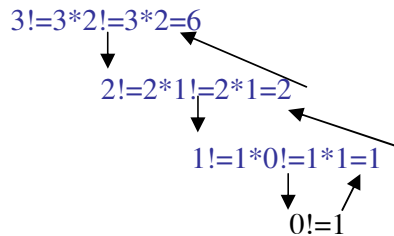
- + 0!=1
- + n!=n\*(n-1)! nếu n>0

- Định nghĩa một phép đệ qui gồm có 2 phần
  - + Trường hợp suy biến: Giúp cho quá trình đệ qui kết thúc
  - + Phần đệ qui (hay phần qui nạp): Trong đó tác động cần được thực hiện cho giá trị hiện thời của các tham số được định nghĩa bằng các tác động hay giá trị được định nghĩa trước đây

Trong ví dụ định nghĩa n! thì trường hợp suy biến định nghĩa 0!, phần qui nạp định nghĩa n! qua các giá trị của n và giá trị của (n-1)!

Dễ nhận xét, nếu (n-1)! đã tính được thì n! sẽ dễ dàng tính được. Với cách suy diễn tương tự, (n-1)! sẽ tính được nếu như (n-2)! đã tính được... cuối cùng 1! sẽ tính được nếu 0! đã tính được. Ta thấy rằng 0! đã cho trong định nghĩa. Do vậy đi ngược từ cuối, vì 0! đã tính được nên 1! cũng tính được,....,sau khi (n-1)! đã có ta sẽ nhận được n!

Minh họa: Tính 3!



Giải thuật được viết dưới dạng thủ tục hàm (tựa Pascal) như sau:

```

function giaithua(n)
begin
 if n=0 then giaithua:=1 (* trường hợp suy biến*)
 else giaithua:=n*giaithua(n-1) (* phần đệ qui*)
end

```

**end**

Chú ý: Không phải lúc nào tính đệ qui trong cách giải bài toán cũng thể hiện rõ nét và dễ phát hiện như ví dụ trên. Do đó muốn biết giải thuật của một bài toán có thể thiết kế dưới dạng giải thuật đệ qui được hay không? Có thể thấy câu trả lời qua việc trả lời các câu hỏi sau:

- + Có thể định nghĩa được bài toán dưới dạng một bài toán cùng loại nhưng “nhỏ” hơn không?
- + Kích thước của bài toán sẽ giảm đi ở mỗi bước gọi đệ qui như thế nào?
- + Trường hợp nào của bài toán được coi là trường hợp suy biến?

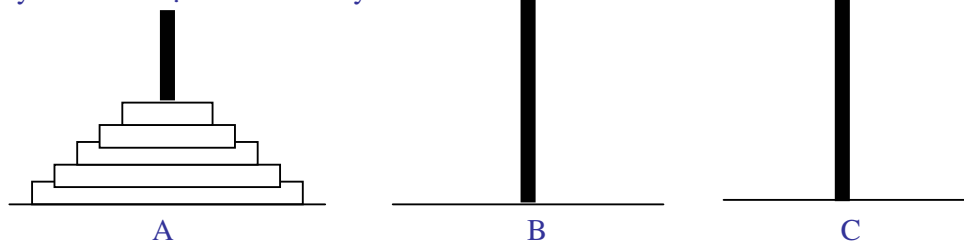
**7. 2. Ví dụ về giải thuật đệ qui : Bài toán tháp Hà Nội**

Bài toán Tháp Hà Nội là một ví dụ cổ điển cho thấy thuật toán đệ qui là đặc biệt thích hợp. Có thể giải quyết bài toán một cách dễ dàng nếu dùng đệ qui, nhưng cách giải không đệ qui là tương đối khó.

Nội dung bài toán: Tại cọc A có n chiếc đĩa, đĩa to ở dưới, đĩa nhỏ ở trên. Chuyển các đĩa từ cọc A sang cọc C có thể nhờ cọc B làm vị trí trung chuyển theo các quy tắc sau:

- Mỗi lần chỉ được chuyển một đĩa và phải là đĩa ở trên cùng
- Đĩa lớn không bao giờ được phép nằm trên đĩa nhỏ

- Khi chuyển một đĩa, nó phải được đặt vào một trong 3 cọc ở trên  
 Hãy chỉ ra thứ tự các bước chuyển.



Một truyền thuyết cho rằng các thầy tu ở Điện Bramah được cho một bài toán đố với một nền vàng có 3 kim vàng trên 1 cọc có 64 đĩa vàng. Khi họ chuyển các đĩa vàng theo các luật của bài toán trên, nếu mỗi giây chuyển được một đĩa và họ bắt đầu công việc từ năm 0, đến khi hoàn thành công việc thì sẽ là ngày tận thế.

Những người mới bắt đầu có thể giải bài toán một cách dễ dàng với số các đĩa là bé, nhưng họ sẽ rất khó khăn khi số đĩa tăng lên 7,8 và lớn hơn. Tuy nhiên với một nhà lập trình thì có thể giải bài toán một cách không mấy khó khăn.

Cách giải: Nếu có một đĩa, chuyển nó từ cọc A sang cọc C. Bài toán giải được với  $n=1$

Giả sử rằng bài toán có nghiệm với  $n-1$  đĩa, nghiệm với  $n$  đĩa có thể nhận được một cách dễ dàng nhờ dùng phép đệ quy:

+ Chuyển  $n-1$  đĩa trên cùng ở cọc A sang cọc B, dùng cọc C làm trung chuyển

+ Chuyển đĩa còn lại ở cọc A sang cọc C

+ Chuyển  $n-1$  từ cọc B sang cọc C, dùng cọc A làm trung chuyển

Ta có thể viết giải thuật của bài toán Tháp Hà Nội như sau:

**procedure** Move( $n,A,B,C$ )

(\* Chuyển  $n$  đĩa từ cọc A sang cọc C, dùng cọc B làm trung chuyển\*)

**begin**

**if**  $n=1$  **then** chuyển đĩa từ A sang C

**else**

**begin**

Move( $n-1,A,C,B$ )

(\* Chuyển  $n-1$  đĩa từ A sang B, dùng C làm trung chuyển\*)

Move( $1,A,C$ )

(\* Chuyển 1 đĩa từ cọc A sang cọc C\*)

Move( $n-1,B,A,C$ )

(\* Chuyển  $n-1$  đĩa từ B sang C, dùng A làm trung chuyển\*)

**end**

**end**



## Bài tập chương VI: Thuật giải

Viết giải thuật cho các bài toán sau:

1. Tính  $n$  giai thừa:  $n! = 1.2...n$  với  $n > 1$
2. Tính các tổng:  $S = 1/2 + 1/4 + \dots + 1/(2k)$   
 $Q = 1.1! + 2.2! + \dots + n.n!$
3. Tìm và in ra tất cả các số chính phương nhỏ hơn một số cho trước, cho biết có bao nhiêu số chính phương như vậy.
4. Viết chương trình giải bài toán cổ: "Vừa gà vừa chó, bó lại cho tròn, ba mươi sáu con, một trăm chân chẵn. Hỏi có bao nhiêu gà, bao nhiêu chó?"
5. Viết chương trình tìm ước số chung lớn nhất của 2 số nguyên dương cho trước.
6. Tính  $E_x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \dots + \frac{x^n}{n!} + \dots$  với độ chính xác  $\epsilon = 10^{-4}$  ( $ABS(x^n/n!) < \epsilon$ ), giá trị  $x$  được nhập vào từ bàn phím khi chạy chương trình.
7. Cần có 50000 đ từ các loại giấy bạc 1000đ, 2000đ và 5000đ. Tìm tất cả các phương án có thể.
8. Chuyển một số thập phân nguyên dương thành một số nhị phân, in ra màn hình dạng  $X_{10} = Y_2$
9. Tính tích phân xác định của một hàm số trên một đoạn cho trước
10. Viết chương trình tìm và in ra màn hình các số nguyên tố nhỏ hơn một số cho trước.
11. Cho dãy số sau:  $a_1, a_2, \dots, a_n$ . Viết chương trình tìm phân tử lớn nhất, phân tử nhỏ nhất của dãy số đó.
12. Cho dãy số sau:  $a_1, a_2, \dots, a_n$ . Viết chương trình sắp xếp dãy theo thứ tự tăng dần.
13. Cho dãy số sau:  $a_1, a_2, \dots, a_n$ . Viết chương trình đếm số phân tử dương và xoá đi phân tử thứ  $m$  trong dãy ( $m \leq n$ ).
14. Cho dãy số sau:  $a_1, a_2, \dots, a_n$ . Viết chương trình tìm các phân tử có giá trị là  $x$  nhập vào từ bàn phím.
15. Cho dãy số sau:  $a_1, a_2, \dots, a_n$ . Viết chương trình thêm phân tử có giá trị là  $x$ , vào vị trí  $m$  trong dãy. Sau đó tính tổng các phân tử của dãy mới.
16. Viết chương trình in ra các số nguyên tố trong khoảng từ  $1 \rightarrow n$
17. Cho ma trận có  $m$  dòng và  $n$  cột, các phần tử là nguyên. Tìm phần tử nhỏ nhất của ma trận.
18. Cho ma trận có  $m$  dòng và  $n$  cột, các phần tử là nguyên. Tính tổng và trung bình cộng các phần tử của ma trận.

## PHẦN II NGÔN NGỮ LẬP TRÌNH TURBO PASCAL

### CHƯƠNG I: CẤU TRÚC CỦA CHƯƠNG TRÌNH PASCAL

#### 1. Giới thiệu về ngôn ngữ lập trình

##### 1.1. Ngôn ngữ lập trình là gì ?

Ngôn ngữ lập trình là một hệ thống các kí hiệu, hệ thống các quy tắc cú pháp và một bộ các chỉ thị (hay còn gọi là câu lệnh) dùng để viết các thuật toán giải các bài toán thành các chương trình máy tính. Những chương trình này sẽ được đưa vào máy "chạy" để tính toán, xử lí bài toán.

Ngôn ngữ lập trình ra đời cùng với sự ra đời của MTĐT vì khi MTĐT ra đời đương nhiên người ta phải nghĩ đến việc xây dựng ngôn ngữ cho chúng. Các thế hệ MTĐT liên tục phát triển, do đó ngôn ngữ lập trình cũng phát triển theo. Có nhiều loại ngôn ngữ lập trình khác nhau, từ ngôn ngữ bậc thấp chuyển lên ngôn ngữ bậc cao.

+ Ngôn ngữ bậc thấp (ngôn ngữ máy): Chỉ sử dụng 2 kí hiệu 0 và 1 để mã hoá mọi đại lượng và phép toán. Ngôn ngữ này có ưu điểm là chương trình chạy nhanh, không phải qua khâu dịch. Tuy nhiên những chương trình viết bằng ngôn ngữ này rất cồng kềnh, viết mất rất nhiều thời gian, dễ sai sót, khó kiểm tra; mặt khác mỗi loại máy có một ngôn ngữ máy riêng.

+ Ngôn ngữ tập hợp: Khắc phục một số nhược điểm trên người ta nghĩ cách thay thế một số mã nhị phân bằng các chữ cái và cho ra đời ngôn ngữ tập hợp (còn gọi là ngôn ngữ kí hiệu). Ưu điểm là việc lập trình đã dễ dàng hơn, tuy nhiên vẫn cần phải có chương trình dịch từ ngôn ngữ tập hợp ra ngôn ngữ máy, và vẫn còn phụ thuộc vào từ máy.

+ Ngôn ngữ bậc cao: Chương trình được viết gần với ngôn ngữ tự nhiên của con người, do đó dễ dàng cho người lập trình, tuy nhiên chương trình sẽ chạy chậm hơn. Năm 1958, ALGOL - ngôn ngữ lập trình bậc cao đầu tiên ra đời. Năm 1960 nó được bổ sung, chỉnh lí và có ảnh hưởng sâu sắc đến các ngôn ngữ lập trình bậc cao sau này như FORTRAN, C, BASIC, PASCAL,...

##### 1.2. Turbo Pascal

PASCAL là ngôn ngữ lập trình bậc cao của tác giả Niklaus Wirth (giáo sư người Thụy Sĩ), được công bố vào đầu những năm 1970. Tên PASCAL là để kỉ niệm nhà Toán học người Pháp B. Pascal.

PASCAL là ngôn ngữ lập trình có **tính cấu trúc và tính hệ thống**: các kiểu dữ liệu đa dạng, các cấu trúc điều khiển chặt chẽ, các cấu trúc khối trong chương trình rõ ràng...

PASCAL là ngôn ngữ lập trình có **định kiểu rõ ràng**: các đại lượng (biến và hằng) đã được khai báo để sử dụng với kiểu dữ liệu này thì không thể đem dùng lẫn với kiểu khác.

PASCAL ban đầu được sáng tác để làm ngôn ngữ dạy học cho những người mới học lập trình. Đặc tính sáng sủa, dễ hiểu, dễ đọc của nó giúp người mới học có thể viết một chương trình máy tính một cách dễ dàng. Sau khi ra đời, do có nhiều ưu điểm, PASCAL đã được nhiều hãng máy tính phát triển và cài đặt cho nhiều hệ thống máy tính, như: ISO PASCAL (PASCAL chuẩn), ANSI PASCAL v.v. Phổ biến nhất hiện nay ở nước ta cũng như trên thế giới là việc cài đặt TURBO PASCAL cho các hệ thống máy tính. TURBO PASCAL được hãng BORLAND INTERNATIONAL hoàn thiện với các ưu điểm là chương trình viết gọn,

dịch nhanh, không ngừng được cải tiến đáp ứng yêu cầu của người sử dụng. TURBO PASCAL đã được phát triển qua các phiên bản ngày càng mở rộng và phức tạp. Việc khai thác sâu các tính năng của từng phiên bản là do khả năng và nhu cầu của người lập trình. Giáo trình này chọn version 5.0 làm chuẩn để minh họa, tuy nhiên người đọc có thể áp dụng cho các phiên bản sau này như 6.0, 7.0.

## 2. Yêu cầu phần mềm và cách khởi động

### 2.1. Yêu cầu phần mềm

Phần mềm TURBO PASCAL có khá nhiều File, tuy nhiên để có thể bước đầu học và sử dụng được, trên đĩa cần có tối thiểu các file sau:

TURBO.EXE file chính, chứa cả chương trình dịch, hệ soạn thảo và các dịch vụ gỡ rối...

TURBO.TPL là thư viện các chương trình mẫu có sẵn của TURBO PASCAL

Nếu có sử dụng đồ họa thì cần chép thêm đơn vị GRAPH.TPU và một số file phụ trợ khác tùy theo kiểu màn hình và kiểu chữ sử dụng.

### 2.2. Khởi động và thoát TURBO PASCAL

Thông thường các file nói trên nằm trong thư mục TP (hoặc TURBO hay TP5, TP7 tùy theo phiên bản hay người sử dụng)

Ta chuyển vào thư mục này bằng lệnh CD TP↵, sau đó gõ TURBO ↵. Màn hình sẽ hiện ra như sau:

| File                                                        | Edit  | Run    | Compile | Options  | Debug | Break/Watch  |
|-------------------------------------------------------------|-------|--------|---------|----------|-------|--------------|
| Line 1                                                      | Col 1 | Insert | Indent  | Unindent |       | C:NONAME.PAS |
|                                                             |       |        |         |          |       |              |
| F1-Help F5-Zoom F6-Switch F7-Trace F8-Step F9-Make F10-Menu |       |        |         |          |       |              |

Dòng 1: là dòng menu chính, trên đó có ghi các mục cơ bản nhất để người sử dụng có thể chọn bằng cách ấn đồng thời Alt+ chữ cái đầu của mục. Ví dụ: Alt+F để mở mục File.

Dòng 2: là dòng trạng thái, cho người dùng biết một số thông số soạn thảo như tọa độ con trỏ, chế độ soạn thảo là viết chèn hay viết đè, tên tệp đang làm việc...

Dòng cuối cùng là dòng hướng dẫn, nhắc người sử dụng một số phím chức năng chính.

Phần màn hình rộng ở giữa là phần soạn thảo để ta nhập và chạy chương trình.

Người mới học lập trình nên ghi nhớ ngay các phím chức năng:

F2 ghi tệp đang soạn thảo vào đĩa. Nếu chưa đặt tên sẽ có thông báo để ta gõ tên vào (tên được đặt theo quy định của MS-DOS)

- F3 mở tệp mới
- Alt+F9 dịch chương trình
- Ctrl+F9 dịch và cho chạy chương trình
- Alt+X Kết thúc PASCAL trở về hệ điều hành

### 3. Những quy định về cách viết biểu thức và chương trình

#### 3.1. Bộ kí tự

TURBO PASCAL sử dụng bộ chữ viết như sau:

- Bộ 26 chữ cái Latinh: A, B, ..., Z ; a, b, ..., z
- Bộ các chữ số: 0, 1, ..., 9
- Các kí hiệu Toán học: +, -, \* , /, =, <, >, ...
- Dấu gạch nối, dấu chấm, dấu ngoặc và các kí hiệu đặc biệt khác: \_ . , ; ? [ ] ( ) \$ % ...

#### 3.2 Từ khoá

Từ khoá là một số từ dành riêng của TURBO PASCAL, những từ này đã được gán sẵn những chức năng đặc biệt, ta không được dùng vào việc đặt tên hay các công việc khác. Sau đây là một số từ quan trọng, bạn đọc sẽ lần lượt biết các từ khoá và chức năng của từng từ qua các bài học.

- Các từ khoá chung: Program, begin, end
- Các từ khoá để khai báo: uses, const, type, label, var  
procedure, function  
array, string, record, ...
- Các từ khoá lệnh:  
if... then... else...  
case ...of ....  
for .... to ... do...  
while... do...  
repeat... until...
- Một số tên chuẩn:  
Boolean, char, integer, real,  
read, readln, write, writeln,  
abs, cos, sin, sqrt, exp...

#### 3.3 Tên

Tên là một dãy kí tự dùng để chỉ tên các đại lượng như tên hằng, tên biến, tên kiểu hay tên chương trình con.

Tên được đặt theo quy tắc sau: bắt đầu bởi một chữ cái, rồi đến chữ số hay dấu gạch nối và không dài quá 127 kí tự (!). Tên thường được đặt mang tính gọi nhớ. TURBO PASCAL không phân biệt chữ hoa và chữ thường khi viết từ khoá hay tên.

Ví dụ : khi viết chương trình giải phương trình bậc 2, ta có thể đặt tên chương trình là GIAI\_PHUONG\_TRINH\_BAC2, đặt tên biến chứa biệt thức  $\Delta$  là delta (không được dùng kí hiệu  $\Delta$ ), đặt tên các nghiệm là X1 và X2,...

- Dấu chấm phẩy (;) dùng để ngăn cách các câu lệnh trong một chương trình
- Lời giải thích được đặt trong cặp dấu (\* ... \*) hoặc { ... }, các nội dung được chứa trong cặp dấu này sẽ bị chương trình bỏ qua trong khi dịch

## 4. Cấu trúc chung của một chương trình PASCAL

Ta hiểu một chương trình máy tính là một dãy các lệnh, các chỉ thị hướng dẫn máy thực hiện một nhiệm vụ, một xử lý nào đó trên tập các dữ kiện vào và cho ra kết quả.

Một chương trình PASCAL có cấu trúc chung như sau:

```
PROGRAM Tên_chương_trình;
USES Danh_sách_đơn_vi.;
LABEL Danh_sách_nhãn;
CONST Danh_sách_hằng;
TYPE Danh_sách_kiểu;
VAR Danh_sách_biến;
PROCEDURE Khai_báo_thủ_tục;
FUNCTION Khai_báo_hàm;
BEGIN
 { Các câu lệnh; }
END.
```

Một chương trình PASCAL điển hình gồm 3 phần: Phần tiêu đề, phần khai báo, phần thân chương trình.

### 4.1. Phần tiêu đề

Phần này có dạng

```
PROGRAM Tên_chương_trình;
```

bắt đầu bằng từ khoá PROGRAM, ít nhất một dấu cách rồi đến tên chương trình, kết thúc bởi dấu chấm phẩy (;). Tên\_chương\_trình do ta đặt ra, mang ý nghĩa nội dung của chương trình, dấu ; là bắt buộc phải có.

Ví dụ : Program Giai\_PT\_bac2;

Phần tiêu đề là tùy chọn, có thể có hay không cũng được, nhưng nếu đã có thì phải đủ các thành phần như trên.

### 4.2. Phần khai báo

Phần này mô tả các đối tượng của bài toán mà chương trình sẽ xử lý. Các khai báo khác nhau được viết cách nhau bởi dấu ;. Gồm:

```
Khai báo đơn vị uses ...
Khai báo nhãn label
Khai báo hằng: const
Khai báo kiểu: Type
Khai báo biến: Var
Khai báo chương trình con: Procedure
 Function
```

Các khai báo của phần này tùy thuộc từng bài toán cụ thể mà có thể có hay không, hoặc có một hoặc một vài khai báo. Cách khai báo và sử dụng cụ thể sẽ được giới thiệu ở trong mỗi phần sau.

### 4.3. Phần thân chương trình

Phần này được bắt đầu bởi từ khoá BEGIN , kết thúc bởi từ khoá END và dấu chấm (.) hết. Đặt giữa BEGIN và END là các lệnh của chương trình , các lệnh được viết cách nhau bởi dấu ; như sau:

```
BEGIN
 Lệnh 1;
 Lệnh 2;
 ...
 Lệnh n;
END.
```

Phần này bắt buộc phải có đối với mọi chương trình, nó chứa các lệnh xử lí các dữ liệu đã được mô tả ở phần khai báo. Đây là phần chính của chương trình để giải quyết bài toán. Người lập trình phải bỏ nhiều công sức để thiết kế thuật giải và viết chương trình cho phần này.

## 4.4. Các bước cơ bản khi lập trình

Trước khi viết một chương trình giải quyết một bài toán nào đó, người lập trình phải biết cách giải bài toán đó hay chính xác hơn phải biết thuật giải để giải bài toán và trình bày thuật giải bằng ngôn ngữ lập trình .

Bước 1: Soạn thảo chương trình

Đây là bước viết mới một chương trình, có thể viết trên giấy trước rồi nhập vào máy, sửa lỗi, cập nhật. Các chương trình này được gọi là chương trình nguồn, tên chúng được tự động gán đuôi PAS. Ta có thể dùng một hệ soạn thảo văn bản nào đó, thường là hệ soạn thảo văn bản của chính TURBO PASCAL.

Bước 2: Dịch chương trình (Alt+F9)

Gọi chương trình dịch (compiler) dịch chương trình nguồn đã viết ở bước 1 sang dạng mã máy, kết quả thường tạo ra các tệp dạng \*.EXE, \*.OBJ. Bước này sẽ cho phép ta phát hiện các lỗi để sửa. Thường các lỗi cú pháp như thiếu dấu ; hoặc viết sai từ khoá, sai tên sẽ được thông báo. Ta phải sửa hết các lỗi rồi chuyển sang bước 3.

Bước 3: Chạy chương trình và thử (Ctrl+F9)

Nếu các dữ kiện được cung cấp chính xác mà chương trình cho kết quả sai thì ta phải xem lại thuật giải. Đây là lỗi đặc biệt nghiêm trọng vì nó không thể hiện ra ngoài qua các thông báo lỗi và có thể làm sai toàn bộ bài toán. Ta cũng phải quay lại bước 1 để sửa và chạy lại.

### Câu hỏi ôn tập chương I

- 1- Nêu quy tắc đặt tên trong Turbo Pascal. Cho 3 ví dụ tên đặt đúng, 3 ví dụ tên đặt sai.
- 2- Nêu cấu trúc chung của một chương trình Turbo Pascal.

## CHƯƠNG II CÁC KIỂU DỮ LIỆU CƠ SỞ VÀ CÁCH KHAI BÁO

### 1. Khái niệm dữ liệu, kiểu dữ liệu

#### 1.1. Khái niệm dữ liệu

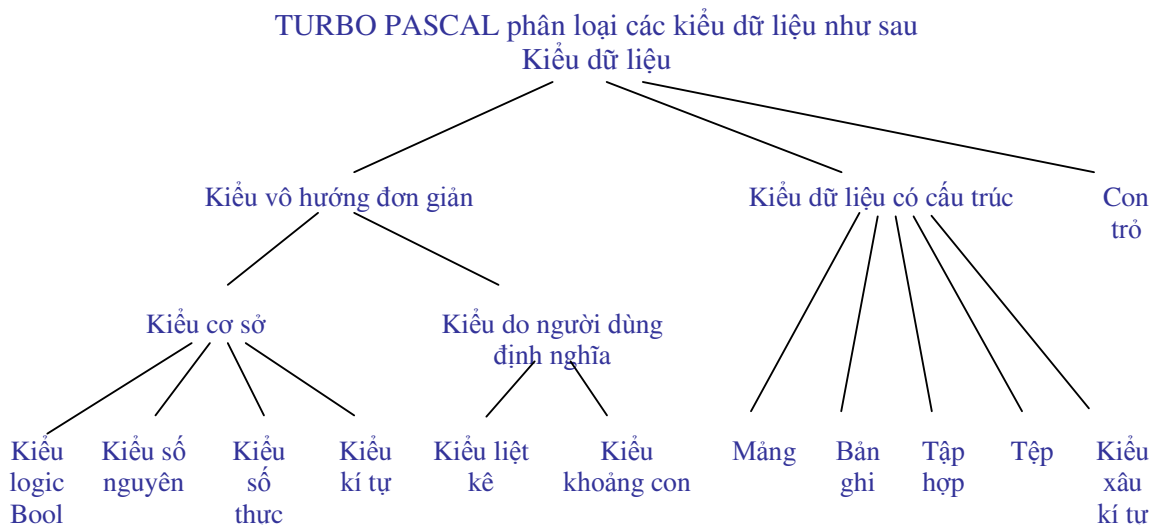
Dữ liệu là tất cả các thông tin có thật được máy tính xử lí. Chúng có nhiều dạng khác nhau, thể hiện qua các đối tượng cần xử lí như văn bản, số liệu, âm thanh, hình ảnh,...

#### 1.2. Kiểu dữ liệu

Kiểu dữ liệu là tập hợp tất cả các giá trị mà một đại lượng thuộc kiểu đó có thể nhận được, trên đó xác định một số phép toán.

Điều đó có nghĩa là một đại lượng (biến, hằng,...) chỉ có thể nhận một tập hợp các giá trị nhất định (chứ không phải mọi giá trị có thể có) và trên tập hợp các giá trị đó có xác định một số phép toán. Chẳng hạn kiểu số nguyên thông thường chỉ nhận các giá trị trong phạm vi  $-2^{15}$  đến  $2^{15}-1$ , trên đó có các phép toán số học  $+$ ,  $-$ ,  $*$ ,  $/$ ,  $\text{div}$ ,  $\text{mod}$  và các phép so sánh  $=$ ,  $<$ ,  $>$ .

#### 1.3. Phân loại các kiểu dữ liệu



### 2. Các kiểu dữ liệu đơn giản

#### 2.1. Kiểu số nguyên

Kiểu số nguyên bao gồm tất cả các số nguyên có thể biểu diễn được trên máy tính.

Từ khoá : Integer

Phạm vi biểu diễn: từ -32768 đến 32767, tức  $-2^{15}$  đến  $2^{15}-1$  ( dùng 2 byte=16 bit )

Như vậy kiểu integer chỉ là một đoạn của tập số nguyên

Các phép toán: Kiểu integer được thực hiện với tất cả các phép toán dành cho số nguyên thông thường

+ Các phép toán số học: + ( cộng ) , - ( trừ ) , \* ( nhân ) , / ( chia ) , DIV ( phép chia lấy phần nguyên, ví dụ: 7 div 3=2), MOD ( phép chia lấy phần dư, ví dụ: 7 mod 3=1)

+ Các phép so sánh: = ( bằng ) , < ( nhỏ hơn ) , <= ( nhỏ hơn hoặc bằng ) , > ( lớn hơn ) , >= ( lớn hơn hoặc bằng ) , <> ( khác )

Trong TURBO PASCAL, ngoài kiểu số nguyên được khai báo với từ khoá integer nói trên còn có các kiểu số nguyên khác ( xem bảng ) phục vụ cho các mục đích tính toán khác nhau

| Kiểu<br>( từ khoá ) | Phạm vi biểu diễn        | Kích thước (byte) |
|---------------------|--------------------------|-------------------|
| Byte                | 0..255                   | 1                 |
| Shortint            | -128..127                | 1                 |
| Integer             | -32768..32767            | 2                 |
| Word                | 0..65535                 | 2                 |
| Longint             | -2147483648.. 2147483647 | 4                 |

## 2.2. Kiểu số thực

Kiểu số thực là tập hợp tất cả các số thực có thể biểu diễn được trên máy tính. Không phải tất cả các số thực đều có thể biểu diễn được trên máy tính mà độ lớn và độ tinh xác của nó phụ thuộc vào máy và chương trình dịch.

- Kiểu số thực được khai báo với từ khoá real

- Phạm vi biểu diễn: từ  $1.9 \times 10^{-39}$  đến  $1.7 \times 10^{38}$

- Các phép toán: Kiểu real có thể thực hiện với tất cả các phép toán của số thực thông thường: +, -, \*, / và các phép so sánh ( không có phép div và mod dành cho số thực ).

- Cách viết số thực dạng dấu phẩy động:

Trong máy tính số thực được viết dùng dấu chấm để ngăn giữa phần nguyên và phần lẻ thập phân. Trong dạng dấu phẩy động số thực được viết theo dạng: AE+b, trong đó A gọi là phần định trị, E+b gọi là phần mũ.

Ví dụ: Ta xem xét các cách viết khác nhau của cùng 1 số

$824.12345 = 8.2412345 \times 10^2$  hay  $8.2412345E+2$

$= 824123.45 \times 10^{-3}$  hay  $824123.45E-3$

Trong TURBO PASCAL, ngoài kiểu số thực được khai báo với từ khoá real nói trên còn có các kiểu số thực khác ( xem bảng ) phục vụ cho các mục đích tính toán với độ lớn và độ chính xác khác nhau

| Kiểu<br>( từ khoá ) | Phạm vi biểu diễn    | Chữ số có nghĩa | Kích thước ( byte ) |
|---------------------|----------------------|-----------------|---------------------|
| Single              | 1.5E-45..3.4E+38     | 7-8             | 4                   |
| Real                | 2.9E-39..1.7E+38     | 11-12           | 6                   |
| Double              | 5.0E-324..1.7E+308   | 15-16           | 8                   |
| Extended            | 3.4E-4932..1.1E+4932 | 19-20           | 10                  |



\* Các hàm số học chuẩn dùng cho kiểu số nguyên và số thực: Đây là những hàm được định nghĩa sẵn trong TURBO PASCAL, ta có thể sử dụng như các công cụ để tính toán

| Tên hàm ( trong TP ) | Cho kết quả ( hàm )                   |
|----------------------|---------------------------------------|
| ABS(X)               | X ( giá trị tuyệt đối của X )         |
| SQR(X)               | X <sup>2</sup> ( X bình phương )      |
| SQRT(X)              | $\sqrt{X}$ ( căn bậc 2 của X )        |
| LN(X)                | lnX (Logarit Nepe)                    |
| EXP(X)               | e <sup>X</sup>                        |
| SIN(X)               | sinX ; X cho bằng radian              |
| COS(X)               | cosX; X cho bằng radian               |
| ARCTAN(X)            | arctangX                              |
| SUCC(n)              | n+1 ( số tiếp theo của n ) , n nguyên |
| PRED(n)              | n-1 ( số kè trước của n ) , n nguyên  |
| TRUNC(X)             | cắt, cho phần nguyên của X            |
| ROUND(X)             | Làm tròn phần lẻ của X                |

### 2.3. Kiểu kí tự

Kiểu kí tự là tập hợp tất cả các kí tự của bảng mã máy tính ( ASCII )

Từ khoá Char

Có thể sử dụng các phép toán so sánh đối với kiểu kí tự, kí tự nào đứng trước theo thứ tự của bảng mã máy tính sẽ được coi là nhỏ hơn, kí tự đứng sau được xem là lớn hơn

Ví dụ: 'a' < 'b'

Hàng kí tự được viết trong cặp dấu nháy ''

Các hàm dùng cho kiểu kí tự:

| Tên hàm ( trong TP ) | Cho kết quả ( hàm )                     |
|----------------------|-----------------------------------------|
| ORD(C)               | cho số thứ tự của kí tự C trong bảng mã |
| CHR(n)               | cho kí tự có số thứ tự là n             |
| SUCC(C)              | cho kí tự đứng kè sau                   |
| PRED(C)              | cho kí tự đứng kè trước                 |

### 2.4. Kiểu logic (Boolean)

Từ khoá Boolean

Các phép toán logic: NOT , AND, OR, XOR

Các phép toán so sánh: =, <, >, ...

Kiểu logic boolean chỉ có 2 giá trị : True và False

Người ta định nghĩa: False < True

## 2.5. Kiểu liệt kê (Enumerated scalar type)

Là kiểu cho phép người lập trình có thể tự định nghĩa ra các kiểu vô hướng bằng cách liệt kê các giá trị của kiểu vô hướng ra thông qua các tên do người lập trình tạo ra. Danh sách các giá trị này được đặt trong ngoặc đơn và được mô tả bằng một tên kiểu trong phần mô tả (Phần TYPE).

Từ khoá: **TYPE**

Ví dụ:

**TYPE**

Boolean=(False, True);

Color=(Red, Blue, Green, While, Black);

Một biến vô hướng có thể định nghĩa thông qua các kiểu đã được mô tả trong phần **TYPE** như sau:

**VAR**

Ketqua: Boolean;

Mau1, Mau2: Color;

Hoặc khai báo trực tiếp với mô tả kiểu dữ liệu:

**VAR**

Gioitinh: (Nam, nu);

Ngay: (Chunhat, Hai, Ba, Tu, Nam, Sau, Bay);

Ví dụ:

Ketqua:= True;

Mau1:=Blue;

Gioitinh:=Nam;

Ngay:=Chunhat;

## 2.6. Kiểu khoảng con (Sub- range Type)

Là kiểu vô hướng được dùng khi một biến chỉ được lấy giá trị trong một khoảng (xác định bởi cận trên và cận dưới).

- Quy tắc định nghĩa:

**Hang\_can\_duoi..Hang\_can\_tren;**

Trong đó: **Hang\_can\_duoi < Hang\_can\_tren**, và cùng kiểu.

- Tác dụng: - Tiết kiệm ô nhớ.  
- Có thể kiểm tra giá trị của biến khi chạy chương trình không được vượt ra khỏi giới hạn của khoảng con.

Ví dụ:

**TYPE**

Ngay= (Chunhat, Hai, Ba, Tu, Nam, Sau, Bay);

Chu\_cai\_hoa='A'..'Z';

**VAR**

ch: Chu\_cai\_hoa;

Ngay\_lam\_viec: Hai..Bay;

## 3. Các khai báo

### 3.1. Khai báo đơn vị

Đơn vị là một kiểu tổ chức của chương trình Pascal, trong đó có các chương trình con, nhằm thực hiện một một chức năng nào đó. Ta khai báo các đơn vị sử dụng trong chương trình theo cú pháp sau:

```
Uses Tên_đơn_vị;
```

CRT là đơn vị chuẩn quản lý màn hình, bàn phím; GRAPH là đơn vị đồ họa. PRINTER là đơn vị khai báo máy in, máy in được cài đặt tên là LST.

Ví dụ 1: Uses crt;

Ví dụ 2: Uses crt, graph, printer;

### 3.2. Khai báo nhãn

Nhãn là một tên trong Pascal. Nhãn dùng đánh dấu một câu lệnh trong chương trình để có thể chuyển điều khiển tới đó. Khai báo nhãn trong chương trình theo cú pháp sau:

```
Label Tên_nhãn;
```

Cách đánh dấu nhãn cho câu lệnh như sau:

```
Tên_nhãn : Câu_lệnh;
```

Ví dụ:

```
Label Tiếp;
```

```
...
```

```
Tiếp : Readln(x);
```

### 3.3. Khai báo hằng

Hằng là đại lượng không thay đổi giá trị trong quá trình xử lý. Ta khai báo hằng theo cú pháp:

```
const Tên_hằng=Giá trị ;
```

ở đó: Tên\_hằng do ta đặt ra, Giá trị là giá trị mà tên hằng sẽ nhận và giữ không đổi suốt quá trình xử lý.

```
Ví dụ: const n=10; { khai báo hằng số nguyên n=10}
```

```
pi=3.1416; { khai báo hằng số pi }
```

```
ki_tu='Y'; { khai báo hằng kí tự ki_tu nhận kí tự Y }
```

### 3.4. Khai báo kiểu

Khai báo kiểu là định nghĩa một kiểu dữ liệu do người sử dụng tạo ra. Khai báo kiểu có cú pháp sau:

```
Type Tên_kiểu = Mô_tả_kiểu;
```

```
Ví dụ: Type Mau = (do, xanh, vang, tim, nau);
```

```
Type Phuong_tien_GT = (xe_dap, xe_may, o_to, tau_hoa);
```

```
Type Tui_nguoi = 1..300;
```

### 3.5. Khai báo biến

Biến là đại lượng có thể thay đổi giá trị trong quá trình xử lý. Biến của chương trình là tên của ô nhớ cất giữ dữ liệu. Muốn sử dụng biến nào ta phải khai báo trước biến đó bằng cách viết tên biến trong phần khai báo ở đầu chương trình sau từ khoá Var.

Cú pháp:

Var Tên\_biến:Kiểu\_dữ\_liệu;

ở đó: Tên\_biến do ta đặt ra, Kiểu\_dữ\_liệu là tên của một trong các kiểu dữ liệu của TURBO PASCAL, dấu hai chấm bắt buộc phải có để ngăn giữa 2 phần của khai báo, dấu chấm phẩy kết thúc dòng khai báo. Các biến có cùng kiểu có thể được khai báo cùng nhau bằng cách viết các tên biến cách nhau bởi dấu phẩy (,), nhóm các biến khác kiểu nhau được viết cách nhau bởi dấu ;

Ví dụ: Var i:integer; { khai báo một biến có tên là I, biến này có kiểu nguyên }  
x,y,z:real; { khai báo 3 biến x,y,z nhận các giá trị có kiểu thực }  
t: char; { khai báo biến t có kiểu kí tự }  
kiem\_tra:boolean; { khai báo biến logic }  
Tuoi: 1..300;

Đối với mỗi bài toán, khi phân tích thuật toán, ta phải xác định rõ ngay chương trình cần phải nhập vào các dữ kiện gì ? các kết quả trung gian nào cần phải được chứa, dữ liệu ra là những gì?... để từ đó có phần khai báo chính xác, không bị thiếu mà cũng không thừa gây lãng phí ô nhớ (!)

## 4. Biểu thức và câu lệnh

### 4.1. Biểu thức

Biểu thức là một công thức tính toán để có một giá trị theo một công thức toán học nào đó. Một biểu thức bao gồm các toán tử kết hợp với các toán hạng.

Các toán tử trong TURBO PASCAL gồm: các phép toán số học (+, -, \*, /), các phép so sánh (=, <, >, ...), các phép toán logic (NOT, AND, OR);

Các toán hạng: gồm các hằng, các biến, các hàm được định nghĩa trong TURBO PASCAL

Thứ tự ưu tiên các phép toán như sau:

1. dấu ngoặc ( ) nội dung viết trong dấu ngoặc được ưu tiên thực hiện trước nhất
2. phép lấy đối ( - ), phép NOT
3. các phép tính loại nhân, chia: \*, /, DIV, MOD, AND
4. các phép tính loại cộng, trừ: +, -, OR
5. các phép so sánh: +, <>, <, >, ...

Chú ý: Biểu thức trong Pascal chỉ được viết trên 1 dòng do đó phải chú ý dùng các dấu ( ) để viết cho chính xác.

Ví dụ: Biểu thức toán học viết là: 
$$\frac{b^2 - 4ac}{1 + \sqrt{2ab}}$$

Trong Pascal viết là: (b\*b- 4\*a\*c)/(1+sqrt(2\*a\*b))

Biểu thức toán học viết là: 
$$\frac{2 + \sqrt{\sin x + \cos x}}{2 \sin x + 1}$$

Trong Pascal viết là: (2+ sqrt (sin(x)+cos(x)))/(2\* sin(x)+1)

## 4.2. Câu lệnh

Câu lệnh xác định công việc, thao tác mà chương trình phải thực hiện để xử lý các dữ liệu đã mô tả, khai báo. Các lệnh viết cách nhau bởi dấu (;), TURBO PASCAL phân loại các lệnh như sau:

- Lệnh đơn: là các lệnh không chứa các lệnh khác
- Lệnh có cấu trúc: thường chứa nhiều hơn một lệnh đơn, bao gồm khối lệnh, các lệnh thử và rẽ nhánh, các lệnh lặp.
- Lệnh hợp thành ( còn gọi là lệnh phức hay lệnh ghép ): gồm một nhóm các lệnh được đặt giữa cặp từ khoá begin ... end; ( dấu chấm phẩy (;) kết thúc chứ không phải dấu chấm ) như sau:

```
begin
 lệnh1;
 lệnh2;
 ...
 lệnhn;
end;
```

Dạng lệnh này cho ta thấy tính có cấu trúc của TURBO PASCAL. Một nhóm các lệnh giải quyết trọn vẹn một công việc nào đó sẽ được tổ chức trong một khối lệnh dạng này, và từng công việc của bài toán lớn sẽ được giao cho từng người làm trước khi ghép lại để giải bài toán tổng thể. Chúng ta sẽ đề cập kỹ hơn về vấn đề này ở phần chương trình con ( thủ tục và hàm ).

## Bài tập chương 2

Hãy viết các khai báo biến cho các bài toán sau:

1. Tính n giai thừa:  $n! = 1.2...n$  với  $n > 1$
2. Tính các tổng:  $S = 1/2 + 1/4 + ... + 1/(2k)$   
 $Q = 1.1! + 2.2! + ... + n.n!$
3. Tìm và in ra tất cả các số chính phương nhỏ hơn một số cho trước, cho biết có bao nhiêu số chính phương như vậy.
4. Viết chương trình giải bài toán cô: " Vừa gà vừa chó, bó lại cho tròn, ba mươi sáu con, một trăm chân chẵn. Hỏi có bao nhiêu gà, bao nhiêu chó?"
5. Viết chương trình tìm ước số chung lớn nhất của 2 số nguyên dương cho trước.
6. Tính  $E_x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + ... + \frac{x^n}{n!} + ...$  với độ chính xác  $\epsilon = 10^{-4}$  (  $ABS(x^n/n!) < \epsilon$  ), giá trị x được nhập vào từ bàn phím khi chạy chương trình.
7. Cần có 50000 đ từ các loại giấy bạc 1000đ, 2000đ và 5000đ. Tìm tất cả các phương án có thể.
8. Chuyển một số thập phân nguyên dương thành một số nhị phân, in ra màn hình dạng  
 $X_{10} = Y_2$

## CHƯƠNG III

### CÁC THỦ TỤC VÀO RA DỮ LIỆU VÀ CÁC CẤU TRÚC ĐIỀU KHIỂN

#### 1. Phép gán

Phép gán dùng để gán giá trị của một biểu thức cho một biến. Kí hiệu phép gán là :=

##### Cú pháp

Tên\_biến:=Biểu thức;

**Ý nghĩa:** Khi gặp lệnh này, trước hết máy sẽ tính giá trị của biểu thức ở vế phải rồi gán giá trị tính được cho biến ở vế trái.

Chú ý: kiểu dữ liệu của biểu thức phải phù hợp với kiểu dữ liệu của biến ở vế trái.

Ví dụ: Với các biến đã được khai báo thích hợp ở đầu chương trình, trong thân chương trình ta có thể viết

```
x:=5; nghĩ là biến x nhận giá trị 5
x:=x+1; giá trị của biến x được thay thế bởi giá trị của chính nó tăng thêm một
delta:=b*b-4*a*c; tính giá trị của delta trong chương trình giải phương trình bậc 2
doan:=true;
chu_cai:='A';
```

#### 2. Các thủ tục Vào- Ra dữ liệu

##### 2.1. Các thủ tục vào dữ liệu

Các biến của chương trình có thể nhận giá trị thông qua phép gán ở trong chương trình, tuy nhiên, để gán giá trị cho biến thông qua các thiết bị nhập chuẩn như bàn phím ta dùng một trong 2 thủ tục sau

```
read(danh sách các tên biến);
readln(danh sách các tên biến);
```

ý nghĩa: nhập giá trị cho từng biến trong danh sách tên biến bằng cách gõ vào từ bàn phím. Các biến trong danh sách các tên biến viết cách nhau bởi dấu phẩy (.). Khi nhập các giá trị cụ thể, các giá trị phải phù hợp với các biến trong danh sách về số lượng biến, kiểu của biến và vị trí các biến trong danh sách. Các giá trị số được viết cách nhau bởi ít nhất một dấu khoảng trắng (dấu cách).

Các biến trong thủ tục này cho phép là các kiểu sau: Nguyên, thực, kí tự, xâu kí tự, khoảng con,... Không được là kiểu logic.

Ví dụ: với các khai báo ở trên (bài 2.III), ta có thể viết

```
read(x,y,z);
readln(t);
```

...

khi chạy chương trình ta có thể gõ: 3 6 4 ↵  
N ↵

nghĩa là biến x nhận giá trị 3, biến y nhận giá trị 6, biến z nhận giá trị 4 còn biến t nhận giá trị N ( kiểu char).

Hai cách viết read và readln có ý nghĩa như nhau về tác dụng nhập dữ liệu, chỉ khác là sau khi thực hiện xong chức năng này thì lệnh readln sẽ đưa con trỏ về đầu dòng tiếp theo, còn lệnh read thì không.

Ngoài 2 cách viết trên còn có thủ tục readln; không có tham số cho phép dừng chương trình chờ gõ một phím bất kì trước khi tiếp tục.

Chú ý: Khi vào dữ liệu cho biến kí tự hoặc biến xâu kí tự ta dùng thủ tục Readln(..) và mỗi biến trong một thủ tục.

Ví dụ: Đầu chương trình khai báo các biến

Var t: string; p: char; i,j,k : integer;

Trong thân chương trình vào dữ liệu cho các biến trên có thể viết như sau:

Readln(t); readln(p); readln(i,j,k);

## 2.2. Các thủ tục ra dữ liệu

Có 2 cách viết dữ liệu ra màn hình:

write( các mục cần ghi ra );

writeln( các mục cần ghi ra );

- ý nghĩa: đưa ra màn hình các kết quả tính toán trong chương trình, giá trị các biến, hay tất cả những gì nằm giữa cặp dấu nháy ' ... '

- Ví dụ: write(x,y); In ra màn hình các giá trị x và y, kết quả trên màn hình là 36

Có thể viết thành

write(x); write(y);

hay rõ hơn

write('x=',x);write('y=',y); thì kết quả là x=3 y=6

các kí tự 'x=' và 'y=' nằm trong cặp dấu ngoặc chỉ có tác dụng trình bày, làm phân biệt rõ các giá trị được viết ra.

Hai cách viết trên có ý nghĩa như nhau về việc thể hiện dữ liệu ra màn hình, tuy nhiên thủ tục writeln sẽ chuyển con trỏ xuống đầu dòng tiếp theo sau khi thực hiện, con thủ tục write giữ nguyên vị trí con trỏ.

Ngoài 2 dạng trên còn có thủ tục writeln; không có tham số chỉ có tác dụng đưa ra một dòng trống không chứa gì.

## 2.3. Kết hợp read, readln và write, writeln để nhập dữ liệu

Ta có thể kết hợp các thủ tục trên để tạo ra dạng nhập dữ liệu sáng sủa và đẹp mắt theo mẫu: write(' câu nhắc nhập dữ liệu '); readln(biến);

Ví dụ:

write(' Hãy nhập dữ liệu cho biến x:'); hay write('x=');

readln(x);

readln(x);

kết quả sẽ là:

Hãy nhập dữ liệu cho biến x:( ta gõ ) 3 hay x= ( ta gõ ) 3

nên viết câu dẫn nhập gợi ý biến sẽ được nhập vào, không nên bỏ qua hoặc viết

writeln(' Hãy nhập dữ liệu cho biến x:'); hay writeln('x=');

read(x);

read(x);

vì như thế khi chạy chương trình con trỏ nhập sẽ nhảy xuống dòng dưới dòng hướng dẫn, nhìn không đẹp mắt thế này

Hãy nhập dữ liệu cho biến x: hay x=

( ta gõ ) 3

( ta gõ ) 3

## 2.4. Viết ra dữ liệu có quy cách

Một trong những yêu cầu khi trình bày dữ liệu ra màn hình là phải sáng sủa, đúng quy cách, dễ đọc, dễ hiểu. TURBO PASCAL có các quy định cho các kiểu dữ liệu như sau

### \*. *Viết ra kiểu số nguyên*

`write(i:n);` hoặc `writeln(i:n);`

trong đó I là số nguyên cần ghi ra, n là số chỗ dành để viết ra số nguyên đó trên màn hình, máy sẽ bố trí số nguyên từ phải sang trái, nếu thừa sẽ bỏ trống bên trái.

Ví dụ: với I=23, j=234 thì `writeln(i);writeln(i:5); writeln(j:5);` sẽ cho

```
23
 _ _ 23
 _ 234
```

Như vậy viết không có quy cách sẽ căn lề trái, trong khi viết có quy cách sẽ căn phải.

### \*- *Viết ra kiểu số thực*

mẫu 1:

`write(r:m:n);` hoặc `writeln(r:m:n);`

trong đó r là số thực cần ghi ra, m là số chỗ dành cho cả số thực ( kể cả dấu chấm ngăn cách phần nguyên và phần lẻ, nếu có) n là số chỗ dành cho phần lẻ thập phân.

mẫu 2:

`write(r:n);`

trong đó n là số chỗ cho cả số thực viết dưới dạng e-mũ

Ví dụ: với r=123.45

thì `writeln(r); writeln(r:8:3); writeln(r:7);` sẽ cho kết quả

```
1.234500000E+02 (10 chữ số dành cho phần lẻ)
123.450
1.2E+00 (có làm tròn)
```

### \*- *Viết ra kiểu kí tự*

Với kiểu kí tự, viết không quy cách sẽ cho ra kí tự bình thường, mỗi kí tự chiếm một chỗ, còn viết có quy cách thì các kí tự sẽ được bố trí từ phải sang trái, thêm các dấu khoảng cách vào bên trái nếu thừa chỗ.

Ví dụ: với t='Y' thì

`writeln(t); writeln(t:3); writeln('PASCAL'); writeln('PASCAL':8);`

sẽ cho

```
Y
 Y
PASCAL
 PASCAL
```

### \*- *Viết ra kiểu boolean*

kiểu boolean với 2 giá trị là TRUE và FALSE cũng được viết ra theo dạng

`write(ok);` hoặc `writeln(ok:n);`

trong đó ok là biến kiểu boolean, n là số chỗ để viết ra biến ok.

### *e- In ra máy in*

Các thủ tục `write` và `writeln` cũng dùng để đưa dữ liệu ra máy in. Muốn vậy, ở đầu chương trình phải có lời gọi chương trình chuẩn `USES PRINTER;` ở đầu chương trình và phải có thành phần `Lst` và dấu phẩy (,) đứng trước nội dung cần in ra.

Ví dụ: `writeln(Lst, 'kết quả là', S:8:2);`



**\*- Các thủ tục trình bày màn hình của TURBO PASCAL**

Đây là những thủ tục có sẵn của TURBO PASCAL phục vụ cho việc trình bày màn hình. Muốn dùng các thủ tục này phải có lời gọi chương trình chuẩn USES CRT; ở đầu chương trình. Các thủ tục đó là:

CLRSCR; xoá toàn bộ màn hình, chuyển con trỏ về góc trên bên trái của màn hình.

CLREOF; xoá các kí tự bên phải vị trí con trỏ hiện thời

GOTOXY(x,y); chuyển con trỏ màn hình đến điểm có tọa độ (x,y) ( màn hình văn bản có 25 dòng, 80 cột, trong thủ tục này x là tọa độ cột, y là tọa độ dòng )

TEXTCOLOR(Mau); thiết lập màu cho văn bản, Mau là số nguyên có thể nhận giá trị từ 0 trở lên, giá trị lớn nhất của Mau phụ thuộc loại màn hình của máy. Ta có thể chỉ định Mau bằng một chữ cái tiếng Anh chỉ màu ( xem bảng )

| Số nguyên chỉ màu | Tên màu      |
|-------------------|--------------|
| 0                 | Black        |
| 1                 | Blue         |
| 2                 | Green        |
| 3                 | Cyan         |
| 4                 | Red          |
| 5                 | Magenta      |
| 6                 | Brown        |
| 7                 | LightGrey    |
| 8                 | DarkGrey     |
| 9                 | LightBlue    |
| 10                | LightGreen   |
| 11                | LightCyan    |
| 12                | LightRed     |
| 13                | LightMagenta |
| 14                | Yellow       |
| 15                | White        |

TEXTBACKGROUND(Mau); xác lập màu nền cho văn bản.

Ví dụ: đoạn lệnh chương trình in dòng chữ THU DO HA NOI màu xanh trên nền vàng lên màn hình

```
uses CRT;
textbackground(14);
textcolor(blue);
write(' THU DO HA NOI');
```

## 2.5. Các chương trình ứng dụng những lệnh đã học

**\* Bài toán 1:** Tính diện tích và chu vi của hình tròn.

Phân tích bài toán :Trong bài toán này biến vào là R; biến ra là: s (diện tích), cv ( chu vi); các biến đều có kiểu số thực.

Tính toán theo công thức sau:

$$s = 3.14 * R^2 ;$$

$$cv = 2 * 3.14 * R.$$

Chương trình được viết như sau:

```
Program Dien_tich_chu_vi_hinh_tron;
Uses crt;
Var cv, s, r : real;
Begin
 Clrscr;
 Write(' hay nhap vao ban kinh r: ');
 Readln(r);
 S:= 3.14 * r*r;
 Cv:=2*3.14*r;
 Writeln(' Dien tich = ', s:8:2);
 Writeln(' Chu vi = ', cv:8:2);
 Readln;
End.
```

**\* Bài toán 2:** Bài toán tính lương, bảo hiểm xã hội, bảo hiểm y tế và tiền lĩnh của cán bộ công chức nhà nước. Theo quy định hiện nay thì lương = hệ số lương \* 350000;

bảo hiểm xã hội = 5% \* lương ; bảo hiểm y tế = 1% \* lương;

tiền lĩnh = lương - bảo hiểm xã hội - bảo hiểm y tế.

Phân tích bài toán: Trong bài toán này các biến vào là Ht ( họ và ten) kiểu xâu kí tự và biến HS (hệ số lương) kiểu số thực. Các biến ra là Ht, HS, lương, BHXH ( bảo hiểm xã hội) kiểu thực, BHYT ( bảo hiểm y tế) kiểu thực, TL ( tiền lĩnh ) kiểu thực.

Tính toán theo các công thức đã cho ở trên.

Trong chương trình có sử dụng các câu lệnh trình bày màn hình.

Chương trình như sau:

```
Program Tinh_luong;
Uses Crt;
Var HT:string[25]; LUONG, HS, BHXH, BHYT, TL:Real;
Begin
 Clrscr;
 Textcolor(red);
 Textbackground(Blue);
 Gotoxy(10,5);
 Write(' Nhap ho va ten: '); Readln(HT);
 Write(' Nhap he so luong '); Readln(HS);
 LUONG:= HS * 350000;
 BHXH:= LUONG * 0.05;
 BHYT:= LUONG * 0.01;
 TL:= LUONG-BHXH-BHYT;
 WRITELN(HT:25,HS:6:2,LUONG:10:1,BHXH:10:1,BHYT:10:1,TL:10:1);
 READLN;
END.
```

### 3. Các lệnh điều kiện

Phần này ta sẽ nghiên cứu các lệnh cho phép chương trình rẽ nhánh thực hiện một công việc dựa trên giá trị một điều kiện nào đó. Có 2 dạng lệnh điều kiện là câu lệnh điều kiện (cho phép rẽ tối đa 2 nhánh) và câu lệnh lựa chọn (cho phép rẽ nhiều nhánh).

#### 3.1. Câu lệnh điều kiện

##### a. Dạng lệnh

- Dạng đơn giản

IF <điều kiện> THEN <lệnh >;

trong đó

<điều kiện> là một biểu thức điều kiện liên hệ bởi các toán tử: =, <, >, <=, >= và (hoặc) các phép toán logic NOT, AND, OR

<lệnh > có thể là một câu lệnh đơn giản hoặc phức hợp.

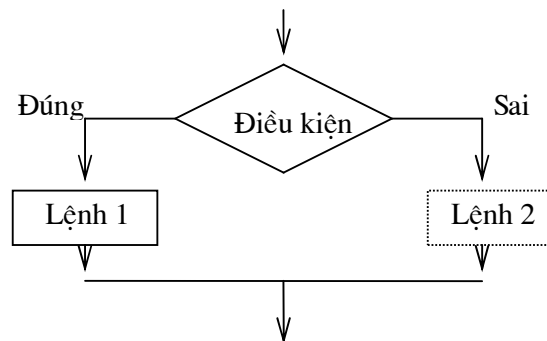
- ý nghĩa : khi gặp lệnh này, trước tiên máy sẽ kiểm tra <điều kiện>, nếu <điều kiện> là đúng thì sẽ cho thực hiện <lệnh >, còn nếu <điều kiện> sai thì máy sẽ bỏ qua <lệnh > và chuyển sang công việc tiếp theo.

- Dạng tổng quát

IF <điều kiện > THEN <lệnh1 > ELSE <lệnh2 >;

- ý nghĩa : khi gặp lệnh này, trước tiên máy cũng sẽ kiểm tra <điều kiện>, nếu <điều kiện> là đúng thì sẽ cho thực hiện <lệnh1 >, còn trái lại là <điều kiện> sai thì máy sẽ thực hiện <lệnh2 >.

Sơ đồ khối của câu lệnh điều kiện như sau



##### b. Ví dụ áp dụng

VD1: Viết chương trình nhập vào một số a tùy ý, rồi kiểm tra nếu a không âm thì in ra căn bậc 2 của a, trái lại in ra thông báo 'số âm không có căn bậc 2'

Chương trình được viết như sau:

```
Program canbac2;
Uses crt;
Var a: real;
Begin
 write(' Hay nhap vao so a:');
 readln(a);
 if a >= 0 then writeln(' Can bac hai cua a la:', sqrt(a))
 else writeln(' So am khong co can bac hai');
 Readln;
End.
```

VD2: Viết chương trình giải phương trình bậc hai  $ax^2+bx+c=0$

```

Program GPTB2;
Uses crt;
var
 a,b,c,x1,x2,dta:real;
Begin
 clrscr;
 gotoxy(5,5);
 writeln('CHUONG TRINH GIAI PHUONG TRINH BAC HAI');
 writeln;
 write('Vao cac he so:');
 readln(a,b,c);
 dta:=b*b-4*a*c;
 if dta>0 then
 begin
 writeln('Phuong trinh co 2 nghiem thuc:');
 X1:=(-b+sqrt(dta))/(2*a);
 X2:=(-b-sqrt(dta))/(2*a);
 writeln('X1=',X1:6:2);
 writeln('X2=',X2:6:2);
 end;
 if dta=0 then
 writeln('Phuong trinh co nghiem kep X=',-b/(2*a):6:2);
 if dta<0 then
 begin
 writeln('Phuong trinh co 2 nghiem phuc:');
 writeln('X1=',-b/(2*a):6:2,'-',sqrt(-dta):6:2,'i');
 writeln('X2=',-b/(2*a):6:2,'+',sqrt(-dta):6:2,'i');
 end;
 readln;
End.

```

Lưu ý: - Câu lệnh phức hợp: trong chương trình có những chỗ TURBO PASCAL chỉ cho phép viết một lệnh, nhưng ta lại muốn viết nhiều hơn một lệnh thành phần thì các lệnh thành phần đó phải được đặt trong cặp từ khoá Begin ... end; (dấu chấm phẩy kết thúc). Chẳng hạn trong ví dụ trên, ứng với trường hợp  $dta > 0$  máy phải thực hiện cả 5 lệnh trong cặp từ khoá begin...end; sau từ khoá then, còn ứng với trường hợp  $dta = 0$  thì máy chỉ phải thực hiện đúng một lệnh, do đó không cần dùng câu lệnh phức hợp.

- Các câu lệnh điều kiện có thể viết lồng nhau. Tức là các <lệnh1> và <lệnh2> lại có thể là câu lệnh điều kiện dạng IF...THEN IF...THEN...ELSE... khi đó cần lưu ý IF nào đi với THEN nào. Chẳng hạn xem đoạn chương trình phân loại kết quả học tập sau:

```

IF diem>=5 THEN
 IF diem>=7 THEN loai:='Kha gioi' ELSE loai:='TB'
ELSE
 IF diem>=3 THEN loai:='Yeu' ELSE loai:='Kem';

```

### 3.2. Câu lệnh lựa chọn: CASE ... OF...

Câu lệnh điều kiện chỉ cho phép ta thực hiện rẽ 2 nhánh ứng với hai giá trị đúng hay sai của biểu thức điều kiện. Để có thể thực hiện rẽ nhiều nhánh ứng với nhiều giá trị khác nhau của một biểu thức, ta phải sử dụng câu lệnh lựa chọn.

#### a. Dạng lệnh

Dạng đơn giản:

```
CASE <biểu thức> OF
 hằng1: <lệnh1>;
 hằng2: <lệnh2>;
 ...
 hằngn: <lệnhn>;
END;
```

Dạng tổng quát:

```
CASE <biểu thức> OF
 hằng1: <lệnh1>;
 hằng2: <lệnh2>;
 ...
 hằngn: <lệnhn>
ELSE <lệnh n+1>;
```

END;

Trong đó <biểu thức> và các hằng phải có *cùng kiểu* và phải là các kiểu vô hướng đếm được (các kiểu số nguyên kiểu kí tự, kiểu Boolean đã biết).

ý nghĩa:

Khi gặp câu lệnh rẽ nhánh, <biểu thức> đã được nhận một giá trị nào đó, giá trị này là một hằng (số hoặc kí tự). Nếu <biểu thức> nhận giá trị là hằng nào thì <lệnh> tương ứng với nó sẽ được thực hiện. Còn nếu <biểu thức> nhận giá trị không rơi vào hằng nào thì máy sẽ bỏ qua lệnh đối với dạng đơn giản, máy sẽ thực hiện <lệnh n+1> sau từ khoá ELSE đối với dạng tổng quát.

Lưu ý: Từ khoá END với dấu chấm phẩy (;) ở trong câu lệnh này để chỉ rằng kết thúc câu lệnh lựa chọn chứ không phải kết thúc chương trình con.

#### b. Ví dụ áp dụng

VD1: Viết chương trình xem thời khoá biểu của một ngày trong tuần. Yêu cầu: máy in ra câu hỏi 'Bạn muốn xem thời khoá biểu của thứ mấy?' ta gõ vào ngày thứ (của tuần) muốn xem và máy sẽ in ra thời khoá biểu của ngày hôm đó.

Chương trình được viết như sau: (giả sử xem thời khoá biểu của một lớp phổ thông, bạn đọc có thể hiệu chỉnh theo ý mình)

```
Program XEM_TKB;
Uses crt;
Var thu: 2..7; {biến thu để chứa các thứ ngày trong tuần}
Begin
 write(' Ban muon xem thoi khoa bieu cua thu may?');
 readln(thu);
 CASE thu OF
 2:writeln('Toan Ly Van');
 3:writeln('Sinh Ki Hoa');
 4: writeln('Toan Hoa Ly');
 5: writeln('Van Sinh The duc');
 6: writeln('Su Dia Chinh tri');
 7: writeln('Van Toan Sinh hoat');
 END;
 Readln;
End.
```

Bạn đọc có thể áp dụng câu lệnh dạng tổng quát để mở rộng chương trình xem thời khoá biểu này sao cho khi gõ vào một số nào đó không phải là thứ trong tuần (2..7) máy sẽ in ra thông báo rằng đó là ngày nghỉ.

Lưu ý:

- Sau mỗi hàng máy chỉ thực hiện đúng một lệnh. Do đó nếu muốn dùng nhiều hơn một lệnh thì ta phải sử dụng câu lệnh phức hợp.

- Các hàng có thể viết gộp lại với nhau. Chẳng hạn nếu thứ 3 và thứ 5 có cùng thời khoá biểu thì thay vì phải viết 2 dòng lệnh, ta có thể viết:

```
3,5: writeln('Sinh Ki Hoa');
```

- Các câu lệnh lựa chọn có thể lồng nhau. Tức là trong câu lệnh lựa chọn lại có thể chứa câu lệnh lựa chọn khác.

#### 4. Các lệnh lặp

Trong lập trình giải quyết các bài toán, ta có thể gặp trường hợp chương trình phải thực hiện lặp đi lặp lại một công việc nào đó. Việc thực hiện lặp phải theo một quy luật nào đó. Số lần lặp có thể được xác định trước hoặc không xác định. Sau đây ta sẽ lần lượt xét các lệnh lặp dạng này.

### 4.1- Lệnh lặp có số lần lặp xác định

Đây là một câu lệnh có cấu trúc cho phép thực hiện lặp đi lặp lại một công việc (đoạn lệnh) nào đó với số lần thực hiện được xác định trước. Để dễ hiểu trước hết ta xét một ví dụ đơn giản sau:

Giả sử ta phải viết ra các số từ 1 đến 10, mỗi số chiếm một dòng trên màn hình. Ta có thể thực hiện công việc này bởi 10 lệnh writeln như sau:

```
writeln(1);
writeln(2);
...
writeln(10);
```

Đây là cách viết dài dòng và đơn điệu trong khi ta có thể thực hiện công việc trên bởi một dòng lệnh ngắn gọn:

```
FOR I:=1 TO 10 DO writeln(I);
```

ý nghĩa của câu lệnh này là: cho một biến nhận giá trị nguyên I chạy từ 1 đến n ( biến I lần lượt nhận các giá trị số nguyên từ 1 đến n), với mỗi giá trị của I máy sẽ thực hiện công việc sau từ khoá DO là in ra giá trị của I.

Chi tiết các bước thực hiện của vòng lặp FOR này như sau:

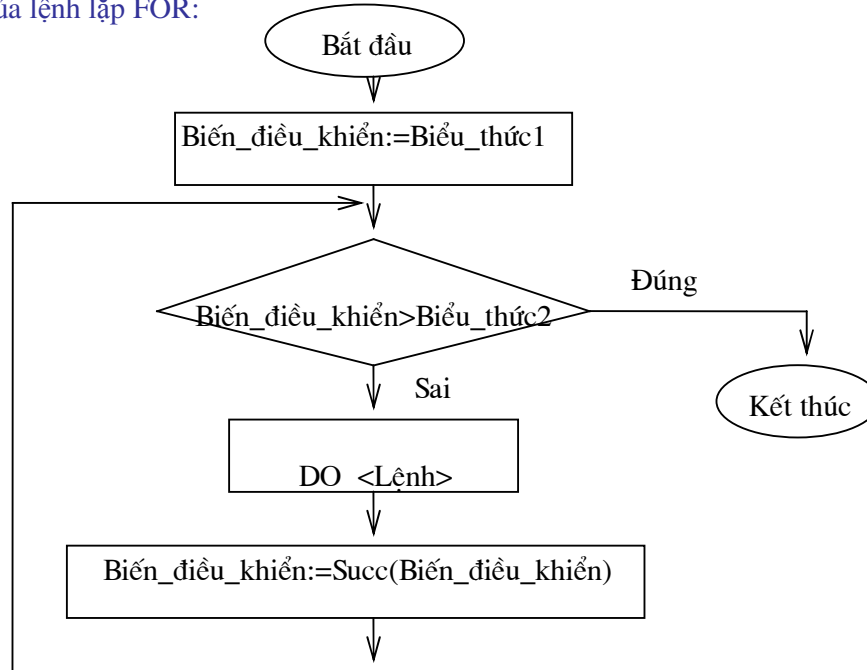
Đầu tiên biến I nhận giá trị khởi đầu là 1 (do lệnh gán I:=1), máy kiểm tra giá trị này không vượt quá giá trị cuối là 10 nên cho thực hiện lệnh writeln(I) viết ra giá trị 1. Sau đó biến I được tăng thêm một đơn vị, tức là I:=I+1. Bây giờ I=2, chưa vượt qua giá trị cuối là 10 nên lệnh writeln(I) lại được thực hiện để in ra giá trị của I là 2 lên màn hình. Rồi I lại được tăng lên... giá trị cuối cùng của I để lệnh writeln(I) được thực hiện là I:=10. Quá trình kết thúc khi I nhận giá trị 11 vượt quá giá trị cuối 10.

Câu lệnh trên là một ví dụ đơn giản của lệnh tổng quát sau:

#### a. Dạng lệnh lặp tăng (dạng tiến)

```
FOR Biến_điều_khiển:=Biểu_thức1 TO Biểu_thức2 DO <Lệnh>;
```

Trong đó *Biến\_điều\_khiển*, *Biểu\_thức1* và *Biểu\_thức2* phải cùng kiểu dữ liệu và phải là các kiểu vô hướng đếm được (như các kiểu số nguyên, kiểu kí tự...) ý nghĩa của câu lệnh này như sau: Trước tiên *Biến\_điều\_khiển* nhận giá trị khởi tạo là *Biểu\_thức1*, máy kiểm tra nếu giá trị đó không vượt quá *Biểu\_thức2* thì cho thực hiện <Lệnh>. Thực hiện xong lệnh này, *Biến\_điều\_khiển* được tăng thêm một đơn vị, máy lại kiểm tra với *Biểu\_thức2* để thực hiện <Lệnh>... Quá trình tiếp diễn đến khi *Biến\_điều\_khiển* nhận giá trị vượt quá *Biểu\_thức2* thì dừng. Số lần thực hiện lặp đi lặp lại <Lệnh> được xác định trước bởi giá trị của *Biểu\_thức1* và *Biểu\_thức2* nên câu lệnh trên còn được gọi là lệnh lặp có số lần lặp xác định. Lưu đồ của lệnh lặp FOR:



**\* Ví dụ**

VD1: Viết chương trình tính tổng của n số tự nhiên đầu tiên  $S=1+2+\dots+n$  theo phương pháp cộng dồn.

Phương pháp cộng dồn được thực hiện như sau: ban đầu ta khởi tạo  $S:=0$ , rồi lấy S cộng với 1, rồi cộng với 2,... cộng đến n. Như vậy ta sẽ phải sử dụng một vòng lặp FOR với một biến điều khiển I chạy từ *Biểu\_thức1* là 1 đến *Biểu\_thức2* là n. Ta sẽ dùng chính giá trị của biến điều khiển này để tính tổng S.

Chương trình được viết như sau:

```

Program Tinh_tong;
Uses crt;
Var i,n:integer;
 S:real;
Begin
 Write('Cho biet gia tri cua n:');
 Readln(n);
 S:=0;
 For I:=1 to n do S:=S+i;
 Writeln(' Tong tinh duoc la S:',S:10:2);
End

```

```

Readln;
End.

```

VD2: Một người gửi tiền tiết kiệm với số tiền ban đầu là A đồng, thời gian gửi t tháng, lãi suất tháng là p. Hỏi sau t tháng gửi người đó thu được số tiền lãi là bao nhiêu đồng?

Phân tích bài toán: Ta biết rằng gửi tiết kiệm thì số tiền thu được sau mỗi tháng sẽ bằng số tiền tháng trước cộng với lãi của tháng đó. Nếu gọi S là số tiền thu được sau mỗi tháng, t là số tháng gửi thì ta có:

Ban đầu  $S_0=A$

$t=1 \rightarrow S_1=S_0 + S_0*p$  (gửi 1 tháng)

$t=2 \rightarrow S_2= S_1 + S_1*p$  (gửi 2 tháng)

...

$t=n \rightarrow S_n= S_{n-1} + S_{n-1}*p$  (gửi n tháng)

Đây chính là quy luật để thực hiện vòng lặp tính số tiền thu được. Chương trình được viết như sau:

```

Program Tinh_tien_lai;
Uses crt;
Var i,t:integer; A,P,S:real;
Begin
 Write('Cho biet so tien gui ban dau: '); Readln(A);
 Write('Cho biet so thang gui: '); Readln(t);
 Write('Cho biet lai suat theo thang: '); Readln(P);
 S:=A;
 For I:=1 to t do S:=S+ S*P;
 Writeln(' Lai thu duoc la : ',S-A:10:2);

Readln;
End.

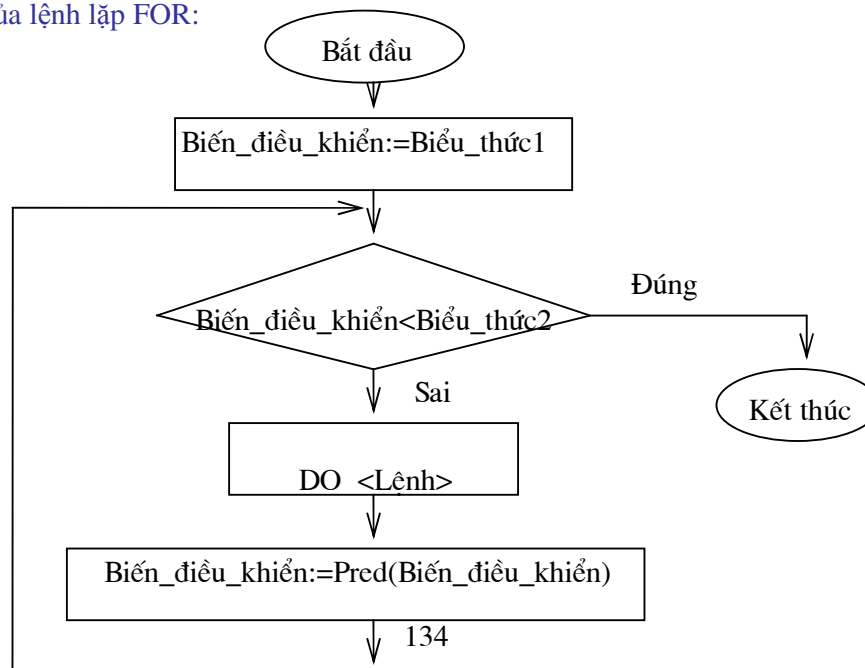
```

### b. Dạng lệnh lặp giảm (dạng lùi)

*FOR* *Biến\_điều\_khiển*:=*Biểu\_thức1* *DOWNTO* *Biểu\_thức2* *DO* <*Lệnh*>;

ý nghĩa hoàn toàn tương tự như dạng trên, chỉ khác là *Biến\_điều\_khiển* nhận giá trị giảm dần chứ không phải tăng dần, và điều kiện kiểm tra là *Biến\_điều\_khiển* không nhỏ quá *Biểu\_thức2*.

Lưu đồ của lệnh lặp FOR:





Với dạng này thì câu lệnh FOR trong ví dụ 1 ở trên được viết là:

```
For I:=n downto 1 do S:=S+i;
```

Tổng S sẽ được cộng dồn theo thứ tự từ n đến 1, tức là  $S=n+(n-1)+\dots+2+1$ .

Ví dụ: Viết chương trình in ra các kí tự trong bảng mã ASCII theo thứ tự giảm dần.

Các kí tự trong bảng mã ASCII có mã giảm dần từ 255 đến 0. Ta dùng hàm CHR(n) để nhận được kí tự.

Chương trình được viết như sau:

```
Program In_cac_ki_tu;
Uses crt;
Var I:integer;
Begin
 Clrscr;
 For i:=255 downto 0 do write(chr(i):4);
 Readln;
End.
```

### c. Các lưu ý

1) Sau từ khoá DO chỉ được viết một lệnh, do đó nếu cần thực hiện nhiều hơn một lệnh đơn thì phải sử dụng câu lệnh phức hợp

2) Các lệnh lặp có thể lồng nhau, chẳng hạn dạng:

```
FOR I:=1 TO n DO
 FOR J:=1 TO m DO
 <Lệnh>
```

Khi đó với mỗi giá trị của biến điều khiển của vòng lặp ngoài thì biến điều khiển của vòng lặp trong sẽ chạy hết các giá trị của nó. Tức là với mỗi giá trị của I thì J sẽ chạy từ 1 tới m. Xét ví dụ đoạn chương trình sau:

```
For I:= 1 to 2 do
 For J:=1 to 3 do
 begin
 k:=i+j;
 writeln(k);
 end;
```

sẽ cho kết quả là:

```
2
3
4
3
4
5
```

3) <Lệnh> sau từ khoá DO không được tùy tiện thay đổi giá trị của Biến\_điều\_khiển, làm như vậy ta có thể không kiểm soát được giá trị của Biến\_điều\_khiển và có thể làm rối vòng lặp.

Ví dụ: In ra tất cả các chữ số có 3 chữ số mà tổng các chữ số chia hết cho 3.

Trong bài toán này mỗi số đều có 3 chữ số, chữ số hàng trăm có thể nhận giá trị từ 1 đến 9, chữ số hàng chục, chữ số hàng đơn vị đều có thể nhận giá trị từ 0 đến 9. Để quét hết các chữ số có 3 chữ số ta dùng 3 vòng lặp lồng nhau, sau đó ta kiểm tra điều kiện nếu tổng các chữ số chia hết cho 3 thì in ra số đó.

Chương trình được viết như sau:

```

Program In_cac_so;
Uses crt;
Var t,c,v:integer;
Begin
 Clrscr;
 For t:=1 to 9 do
 For c:=0 to 9 do
 For v:=0 to 9 do
 If (t+c+v) mod 3 = 0 then writeln(t,c,v);
 Readln;
 End.
 End.

```

## 4.2- Lệnh lặp có số bước lặp không xác định

ở trên ta đã xét lệnh lặp có số bước lặp xác định, tức là số lần thực hiện công việc lặp đi lặp lại là được định trước. Nhưng trong khi lập trình có những bài toán yêu cầu thực hiện một công việc nào đó mà số lần lặp là không thể xác định trước được, mà phụ thuộc vào một biểu thức điều kiện nào đó. Có hai dạng lệnh lặp với số bước lặp không xác định.

### a. Lệnh lặp với điều kiện trước

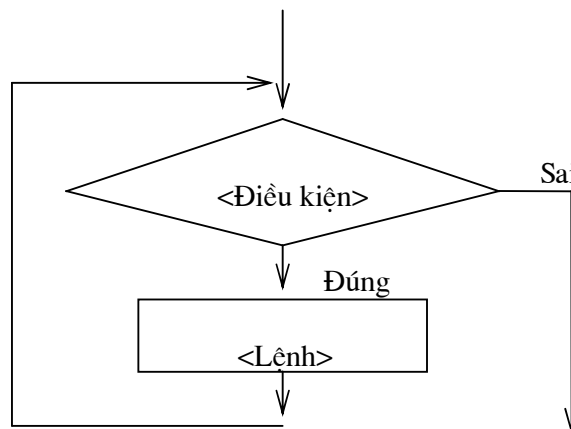
- Dạng lệnh:

WHILE <điều kiện> DO <Lệnh>;

ý nghĩa:

Khi gặp câu lệnh này, trước tiên máy sẽ kiểm tra <điều kiện>. Nếu <điều kiện> có giá trị đúng (TRUE) thì <Lệnh> được thực hiện. Thực hiện xong lệnh này máy sẽ quay lại kiểm tra <điều kiện> để thực hiện <Lệnh>... quá trình tiếp diễn đến khi <điều kiện> sai thì dừng.

Sơ đồ



- Ví dụ:

VD1: Một người gửi tiết kiệm không kì hạn với số tiền ban đầu là A đồng. Hỏi sau bao nhiêu tháng người đó thu được số tiền là B đồng, biết rằng lãi suất là 1.8%?

Phân tích bài toán: Ta biết rằng gửi tiết kiệm không kì hạn thì số tiền thu được sau mỗi tháng sẽ bằng số tiền tháng trước cộng với lãi của tháng đó. Nếu gọi S là số tiền thu được sau mỗi tháng, t là số tháng gửi, ls là lãi suất thì ta có:

nếu  $t=0 \rightarrow S_0=A$  (gửi 0 tháng)

nếu  $t=1 \rightarrow S_1=S_0 + S_0*ls$  (gửi 1 tháng)

nếu  $t=2 \rightarrow S_2= S_1 + S_1*ls$  (gửi 2 tháng)

...

Đây chính là quy luật để thực hiện vòng lặp tính số tiền thu được. Như vậy ta phải so sánh số tiền thu được S sau mỗi tháng với số tiền cần đạt B: nếu S đạt hoặc vượt B thì ta có kết luận của bài toán. Đây cũng là điều kiện dừng vòng lặp. Ta sẽ sử dụng một biến đếm t để đếm số lần thực hiện lặp, cũng chính là số tháng gửi.

Chương trình được viết như sau:

```

Program Gui_tiet_kiem;
Const ls=0.018;
Var A,B,S:real;
 t:integer;
Begin
 Clrscr;
 Write('Cho biet so tien gui ban dau:');
 Readln(A);
 Write('Cho biet so tien can dat :');
 Readln(B);
 S:=A;
 t:=0;
 WHILE S<B DO
 begin
 S:=S+S*ls;
 t:=t+1;
 end;
 Writeln(' Ban can gui it nhat la ', t, ' thang');
 Readln;
End.

```

VD2: Ta có thể viết lại chương trình tính tổng  $S=1+2+\dots+n$  sử dụng lệnh lặp WHILE... DO... như sau:

```

Program Tinh_tong1;
Var i,n:integer;
 S:real;
Begin
 Write('Cho biet gia tri cua n:');
 Readln(n);
 S:=0;
 i:=1;
 While i<= n do
 begin
 S:=S+i;
 i:=i+1;
 end;
End.

```

```
WriteLn(' Tong tinh duoc la S:',S:10:2);
```

End.

Như vậy ta có thể dùng lệnh lặp có số bước lặp không xác định để thay cho lệnh lặp có số bước lặp xác định. Tuy nhiên khi đó biến điều khiển vòng lặp không tự động thay đổi mà ta phải đưa nó vào thân vòng lặp (biến i trong VD trên).

### b. Lệnh lặp với điều kiện sau

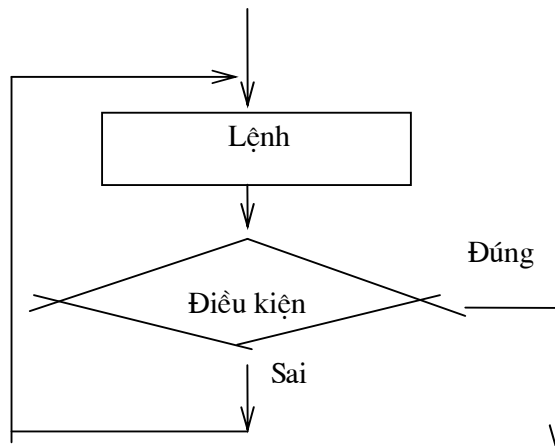
- Dạng lệnh:

```
REPEAT <Lệnh> UNTIL <điều kiện>;
```

ý nghĩa:

Khi gặp câu lệnh này, trước tiên máy sẽ cho thực hiện <Lệnh>, sau đó kiểm tra <điều kiện>. Nếu <điều kiện> có giá trị sai (FALSE) thì quay lại thực hiện <Lệnh>. Thực hiện xong lệnh này máy sẽ lại kiểm tra <điều kiện> để thực hiện <Lệnh>... quá trình tiếp diễn đến khi <điều kiện> đúng thì dừng.

Sơ đồ



- Ví dụ: Lấy lại VD gửi tiết kiệm ở trên. Bây giờ ta sẽ viết lại chương trình sử dụng dạng lệnh lặp với điều kiện sau. Chương trình chỉ cần thay đổi một chút ở lệnh lặp.

```
Program Gui_tiet_kiem;
Const Is=0.018;
Var A,B,S:real;
t:integer;
Begin
Write('Cho biet so tien gui ban dau:');
ReadLn(A);
Write('Cho biet so tien can dat :');
ReadLn(B);
S:=A;
t:=0;
REPEAT
S:=S+S*Is;
t:=t+1;
UNTIL S>=B;
WriteLn(' Ban can gui it nhat la ', t, ' thang');
ReadLn;
```

End.

Ta thấy rằng trong phần thân của vòng lặp các lệnh đặt giữa REPEAT và UNTIL không cần sử dụng câu lệnh phức hợp begin...end;

Bạn đọc có thể sử dụng lệnh này để viết lại chương trình tính tổng ở trên.

### c. Các lưu ý khi sử dụng các lệnh lặp WHILE và REPEAT

1) Trong thân vòng lặp phải có ít nhất một lệnh làm thay đổi giá trị của biểu thức điều kiện nhằm dừng vòng lặp. Nếu điều kiện luôn được thoả mãn thì lệnh có thể rơi vào vòng lặp vô tận. Khi đó chỉ còn cách là tắt máy hoặc dừng chương trình.

Xem ví dụ sau:

```
I:=1;
WHILE I<2 DO write('Stop!');
```

Vòng lặp này sẽ chạy vô tận vì trong thân vòng lặp sau từ khoá DO chỉ có một lệnh in ra xâu chữ 'Stop!', không có lệnh nào làm thay đổi giá trị của I. Tức là với I được khởi tạo ban đầu là 1 luôn nhỏ hơn 2, điều kiện luôn thoả mãn.

2) Trong lệnh lặp với điều kiện trước WHILE thì thân vòng lặp (sau từ khoá DO) phải sử dụng câu lệnh phức hợp begin... end; còn trong lệnh lặp với điều kiện sau thì không cần.

3) Trong lệnh lặp với điều kiện trước, máy luôn kiểm tra điều kiện trước khi thực hiện lệnh, do đó nếu điều kiện sai ngay từ đầu thì lệnh không được thực hiện lần nào. Còn trong lệnh lặp với điều kiện sau, máy luôn thực hiện lệnh một lần rồi mới kiểm tra điều kiện, dù ban đầu điều kiện có thể đúng. Nhớ rằng trong lệnh lặp với điều kiện sau lệnh được quay lại thực hiện chỉ khi điều kiện sai.

4) Các lệnh lặp có thể viết lồng nhau

5) Người ta thường dùng các lệnh lặp có số bước lặp không xác định để quay vòng thực hiện nhiều lần cả một công việc nào đó hoặc cả chương trình, cho phép tạo sự tương tác giữa người sử dụng và máy. Xem ví dụ mẫu sau:

```
Program VD;
Var tieptuc:char;
... { các biến của chương trình }
Begin
Repeat
 ...
 { đoạn chương trình thực hiện công việc và thay đổi điều kiện kiểm tra }
write(' Ban co tiep tục nua hay không (C/K)');
readln(tieptuc);
UNTIL (tieptuc='k') or (tieptuc='K');
```

End;

### d. Lệnh nhảy vô điều kiện GOTO

Lệnh GOTO cho phép chương trình nhảy vô điều kiện tới một vị trí nào đó thông qua tên nhãn. Nhãn là một số nguyên hoặc một tên được khai báo trong phần LABEL ở đầu chương trình. Trong chương trình nhãn được đặt vào vị trí phù hợp kèm theo một dấu hai chấm (:).

Xem ví dụ sau:

```
Program VDNHAN;
Uses crt;
Label N1,N2;
Var a,b,s:real;
```

```

Begin
 a:=1; b:=2;
 N1: a:=a+1;
 if a<5 then GOTO N1;
 N2: s:=a+b;
 if s<10 then GOTO N2;
 writeln('a=', a, 'b=',b);
 readln;
End.

```

Thường người ta ít dùng lệnh GOTO trong chương trình PASCAL vì nó sẽ phá vỡ tính có cấu trúc của chương trình PASCAL. Nếu có thể, ta nên tránh dùng lệnh này.

### Bài tập chương III

Viết chương trình cho các bài toán sau:

- Tính n giai thừa:  $n! = 1.2...n$  với  $n > 1$
- Tính các tổng:  $S = 1/2 + 1/4 + \dots + 1/(2k)$   
 $Q = 1.1! + 2.2! + \dots + n.n!$
- Tìm và in ra tất cả các số chính phương nhỏ hơn một số cho trước, cho biết có bao nhiêu số chính phương như vậy.
- Viết chương trình giải bài toán cô: " Vừa gà vừa chó, bó lại cho tròn, ba mươi sáu con, một trăm chân chẵn. Hỏi có bao nhiêu gà, bao nhiêu chó?"
- Viết chương trình tìm ước số chung lớn nhất của 2 số nguyên dương cho trước.
- Tính  $E_x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \dots + \frac{x^n}{n!} + \dots$  với độ chính xác  $\epsilon = 10^{-4}$  ( $ABS(x^n/n!) < \epsilon$ ), giá trị x được nhập vào từ bàn phím khi chạy chương trình.
- Cần có 50000 đ từ các loại giấy bạc 1000đ, 2000đ và 5000đ. Tìm tất cả các phương án có thể.
- Chuyển một số thập phân nguyên dương thành một số nhị phân, in ra màn hình dạng  $X_{10} = Y_2$
- Tính tích phân xác định của một hàm số trên một đoạn cho trước
- Viết chương trình tìm và in ra màn hình các số nguyên tố nhỏ hơn một số cho trước.

## CHƯƠNG IV

### Kiểu dữ liệu có cấu trúc: Kiểu mảng, Kiểu xâu kí tự, Kiểu tập hợp

#### 1 - Kiểu mảng

##### 1.1- Khái niệm mảng (array)

**a. Khái niệm mảng :** Mảng là một kiểu dữ liệu có cấu trúc bao gồm một số xác định các phần tử có cùng kiểu, có một tên chung. Các phần tử của mảng được truy nhập thông qua các chỉ số.

Trong khái niệm này ta cần chú ý 2 điểm sau:

- Số phần tử của mảng phải là một số xác định, không được khai báo là biến. Ta có thể khai báo bằng một giá trị cụ thể chẳng hạn như 5,10,20,...; hoặc có thể khai báo là hằng.
- Các phần tử của mảng phải cùng kiểu. Có thể là các kiểu đơn giản hoặc kiểu có cấu trúc.

**Ví dụ :** Mảng A gồm 6 phần tử là các số nguyên: A[1] A[2] A[3] A[4] A[5] A[6] .  
Như vậy tên mảng là A, các chỉ số là 1,2,3,4,5,6.

**b. Công dụng :** Mảng là dùng để lưu trữ một dãy dữ liệu có cùng một tính chất. Ví dụ như họ tên của các thí sinh trong 1 lớp, lương của các nhân viên trong 1 cơ quan,...

Trong bộ nhớ của máy tính các phần tử của mảng được lưu trữ bởi các từ máy kế tiếp nhau. Trong ví dụ trên mảng A được lưu trữ trong bộ nhớ bằng 6 từ máy kế tiếp nhau, mỗi từ máy có độ dài là 2 bytes.

##### 1.2- Khai báo mảng

Để khai báo mảng dùng cụm từ sau:

**ARRAY [ Kiểu\_chỉ\_số 1, Kiểu\_chỉ\_số 2, ... ] OF Kiểu\_phần\_tử ;**

- Khai báo bằng định nghĩa kiểu

TYPE

Tên\_kiểu\_mảng = ARRAY [ Kiểu\_chỉ\_số1, Kiểu\_chỉ\_số2, ... ] OF Kiểu\_phần\_tử ;

**VAR**

Tên\_biến\_mảng : Tên\_kiểu\_mảng ;

- Khai báo biến mảng trực tiếp qua khai báo VAR:

**VAR**

Tên\_biến\_mảng : ARRAY [ Kiểu\_chỉ\_số1, Kiểu\_chỉ\_số2, ... ] OF Kiểu\_phần\_tử ;

Trong đó: Kiểu phần tử là kiểu của mỗi phần tử trong mảng. Kiểu phần tử có thể là kiểu bất kỳ.

Chỉ số để truy nhập đến các phần tử của mảng. Kiểu chỉ số chỉ cho phép là các kiểu đơn giản sau đây: Kiểu kí tự ( CHAR), kiểu BOOLEAN, kiểu miền con ( khoảng con), kiểu liệt kê. Kiểu chỉ số không được là kiểu REAL hoặc INTEGER.

Số chỉ số là số chiều của mảng, mảng 1 chiều có 1 chỉ số, mảng 2 chiều có 2 chỉ số, ... , mảng n chiều có n chỉ số.

Kích thước tối đa của mảng phải được khai báo là một số xác định ( là hằng), chẳng hạn ta có thể khai báo là 5 hoặc 10 hay 100,... chứ không được khai báo là một biến như n,m,...

## Ví dụ 1

```
var B: array[1..5] of char ;
ở ví dụ 1 mảng B có kích thước tối đa là 5.
```

Ví dụ 2

```
TYPE AB = ARRAY [1 .. 5] OF INTEGER ;
 COLOR = (Red, Blue, Green, While, Black);
VAR X,Y,Z : AB;
 MAO, MKHAN : COLOR;
```

Ví dụ 3

```
VAR DSHodem,DSTen : Array [1.. 200] of string [20] ;
 DSHeso, DSLuong, DSPhucap,DSTong : array [1.. 200] of real;
 So_lap : array ['a' .. 'z'] of integer;
```

### 1.3- Truy nhập mảng

Có thể truy nhập vào bất kỳ phần tử nào trong mảng. Để truy nhập vào một phần tử trong mảng ta viết theo qui cách sau:

Tên\_biến\_mảng [ chỉ\_số1, chỉ\_số2, ..., chỉ\_số<sub>n</sub> ]

Các phần tử của mảng được coi như một biến, có thể tham gia vào các thủ tục vào/ra, các biểu thức, lời gọi hàm.

Ví dụ: Var a:array[1..20] of integer;

```
A[1]:=1;
readln(a[2]);
a[3]:=a[1]+a[2];
Writeln(a[3]);
```

### 1.4- Mảng 1 chiều

\* **Khai báo mảng một chiều:** Dùng cụm từ sau

ARRAY [kiểu\_chi\_số] OF kiểu\_phần\_tử;

- Dùng khai báo kiểu:

TYPE Tên\_kiểu\_mảng = ARRAY [ kiểu\_chi\_số ] OF kiểu\_phần\_tử ;

VAR Tên\_biến\_mảng: Tên\_kiểu\_mảng ;

- Dùng khai báo biến :

VAR Tên\_biến\_mảng : ARRAY [ kiểu\_chi\_số ] OF kiểu\_phần\_tử ;

Mảng một chiều chỉ có một chỉ số.

\* **Cách dùng :** Mảng 1 chiều thường được dùng cho dữ liệu ở dạng danh sách tuyến tính, ví dụ như dãy số, dãy xâu kí tự, ...

**Ví dụ 1:** Một dãy số nguyên  $a_1, a_2, \dots, a_n$  ta khai báo như sau

```
VAR a: ARRAY [1. . 100] OF integer ;
```

Trong khai báo này n có giá trị tối đa là 100.

**Ví dụ 2:** Một danh sách có n tên học sinh ta khai báo như sau:

```
VAR Ten: ARRAY [1.. 200] OF String [25] ;
```

Trong khai báo này n có giá trị tối đa là 200, mỗi tên có tối đa là 25 kí tự.

**Ví dụ 3:** Danh sách số lần xuất hiện ( tần số ) của các chữ cái viết hoa trong một văn bản ta khai báo như sau:

```
VAR Tan_so : ARRAY ['A' .. 'Z'] OF integer ;
```



Trong khai báo này mảng có kích thước tối đa là 26 ( 26 chữ cái hoa trong bảng chữ cái tiếng Anh).

\* Có thể truy nhập vào bất kỳ phần tử nào trong mảng. Để truy nhập vào một phần tử trong mảng một chiều ta viết theo qui cách sau:

Tên\_biến\_mảng [ chỉ-số ]

Ví dụ : Chỉ ra phần tử thứ 5 trong ví dụ 1 ta viết a[5]

Chỉ ra tên thứ 10 trong danh sách tên ta viết Ten[10]

Chỉ ra tần số của chữ 'B' ta viết Tan\_so[ 'B' ]

### 1.5. Các chương trình dùng mảng một chiều

**Bài toán 1:** Cho một dãy n số nguyên viết chương trình nhập dữ liệu vào, tính và in ra trung bình cộng, phần tử lớn nhất, phần tử nhỏ nhất của dãy số đó.

-Tur tưởng của thuật toán tìm phần tử max, phần tử min : Trước tiên gán phần tử đầu tiên là a[1] cho cả max và min, như vậy vị trí max, vị trí min đều là 1. Sau đó duyệt lần lượt các phần tử từ phần tử đầu cho tới phần tử cuối, nếu phần tử đang xét lớn hơn max thì gán giá trị phần tử đó cho max, vị trí của nó cho vị trí max, nếu phần tử đang xét nhỏ hơn min thì gán giá trị phần tử đó cho min, vị trí của nó cho vị trí min.

Chương trình

```
Program Trung_binh_max_min;
Uses crt;
Var a:Array [1..100] of integer ;
 i,n,max,min,vtmax,vtmin : integer ; tb : real ;
Begin
 Clrscr;
 { Nhập du lieu }
 Write(' Nhập n: '); readln(n);
 for i:=1 to n do
 begin
 Write(' a[', i, ']=');
 readln(a[i]);
 end;
 { Tinh toán }
 tb:=0; max:= a[1]; min:=a[1]; vtmax:=1; vtmin:=1;
 for i:=1 to n do
 begin
 tb:=tb + a[i];
 if max <a[i] then begin max:=a[i]; vtmax:=i; end;
 if min >a[i] then begin min :=a[i]; vtmin:=i; end;
 end;
 { in ket qua }
 writeln(' Trung binh = ', tb/n :8:2) ;
 writeln(' max= ', max, ' tai vi tri : ', vtmax);
 writeln(' min= ', min, ' tai vi tri : ', vtmin);
 readln;
end.
```

**Bài toán 2:** Cho dãy n số thực  $a_1, a_2, \dots, a_n$  sắp xếp dãy theo thứ tự tăng dần.

Trong bài toán này ta dùng thuật toán tìm phần tử nhỏ nhất của dãy chưa sắp và đưa về đúng ở vị trí đầu tiên của dãy đó. Dãy có n phần tử thì ta phải thực hiện n-1 lần tìm phần tử lớn nhất.

Tư tưởng của thuật toán như sau:

-Ta so sánh phần tử đầu tiên của dãy chưa sắp lần lượt với các phần tử đứng sau nó , nếu có phần tử nào nhỏ hơn nó thì ta sẽ đổi giá trị cho nhau. Để đổi giá trị 2 phần tử cho nhau ta dùng một phần tử trung gian.

- Lặp lại bước trên cho đến khi dãy chưa sắp không chỉ còn một phần tử. Như vậy dãy có n phần tử thì ta lặp lại n-1 lần.

### Chương trình

```
PROGRAM SAP_DAY_TANG;
USES CRT;
VAR I,J,N: INTEGER; A:ARRAY[1..100] OF REAL; X:REAL ;
BEGIN
 CLRSCR;
 (* NHAP SO LIEU *)
 WRITE(' HAY NHAP SO PHAN TU N ? ');READLN(N);
 FOR I:=1 TO N DO
 BEGIN
 WRITE('A[',I,']= ');
 READLN(A[I]);
 END;
 (* SAP XEP *)
 FOR I:=1 TO N-1 DO
 FOR J:=I+1 TO N DO IF A[I]>A[J] THEN BEGIN
 X:=A[I];
 A[I]:=A[J];
 A[J]:=X;
 END;
 (* IN KET QUA *)
 WRITELN(' DAY DA SAP THEO THU TU TANG');
 FOR I:=1 TO N DO WRITELN(A[I]:8:2);
 READLN;
END.
```

**Bài toán 3:** Nhập vào một xâu kí tự. Hãy in ra tần số xuất hiện của các chữ cái, không biệt chữ hoa và chữ thường. In ra chữ cái xuất hiện nhiều nhất.

Chương trình

```
program Tan_so_chu_cai ;
uses crt;
var s:string; ts: array ['A'..'Z'] of integer ; n,j,m: integer ; i,vt: char ;
Begin
 clrscr ;
 { nhap xau ki tu }
 Write('Nhap xau ki tu : '); readln(s);
 n:=length(s);
 { Tinh tan so }
 for i:='A' to 'Z' do ts[i] :=0 ;
```

```

for j:= 1 to n do
 for i:='A' to 'Z' do
 if upcase(s[j])=i then ts[i]:=ts[i] + 1;
 for i:='A' to 'Z' do writeln(i,' co tan so = ',ts[i]);
{ Tin tan so max }
m:=ts['A']; vt:='A';
for i:='B' to 'Z' do
 if m<ts[j] then begin
 m:=ts[j];
 vt:=i ;
 end ;
Writeln('ki tu ',vt, 'co tan so lon nhat la : ',m);
readln;
end.

```

## 1.6- Mảng nhiều chiều

Trong phần trình bày mảng 2 chiều, mảng nhiều hơn 2 chiều được suy diễn một cách tương tự.

\* **Khai báo mảng 2 chiều:** Dùng cụm từ sau

ARRAY [kiểu\_chi\_số1, kiểu\_chi\_số2] OF kiểu\_phần\_tử;

- Dùng khai báo kiểu:

TYPE

Tên\_kiểu\_mảng = array [ kiểu\_chi\_số1, kiểu\_chi\_số2 ] OF kiểu\_phần\_tử ;

VAR Tên\_biến\_mảng: Tên\_kiểu\_mảng ;

- Dùng khai báo biến :

VAR

Tên\_biến\_mảng : ARRAY [ kiểu\_chi\_số1,kiểu\_chi\_số2 ] OF kiểu\_phần\_tử ;

Mảng 2 chiều có 2 chỉ số.

\* **Cách dùng :** Mảng 2 chiều thường được dùng cho dữ liệu ở dạng bảng hay ma trận, ví dụ như ma trận số có m dòng ,n cột, ... Mảng hai chiều có 2 chỉ số, chỉ số 1 chỉ dòng, chỉ số 2 chỉ cột.

Ví dụ1: Một ma trận số nguyên a có 2 dòng, 3 cột được khai báo như sau

VAR A: ARRAY [ 1.. 2, 1..3 ] OF integer ;

Trong bộ nhớ máy lưu trữ các phần tử của mảng A kế tiếp nhau theo thứ tự sau:

A[1,1], A[1,2], A[2,1], A[2,2], A[3,1], A[3,2]

Ví dụ 2: Một bảng có 8 dòng, 8 cột, các ô của bảng chứa các chữ cái sẽ được khai báo như sau:

VAR Bang: ARRAY [ 1.. 8, 1..8 ] OF char ;

\* Có thể truy nhập vào bất kỳ phần tử nào trong mảng. Để truy nhập vào một phần tử trong mảng hai chiều ta viết theo qui cách sau:

Tên\_biến\_mảng [ chỉ\_số1, chỉ\_số2 ]

Cách viết trên để chỉ ra phần tử ở dòng có giá trị bằng chỉ số 1 và ở cột có giá trị bằng chỉ số 2.

Ví dụ: Var A:array[1..10,1..5] of integer;

A[1,1]:= 1;

A[1,2]:=3\*A[1,1];

Readln(A[2,1]);

```
Writeln(A[1,2]);
```

### 1.7. Các chương trình dùng mảng 2 chiều

**Bài toán 1:** Viết chương trình nhập một ma trận m dòng, n cột và tìm giá trị lớn nhất, giá trị nhỏ nhất, tính tổng của các phần tử, đếm số phần tử âm.

*Chương trình*

```
PROGRAM TINH_MA_TRAN;
(*TIM MAX, MIN, TINH TONG, DEM SO PHAN TU AM *)
USES CRT;
VAR I,J,M,N,MAX,MIN,T,SOPTAM:INTEGER;
 A:ARRAY[1..30,1..20] OF INTEGER;
BEGIN
 CLRSCR;T:=0;SOPTAM:=0;
 (* NHAP SO LIEU *)
 WRITE('NHAP SO DONG M, SO COT N ');READLN(M,N);
 FOR I:=1 TO M DO
 FOR J:=1 TO N DO BEGIN
 WRITE('A['I','J,']=? ');
 READLN(A[I,J]);
 END;
 (* TINH TOAN *)
 MAX:=A[1,1];
 MIN:=MAX;
 FOR I:=1 TO M DO
 FOR J:=1 TO N DO
 BEGIN
 T:=T+A[I,J];
 IF A[I,J]<0 THEN SOPTAM:=SOPTAM+1;
 IF MAX<A[I,J] THEN MAX:=A[I,J] ELSE IF MIN>A[I,J] THEN MIN:=A[I,J];
 END;
 (* IN KET QUA *)
 WRITELN;
 WRITELN('TONG T= ',T);
 WRITELN('SO PHAN TU AM LA ',SOPTAM);
 WRITELN('GIA TRI LON NHAT = ',MAX);
 WRITELN('GIA TRI NHO NHAT = ',MIN);
 READLN;
END.
```

**Bài toán 2:** Nhân ma trận a có m dòng, n cột với ma trận b có n dòng, l cột.

Kết quả là ma trận c có m dòng, l cột. Các phần tử của ma trận c được tính theo công thức sau:

$$C[i,j] = \sum_{k=1}^n a[i,k] * b[k,j] \quad \text{với } i: 1 \rightarrow m; j: 1 \rightarrow l$$

## Chương trình

```
(* nhan hai ma tram *)
program nhan_ma_tran;
uses crt;
type mang=array[1..20,1..30] of integer;
var i,j,k,m,l,n:integer; a,b,c: mang;
begin
 clrscr;
 (* nhap ma tran a *)
 write('nhap so dong, so cot m,n ? ');readln(m,n);
 for i:=1 to m do
 for j:=1 to n do
 begin
 write('a[', i, ', ', j, ']= ');
 readln(a[i,j]);
 end;
 (* nhap ma tran b *)
 write('nhap so cot l ? ');readln(l);
 for i:=1 to n do
 for j:=1 to l do
 begin
 write('b(',i,',',j,')= ');
 readln(b[i,j]);
 end;
 (* Tinh toan *)
 for i:=1 to m do
 for j:=1 to l do
 begin
 c[i,j]:=0;
 for k:=1 to n do c[i,j]:=c[i,j]+a[i,k]*b[k,j];
 end;
 (* In ket qua *)
 writeln(' Ma tran tich ');
 for i:=1 to m do
 begin
 for j:=1 to l do write(' ',c[i,j]:4);
 writeln;
 end;
 readln;
end.
```

## 2. Kiểu cấu trúc

### 2.1. Khai báo kiểu cấu trúc

*a. Định nghĩa* : Dữ liệu kiểu cấu trúc là kiểu dữ liệu có cấu trúc, dùng để xử lý các cấu trúc

Dữ liệu kiểu xâu được khai báo bằng từ khoá `STRING` , độ dài tối đa của một xâu kí tự là 255 kí tự. Có thể khai báo độ dài tối đa của một xâu kí tự trong dấu ngoặc vuông sau từ khoá `STRING` như sau: `STRING[...]`

Ví dụ: Tên người, quê quán, trình độ văn hóa là các dữ liệu kiểu xâu kí tự.

### ***b. Khai xâu kí tự***

Để khai báo xâu kí tự dùng từ khóa `String` nếu như khai báo độ dài xâu tối đa là 255 kí tự, dùng từ khóa `String [n]` nếu như khai báo độ dài xâu tối đa là `n` kí tự.

- Khai báo kiểu xâu:  
`Type Tên_kiểu_xâu = String[n];`
- Khai báo biến xâu  
`Var Tên_biến: String[n];`

Trong đó `n` là số kí tự tối đa của xâu, nếu không có phần `[n]` thì số kí tự tối đa của xâu mặc định là 255.

Ví dụ `Var Hoten: string[30]; Ngaysinh: string[10]; Quequan: string;`

Hàng xâu kí tự trong Pascal được viết trong ' '

```
Hoten:='Le Thu Ha';
Ngaysinh:='20-10-1960';
Quequan:='Gia lam - Ha noi ';
```

### ***c. Truy nhập xâu kí tự***

Mỗi kí tự trong xâu được chỉ ra bằng 1 số thứ tự được gọi là chỉ số viết trong `[ ]`, kí tự đầu tiên có chỉ số là 1. Có thể truy nhập tới từng kí tự trong xâu theo cách viết sau: `Tên_biến[chỉ_số]`

Với ví dụ trên `Hoten[2]` là 'e', `Ngaysinh[8]` là 9.

Xâu kí tự được lưu trữ như sau: Byte đầu tiên chứa kí tự là độ dài thực của xâu, các byte tiếp theo là các kí tự của xâu.

Với ví dụ trên `Hoten:='Le Thu Ha';` được lưu trữ như sau:

|   |   |   |   |   |   |   |   |   |   |     |    |
|---|---|---|---|---|---|---|---|---|---|-----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ... | 30 |
|   | L | e |   | T | h | u |   | H | a |     |    |

Kí tự đầu chứa độ dài thực của xâu là kí tự 9.

Độ dài của xâu = `ORD( Hoten[0])`

***d. Hàm chuẩn Length(St) :*** Cho ta độ dài của xâu kí tự. Một xâu kí tự không chứa kí tự nào là rỗng, khi đó `Length(St)=0`, xâu kí tự rỗng được kí hiệu "".

## **2.2. Các thao tác trên xâu kí tự**

### ***a. Phép cộng xâu***

- Kí hiệu `+` là ghép nối các xâu kí tự
- Ví dụ `Que:= 'Gia lam ' + 'Ha noi';`  
Cho kết quả `Que= 'Gia lam Ha noi'`

### ***b. Phép So sánh hai xâu kí tự***

Khi so sánh 2 xâu kí tự, so sánh mã ASCII của từng cặp kí tự tương ứng từ 2 xâu theo trình tự từ trái sang phải, sẽ xuất hiện 1 trong các trường hợp sau:

- Nếu gặp một cặp có mã khác nhau thì xâu chứa kí tự có mã nhỏ hơn là xâu nhỏ hơn.
- Nếu tất cả các cặp kí tự đều có mã giống nhau thì 2 xâu bằng nhau.

- Nếu 2 xâu có độ dài khác nhau song các cặp ký tự đều có mã giống nhau đến độ dài của xâu ngắn thì xâu ngắn sẽ là xâu nhỏ hơn.

Kết quả của phép so sánh là giá trị logic True hoặc False.

'hang' < 'hun' cho giá trị True.

'Thoa' = 'Thoa' cho giá trị True.

'nhu' > 'nhung' cho giá trị False.

- Đọc xâu ký tự từ bàn phím: `Readln(St)`; Độ dài thực của xâu St là số ký tự gõ vào từ bàn phím. Nếu không gõ ký tự nào mà gõ Enter luôn thì xâu St rỗng.
- Viết xâu ký tự ra màn hình: `Write(St)` và `Writeln(St)`

### 2.3. Các thủ tục và các hàm chuẩn xử lý xâu ký tự

Giả thiết St là xâu ký tự có độ dài chuẩn là 255 ký tự.

#### a. Hàm *Length(St)*

Hàm này cho độ dài của xâu ký tự.

St:='Tin học' ; `Length(St)` có giá trị là 7.

#### b. Thủ tục *Delete(St,m,n)*

Thủ tục này xoá đi n ký tự, bắt đầu từ vị trí m trong xâu St.

St:='Cong trinh khoa hoc';

`Delete(St,6,5)`;

Kết quả St sẽ còn 'Cong khoa hoc' .

Nếu  $m+n > \text{Length}(St)$  thì chỉ xoá đi những ký tự nằm trong `Length(St)`.

#### c. Thủ tục *Insert (s1,St,m)*

Thủ tục này chèn xâu s1 vào xâu St tại vị trí m.

St:='Hoc tot';

s1:='Toan ';

`Insert(s1,St,5)`;

Kết quả St sẽ là 'Hoc Toan tot'.

Nếu  $\text{Length}(s1)+\text{Length}(St) >$  độ dài cực đại cho phép của xâu thì chỉ những ký tự nào nằm trong khoảng độ dài cực đại cho phép mới được giữ lại.

#### d. Hàm *Copy(St,m,n)*

Hàm này sao chép n ký tự của xâu St từ vị trí m.

St:='Lao dong';

s1:=`Copy(St,5,2)`; Kết quả sẽ cho s1='do'.

Nếu  $m > \text{Length}(St)$  thì Copy sẽ cho 1 xâu rỗng.

Nếu  $m + n > \text{Length}(St)$  thì Copy chỉ nhận các ký tự nằm trong xâu St.

#### e. Hàm *Concat(St1,St2,...,Stn)*

Hàm này ghép nối tất cả các xâu ký tự St1,St2,...,Stn thành một xâu ký tự theo thứ tự đã viết.

Tổng số chiều dài của các xâu ký tự không được lớn hơn 255.

St1:='Lao dong ';

St2:='Hoc tap ';

St3:='Vui chơi';

St:=`Concat(St1,St1,St3)`; Kết quả St là 'Lao dong Hoc tap Vui chơi'.

#### f. Hàm *Pos(s1,St)*

Hàm này cho vị trí đầu tiên của xâu s1 trong xâu St. Nếu không tìm thấy thì hàm cho giá trị là 0.

St:='12345abc';

```

s1:='345';
s2:='35ab';
Pos(s1,St) cho giá trị là 3.
Pos(s2,St) cho giá trị là 0.

```

#### g. Thủ tục *Str(x,St)*

Thủ tục này biến đổi giá trị số nguyên hoặc thực x thành một chuỗi ký tự St biểu diễn số đó. Cách biểu diễn của chuỗi St sẽ được qui định do qui cách của x.

```

I:=1234;
Str(I:5,St) sẽ cho St=' 1234'
x:=1234.56789111;
Str(x:9:4,St) sẽ cho St='1234.5678'

```

#### h. Thủ tục *Var(St,x,m)*

Thủ tục này biến đổi chuỗi ký tự St ( biểu diễn số nguyên hoặc thực ) thành một số nguyên hoặc thực chứa trong x. Biến m là biến nguyên chứa mã lỗi, nếu biến đổi đúng thì m=0, nếu biến đổi sai thì m cho giá trị là vị trí của ký tự sai.

Ví dụ 1: St:='1234.567'; x là biến thực, m là biến nguyên.  
Var(St,x,m); cho ta x=1234.567 và m=0.

Ví dụ 2: St:='1234'; x là biến nguyên, m là biến nguyên.  
Var(St,x,m); cho ta x=1234 và m=0

Ví dụ 3: St:='123ab'; x là biến thực, m là biến nguyên.  
Var(St,x,m); kết quả sai và m≠0.

## 2.4. Các chương trình

**Bài toán 1:** Nhập 1 chuỗi ký tự từ bàn phím, kiểm tra xem chuỗi có đối xứng không.

Chuỗi ký tự S có độ dài n, là chuỗi đối xứng khi có tất cả các cặp ký tự  $S[i] = S[n-i+1]$  với i từ 1 đến  $(n \text{ Div } 2)$ . Chẳng hạn chuỗi '1234321' và chuỗi 'abccba' là các chuỗi đối xứng, chuỗi '12343321' là chuỗi không đối xứng.

Chương trình:

```

Program KT_xau_doi_xung;
Uses crt;
Var s: string; i,n : integer; t: boolean;
Begin
 clrscr;
 Writeln('Nhập xâu kí tự'); readln(s);
 n:=length(s);
 t:=true;
 for i:=1 to n div 2 do
 if s[i] <> s[n-i+1] then t:=false;
 if t then writeln(' Xâu kí tự doi xung')
 else writeln(' Xâu khong doi xung');
 readln;
end.

```

**Bài toán 2:** Nhập vào 2 chuỗi ký tự có độ dài như nhau. Xây dựng chuỗi mới chứa các ký tự xen kẽ của 2 chuỗi nhập vào theo thứ tự từ trái sang phải,. Chẳng hạn s1='123' s2='abc' thì các chuỗi mới xây dựng là s3='1a2b3c', s4='a1b2c3'



Chương trình:

```
Program Ghep_xen_ke_xau_ki_tu;
Uses crt;
Var s1,s2,s3,s4: string; i,n : integer;
Begin
 clrscr;
 Writeln('Nhap xau ki tu mot s1: '); readln(s1);
 Writeln('Nhap xau ki tu hai s2: '); readln(s2);
 s3:='';
 s4:='';
 n:=length(s1);
 for i:=1 to n do
 begin
 s3:=s3+s1[i]+s2[i];
 s4:=s4+s2[i]+s1[i];
 end;
 Writeln(' Cac xau ki tu moi tao');
 Writeln(s3);
 Writeln(s4);
 readln;
end.
```

**Bài toán 3:** Chuẩn hoá văn bản. Nội dung chuẩn hoá văn bản như sau: Nhập vào các dòng văn bản từ bàn phím, chuẩn hoá theo các qui định :

- Các từ chỉ cách nhau 1 khoảng cách.
- Sau dấu chấm phải viết hoa.
- Trước các dấu . , ; : không có khoảng cách.
- Sau các dấu . , ; : có 1 khoảng cách.

## Chương trình

```
{ xuly tep van ban ve cac dau .,;: }
PROGRAM XU_LY_VAN_BAN;
USES CRT;
VAR S,T:STRING;TIEP:CHAR;
PROCEDURE SUA(VAR P:STRING);
VAR I:INTEGER;
BEGIN
 { xoa khoang cach thua }
 I:=1;
 WHILE I<LENGTH(P)-1 DO
 IF (P[I]=' ') AND (P[I+1]≠' ') THEN DELETE(P,I+1,1) ELSE I:=I+1;
 { xoa khoang cach truooc cac dau .,;: }
 I:=1;
 WHILE I<LENGTH(P) DO
 IF (P[I]=' ') AND ((P[I+1]='.') OR (P[I+1]=';') OR (P[I+1]=':') OR (P[I+1]='.'))
 THEN DELETE(P,I,1) ELSE I:=I+1;
 { chen dau khoang trong sau cac dau neu thieu }
 I:=1;
 WHILE I< LENGTH(P)-1 DO
 IF ((P[I]='.') OR (P[I]=';') OR (P[I]=':') OR (P[I]='.')) AND (P[I+1]<>' ')
```

```

 THEN BEGIN INSERT(' ',P,I+1);
 I:=I+1;
 END
 ELSE I:=I+1;
 { Viet hoa sau dau . }
 I:=1;
WHILE I<LENGTH(P)-2 DO
 IF P[I]='.' THEN BEGIN P[I+2]:=UPCASE(P[I+2]);
 I:=I+1;
 END
 ELSE I:=I+1;

END;
{ Than chuong trinh chinh }
BEGIN
 CLRSCR;
 TIEP:='C';
 WHILE UPCASE(TIEP)='C' DO
 BEGIN
 WRITELN('NHAP XAU KI TU ');
 READLN(S);
 SUA(S);
 WRITELN(S);
 WRITE('CO TIEP TUC KHONG(C/K) ');
 READLN(TIEP);
 END;
 READLN;
END.

```

### 3. Kiểu tập

#### 3.1. Khái niệm tập

Tập là một bộ các đối tượng vô hướng và cùng kiểu.  
 Mỗi đối tượng gọi là một phần tử của Tập. Tập có tối đa là 256 phần tử. Nếu phần tử là kiểu số thì chỉ cho phép là các số nguyên có giá trị từ 0..255.

Khái niệm Tập gắn liền với khái niệm Tập hợp trong toán học.

**Ví dụ:** Tập các chữ cái hoa, tập này có 26 phần tử.

Tập các số nguyên dương có 2 chữ số, tập này có 90 phần tử.

#### 3.2. Khai báo tập

Khai báo Tập dùng cụm từ : SET of Kiểu\_phần\_tử;

Kiểu phần tử phải là một kiểu vô hướng.

- **Dùng khai báo kiểu**

Type Tên\_kiểu=Set of kiểu\_phần\_tử;

Var Tên\_biến: Tên\_kiểu;

Ví dụ: Type t=set of 1..200;

```

Chu_hoa: Set of 'A'..'Z';
Var tuoi : t;
Chu_in: chu_hoa;
- Khai báo trực tiếp trong khai báo biến
Var Tên_biến: Set of Kiểu_phần_tử;
Ví dụ: Type mau=(do,xanh,vang,tim,nau);
 hinh=(tam_giac,tron,chu_nhat);
 Var mau_ao: Set of mau;
 hinh_ve: set of hinh;
 chu_so: 1..9;

```

### 3.3- Xây dựng một tập

Xây dựng một tập bằng cách liệt kê các phần tử của tập, chúng được cách nhau bởi dấu phẩy và được viết trong dấu móc vuông ( [ ] ).

Tập rỗng là tập không chứa một phần tử nào, được viết là []  
[1..99] ; tập này có 100 phần tử .

[ 2,4,6,8..12]; tập có các phần tử là 2,4,6,8,9,10,11,12.

['a'..'d','m','n'] ; tập này có các phần tử 'a', 'b', 'c', 'd', 'm', 'n'.

[xanh,do,tim,vang];

Các phần tử của tập cũng có thể cho bằng biến hoặc biểu thức.

[2\*i+j, i\*j-2] ; nếu i=2 và j=3 thì tập có các phần tử là 7, 4.

Ta được dùng kiểu khoảng con để chỉ ra các phần tử của tập.

### 3.4- Các phép toán trên tập

#### a. Phép gán

Gán một tập cho biến kiểu tập.

Ví dụ

```
var chu: Set of 'A'..'Z';
```

```
tuoi: Set of 1..200;
```

Khi đó có thể thực hiện các phép gán sau:

```
chu:=['A', 'C', 'M'..'P'];
```

```
tuoi:=[50..80, 90,100];
```

Tập rỗng có thể đem gán cho mọi biến kiểu tập khác nhau.

```
Chu:=[];
```

```
tuoi:=[];
```

Không thể gán các tập kiểu cơ bản không tương thích.

Chẳng hạn nếu gán chu:=[1..10] là sai.

#### b. Phép hợp

Phép hợp được kí hiệu bằng dấu +

Hợp của 2 tập là một tập có các phần tử thuộc hai tập.

Ví dụ A:=[1,3,5..10];

B:=[2,4,6,8];

C:=[1,5,9];

D:=A+B; tập D sẽ là [1..10]

E:=B+C; tập E sẽ là [1,2,4,5,6,8,9]

#### c. Phép giao

Phép giao được kí hiệu bằng dấu \*

Giao của 2 tập là 1 tập có các phần tử là các phần tử chung của cả 2 tập.

$M:=A*B$ ; tập M sẽ là [6,8]

$N:=B*C$ ; tập N=[]

#### d. Phép hiệu

Phép hiệu được kí hiệu bằng dấu -.

Hiệu của 2 tập là 1 tập chứa các phần tử thuộc tập thứ nhất nhưng không thuộc tập thứ 2.

$G:=A - B$  ; tập G sẽ là [1,3,5,7,9,10]

$H:=C - A$  ; tập H sẽ là tập rỗng.

#### e. Phép thuộc về

Phép thuộc về kí hiệu là IN

Phép thuộc về cho biết 1 phần tử hay 1 tập có thuộc về 1 tập khác không ?, kết quả của phép thuộc về là giá trị kiểu boolean (True hoặc False).

1 in A; kết quả là true.

B in A; kết quả là False.

C in A; kết quả là True;

Phép so sánh =, <>, <=, >=

Hai tập đem so sánh phải cùng kiểu.

Kết quả của phép so sánh là giá trị kiểu boolean.

- Hai tập bằng nhau (=) nếu chúng có các phần tử như nhau từng đôi một.

Ví dụ:  $x:=['a' .. 'd']$ ;

$y:=['a' .. 'd', 'e', 'f']$ ;

$z:=['a', 'b', 'c', 'd']$

$x=y$ ; kết quả là False.

$x=z$ ; kết quả là True.

- Hai tập là khác nhau (<>) nếu chúng có ít nhất 1 cặp phần tử khác nhau.

$x<> y$  ; kết quả là True;

$x<> z$  ; kết quả là False.

- Tập thứ nhất <= tập thứ 2 nếu tất cả các phần tử của tập thứ nhất đều thuộc tập thứ 2.

$x <= y$ ; kết quả là true.

$y <= x$ ; kết quả là False.

- Tập thứ nhất >= tập thứ 2 nếu mọi phần tử của tập thứ 2 đều thuộc tập thứ 1.

$y >= x$ ; kết quả cho True.

$z >= y$ ; kết quả là False.

### 3.5- Các chương trình về tập

**Bài toán :** Tạo một tập chứa các số có tối đa là 2 chữ số nguyên dương < 256 chia hết cho 7, in tập đó ra. Sau đó nhập 1 số bất kỳ kiểm tra xem nó có thuộc vào tập đó không.

*Chương trình*

```
Program thao_tac_tap;
uses crt;
var a: set of 1..99; i,n: byte; lap: char;
Begin
 clrscr;
```

```

{ Tao tap cac so chia het cho 7 }
a:=[];
for i:=7 to 255 do
 if (i mod 7) = 0 then a:=a+[i];
{ In tap }
Writeln(' Tap cac so chia het cho 7');
for i:=7 to 255 do
 if i in a then writeln(i);
{ Kiem tra so nhap thuoc tap chi het cho 7 }
Repeat
 write(' Nhap so bat ky ');
 Readln(n);
if i in a then writeln('Chia het cho 7') else writeln('Khong chia het cho 7');
 Write(' Co tiep tục không (C/K) ? '); readln(lap);
 Until upcase(lap)= 'K';
readln;
end.

```

## Bài tập chương 4

Hãy viết chương trình cho các bài toán sau:

1. Cho dãy số sau:  $a_1, a_2, \dots, a_n$ . Viết chương trình tìm phần tử lớn nhất, phần tử nhỏ nhất của dãy số đó.
2. Cho dãy số sau:  $a_1, a_2, \dots, a_n$ . Viết chương trình sắp xếp dãy theo thứ tự tăng dần.
3. Cho dãy số sau:  $a_1, a_2, \dots, a_n$ . Viết chương trình đếm số phần tử dương và xoá đi phần tử thứ  $m$  trong dãy ( $m \leq n$ ).
4. Cho dãy số sau:  $a_1, a_2, \dots, a_n$ . Viết chương trình tìm các phần tử có giá trị là  $x$  nhập vào từ bàn phím.
5. Cho dãy số sau:  $a_1, a_2, \dots, a_n$ . Viết chương trình thêm phần tử có giá trị là  $x$ , vào vị trí  $m$  trong dãy. Sau đó tính tổng các phần tử của dãy mới.
6. Cho ma trận có  $m$  dòng và  $n$  cột, các phần tử là nguyên. Tìm phần tử nhỏ nhất của ma trận.
7. Cho ma trận có  $m$  dòng và  $n$  cột, các phần tử là nguyên. Tính tổng và trung bình cộng các phần tử của ma trận.
8. Viết chương trình nhập vào một chuỗi ký tự. Hãy xây dựng chuỗi chứa các ký tự đảo của chuỗi đó, đếm xem có bao nhiêu ký tự 'a' hoặc 'A' trong chuỗi.
9. Viết chương trình nhập vào một chuỗi ký tự. Hãy kiểm tra xem chuỗi đó có đối xứng không. In ký tự đầu và ký tự cuối của chuỗi đó.
10. Viết chương trình nhập vào hai chuỗi ký tự có độ dài bằng nhau. Hãy xây dựng chuỗi chứa các ký tự xen kẽ của hai chuỗi đó, theo thứ tự một ký tự của chuỗi 1 rồi đến 1 ký tự của chuỗi 2.

# CHƯƠNG V

## CHƯƠNG TRÌNH CON: HÀM VÀ THỦ TỤC

### 1 - Cấu trúc của hàm và thủ tục

#### 1.1- Chương trình con

Khi lập trình gặp đoạn chương trình cần dùng nhiều lần, để tránh viết lặp lại thì đoạn chương trình đó được tổ chức thành chương trình con và mỗi khi cần gọi tới chương trình con đó.

Mặt khác khi xây dựng chương trình cho các bài toán lớn, phức tạp, để dễ cho việc thiết kế chương trình, hiệu chỉnh chương trình, gỡ rối khi chạy chương trình, bài toán lớn được phân thành các phần nhỏ, mỗi phần xây dựng thành các chương trình con. Chương trình chính sẽ gọi tới các chương trình con.

Trong Pascal có 2 loại chương trình con đó là hàm (Function) và thủ tục (Procedure).

#### *a - Cấu trúc của hàm (Function) và lời gọi hàm*

- **Hàm có cấu trúc đầy đủ như sau:**

```
Function Tên_hàm(Tham_số1 : kiểu; Tham_số2: kiểu; Var tham_số3: kiểu;. . .): kiểu;
 Label { Khai báo các nhãn }
 Const { Khai báo các hằng }
 Type { Định nghĩa các kiểu dữ liệu của người sử dụng }
 Var { Khai báo các biến cục bộ }
 . . .
 Begin
 . . . { Thân chương trình con }
 Tên_hàm:= Giá_trị ;
 End;
```

Các phần nếu có thì theo đúng thứ tự đã nêu.

Kiểu của tham số là các kiểu cơ bản, kiểu có cấu trúc như kiểu xâu kí tự và kiểu mảng, nếu là kiểu mảng thì phải khai báo bằng định nghĩa kiểu ở phần định nghĩa khiêu ở đầu chương trình chính, không được khai báo trực tiếp.

Kiểu của hàm có thể là các kiểu cơ bản, kiểu xâu kí tự.

Các tham số khai báo trong hàm được gọi là tham số hình thức.

- **Lời gọi hàm**

Trong thân chương trình chính sử dụng hàm phải có lời gọi hàm.

Lời gọi hàm được viết như sau:

Tên\_hàm( danh sách các tham số thực sự)

Các tham số thực sự tương ứng cả về số lượng và cả về kiểu dữ liệu với các tham số hình thức khai báo trong hàm.

Lời gọi hàm được coi như 1 biến, có thể tham gia vào biểu thức, tham gia vào các thủ tục vào/ ra.

**Ví dụ1:** Chương trình có xây dựng Function

**Bài toán :** Tính diện tích của tam giác biết 3 cạnh a,b,c.

## Chương trình

```
Program Tinh_dien_tich;
uses crt;
var a,b,c : real;
Function DT(x,y,z : real) : real;
var s,p : real;
begin
 p:=(x+y+z)/2;
 s:=sqrt(p*(p-a)*(p-b)*(p-c));
 DT:=s;
end;
begin { than chương trình chính }
 clrscr;
 a:=2;b:=3;c:=2;
 writeln(' dien tich tam giac 1 = ', DT(a,b,c):8:2);
 write('Nhap 3 canh của tam giac a,b,c: ');
 readln(a,b,c);
 writeln(' dien tich tam giac 2 = ', DT(a,b,c):8:2);
 readln;
end.
```

- Hàm đặt ở vị trí sau khai báo biến trong chương trình chính và trước thân chương trình chính.

- Chỉ được gọi tới hàm sau khi đã khai báo hàm.

- Ở ví dụ 1 ta khai báo hàm có tên là DT có kiểu real, các tham số hình thức của hàm là: x,y,z. Trong thân chương trình có 2 lời gọi hàm, chúng đều nằm trong lệnh Writeln. Trong lời gọi hàm thứ nhất, 3 tham số thực sự là a=2, b=3, c=2. Trong lời gọi hàm thứ hai, 3 tham số thực sự là a,b,c có giá trị được nhập vào từ bàn phím.

## ***b - Cấu trúc của thủ tục (Procedure) và lời gọi thủ tục***

- **Thủ tục có cấu trúc đầy đủ như sau:**

```
Procedure Tên_thủ_tục(Tham_số1 : kiểu; Tham_số2: kiểu; Var tham_số3: kiểu;. . .);
Label { Khai báo các nhãn }
Const { Khai báo các hằng }
Type { Định nghĩa các kiểu dữ liệu của người sử dụng }
Var { Khai báo các biến cục bộ }
Begin
 . . . { thân chương trình con }
End;
```

Các phần nếu có thì theo đúng thứ tự đã nêu.

Kiểu của tham số là các kiểu cơ bản, kiểu có cấu trúc như kiểu xâu kí tự và kiểu mảng, nếu là kiểu mảng thì phải khai báo bằng định nghĩa kiểu ở phần định nghĩa kiểu ở đầu chương trình chính, không được khai báo trực tiếp.

Trong chương trình chính thủ tục đứng trước thân chương trình chính, sau khai báo biến.

Các tham số khai báo trong hàm được gọi là tham số hình thức.

- **Lời gọi thủ tục**

Trong thân chương trình chính sử dụng thủ tục phải có lời gọi thủ tục

Lời gọi thủ tục được viết như sau:

```
Tên_thủ_tục(danh sách các tham số thực sự);
```

Các tham số thực sự tương ứng cả về kiểu và số lượng với các tham số hình thức khai báo trong thủ tục.

Lời gọi thủ tục như 1 câu lệnh đứng độc lập.

**Ví dụ 2:** chương trình có xây dựng Procedure

**Bài toán:** Tính tổng và trung bình cộng của dãy số  $a_1, a_2, \dots, a_n$ .

Chương trình

```
Program Tinh_tong_tb;
uses crt;
type mang= array[1 .. 50] of real ;
var i,n: integer; a: mang; tg,tb: real;
Procedure tong(m: integer; x: mang; var s, p : real);
 Var j: integer ; t: real;
 begin
 t:=0;
 For j:=1 to m do t:=t + x[j] ;
 s:= t;
 p:=t/m;
 end;
begin { than chuong trinh chinh }
 clrscr;
 write(' nhap so phan tu cua day n ');
 readln(n);
 for i:= 1 to n do
 begin
 write(a[' ', i, ']='); readln(a[i]);
 end;
 tong(n,a,tg,tb);
 writeln(' tong= ', tg: 8: 2, 'trung binh = ', tb: 8: 2);
 readln;
end.
```

## 1.2 - Sự khác nhau giữa hàm và thủ tục

- Hàm cho 1 giá trị thông qua tên hàm. Tên hàm trong lời gọi hàm được coi như một biến có thể tham gia vào biểu thức, các thủ tục vào ra. Cuối thân hàm phải có lệnh gán giá trị cho tên hàm.

- Tên thủ tục không cho 1 giá trị nào cả.

- Các tham số viết sau tên hàm, sau tên thủ tục được gọi là tham số hình thức. Tham số hình thức có 2 loại: + Tham số không có từ khóa Var đi kèm trước gọi là tham trị.

+ Tham số có từ khóa Var đi kèm trước gọi là tham biến.

- Trong hàm thường chứa các tham trị, ít khi chứa các tham biến. Trong thủ tục thường có các tham biến.

- Các tham số trong lời gọi hàm, lời gọi thủ tục gọi là tham số thực sự. Các tham số thực sự phải tương ứng về số lượng và kiểu với các tham số hình thức. Các tham số thực sự tương ứng với các tham trị để chứa các dữ liệu vào. Các tham số thực sự tương ứng với các tham biến để chứa kết quả của thủ tục.

- Hàm lấy kết quả ở tên hàm, thủ tục lấy kết quả ở các tham số thực sự tương ứng với các tham biến.



- Như vậy nếu để lấy 1 kết quả ta có thể tổ chức hàm hoặc thủ tục. Nếu muốn lấy nhiều hơn 1 kết quả thì phải tổ chức chương trình con dạng thủ tục.

## 2 - Biến toàn cục, biến cục bộ và truyền dữ liệu

### 2.1 - Biến toàn cục

Biến toàn cục là biến khai báo ở đầu chương trình chính.

Biến toàn cục tồn tại suốt thời gian làm việc của chương trình .

Biến toàn cục có thể sử dụng cả trong chương trình chính và chương trình con.

Ví dụ 1 mục 1 (tính diện tích tam giác) có a,b,c là biến toàn cục.

Ví dụ 2 mục 1 ( tính tổng và trung bình) có i, n, a, tg, tb là biến toàn cục.

### 2.2 - Biến cục bộ

Biến cục bộ là các biến được khai báo ở đầu chương trình con.

Biến cục bộ được cấp phát bộ nhớ khi chương trình con được gọi tới và bị xoá khi ra khỏi chương trình con.

Biến cục bộ chỉ được dùng trong chương trình con.

Biến toàn cục và biến cục bộ có thể trùng tên nhau nhưng chương trình vẫn phân biệt 2 biến khác nhau.

Trong ví dụ 1 mục 1 (tính diện tích tam giác) có s, p là biến cục bộ.

Trong ví dụ 2 mục 2 ( tính tổng và trung bình) có j, t là biến cục bộ.

### 2.3 - Truyền dữ liệu

Khi gặp lời gọi chương trình con máy sẽ thực hiện các bước sau:

- Cấp phát bộ nhớ cho các tham số và cho các biến cục bộ trong chương trình con.
- Truyền giá trị của các tham số thực sự cho tham trị và truyền địa chỉ cho các tham biến.
- Thực hiện các lệnh trong thân chương trình con.
- Thực hiện xong chương trình con máy giải phóng các tham số và các biến cục bộ, rồi trở về chương trình chính.

## 3 - Tính đệ quy của chương trình con

Trong Function và Procedure có thể có lời gọi tới chính nó. Tính chất này được gọi là tính đệ quy.

Phương pháp đệ quy được áp dụng cho các bài toán thuật giải mang tính đệ quy.

Thuật giải đệ quy làm cho chương trình ngắn gọn, đẹp đẽ nhưng lại tốn thời gian tính toán và bộ nhớ. Có những bài toán chỉ có thể giải quyết được bằng xây dựng các chương trình con đệ quy.

Ví dụ1: Bài toán tính giai thừa

- Trường hợp suy biến:  $n! = 1$  khi  $n=0$

- Trường hợp tổng quát:  $n! = (n-1)! \cdot n$  khi  $n \geq 1$

**Có thể xây dựng hàm Giai thừa có tính chất đệ quy như sau:**

```
Function GiaiThua(n: longint): longint;
begin
```

```

if n=0 then Giaithua:= 1
else Giaithua:= Giaithua(n-1) * n ;
end;

```

- \* Muốn xây dựng được chương trình con đệ qui ta phải xác định được 2 trường hợp:
  - Trường hợp suy biến, đó là trường hợp đặc biệt mà xác định được giá trị của hàm.
  - Trường hợp tổng quát lần thứ n được tính dựa vào lần thứ (n-1).

Ví dụ 2: Tìm ước số chung lớn nhất của 2 số x và y có thể được định nghĩa như sau ( $x > y$ ):  
 USCLN(x,y) = x nếu  $y=0$  (đây là trường hợp suy biến)  
 USCLN(x,y) = USCLN(y, phần dư của x/y) nếu  $y \neq 0$  (đây là trường hợp tổng quát).  
 Trong hàm xây dựng với  $x > y$ , nếu  $y > x$  thì chương trình trao đổi giá trị giữ x và y.

**Hàm USCLN được viết như sau:**

```

Function USCLN(x,y: integer) : integer ;
Var t:Integer;
Begin
 If y>x then begin t:=x; x:=y; y:=t; end;
 if y=0 then USCLN:= x
 else USCLN := USCLN(y, x mod y) ;
end;

```

**4. Một số chương trình con của turbo pascal**

Pascal đã xây dựng sẵn một số hàm và thủ tục, Ta có thể gọi tới các hàm, thủ tục đó theo đúng quy cách của nó để sử dụng.

Ngoài các hàm và thủ tục đã xét ở các phần trên, trong phần này bổ sung thêm một số thủ tục sau đây:

**\* Procedure GotoXY(Xpos, YPos);**

Đưa con trỏ(cursor) của màn hình về vị trí có tọa độ Xpos và Ypos trên màn hình. Xpos, Ypos kiểu số nguyên.

**\* Procedure ClrScr;**

Xoá toàn bộ màn hình và đặt con trỏ vào vị trí phía trên, bên trái.

**\* Procedure ClrEof;**

Xoá toàn bộ các kí tự bên phải con trỏ màn hình. Sau khi xoá con trỏ vẫn ở tại chỗ.

**\* Procedure Deline;**

Xoá toàn bộ dòng màn hình chứa con trỏ, sau đó dồn các dòng ở dưới lên.

**\* Procedure InsLine;**

Xen một dòng trắng vào màn hình từ vị trí con trỏ.

**\* Procedure LowVideo và NormVideo;**

Sau khi gọi **LowVideo** mọi kí tự viết ra màn hình đều có độ sáng yếu đi cho tới khi gọi thủ tục **NormVideo** (Normal Video).

**\* Procedure Delay(Time);**

Tạo ra thời gian trễ Time (khoảng ms). Time là một số nguyên. Delay thường được dùng để làm chậm chương trình lại cho ta quan sát, khảo sát...

**\* Procedure Sound(F) và NoSound;**

Tạo ra dao động âm thanh với tần số là F (F: số nguyên) cho đến khi ta gọi **NoSound**;

**Bài tập chương 5**

1. Viết chương trình tính tổ hợp chập m của n phần tử:  $C_n^m$   
Chương trình viết có chương trình con.
2. Viết chương trình tính  $P_n(x) = (\dots ((a_n * x + a_{n-1}) * x + a_{n-2}) * x + \dots + a_1) * x + a_0$   
Chương trình có chương trình con.
3. Cho dãy số sau:  $a_1, a_2, \dots, a_n$ . Viết chương trình tính tổng, trung bình cộng các phần tử của dãy số đó. Chương trình viết có chương trình con.
4. Lập chương trình tính diện tích và chu vi của các hình: Tam giác biết 3 cạnh a,b,c, hình chữ nhật biết hai cạnh a,b, hình tròn biết bán kính. Chọn hình để tính thông qua câu hỏi 'Ban tính cho hình gì TG=1, CN=2, TR =3 '. Chương trình viết có sử dụng chương trình con.
5. Cho hai số nguyên x1 và x2, lập chương trình nhập x1 và x2 từ bàn phím, sử dụng tính đệ quy của chương trình con để tìm ước số chung lớn nhất của x1 và x2.

## CHƯƠNG VI

### Kiểu dữ liệu có cấu trúc: Kiểu bản ghi và Kiểu tệp

#### 1 - Kiểu bản ghi (record)

##### 1.1 - Khái niệm

Kiểu bản ghi là một kiểu dữ liệu có cấu trúc gồm một số cố định các phần tử có kiểu khác nhau.

Kiểu bản ghi dùng để mô tả các dữ liệu có nhiều thành phần khác kiểu liên kết với nhau như dữ liệu của các bảng, các cột của bảng là các thành phần, mỗi cột có một kiểu dữ liệu khác nhau, các cột liên kết với nhau để biểu diễn một nội dung nhất định.

**Ví dụ1:** Bảng lương bao gồm các cột: Số thứ tự, Họ và tên, Ngày sinh, Hệ số, Lương, Bảo hiểm xã hội, Tổng lĩnh. Mỗi dòng của bảng lương thuộc kiểu dữ liệu bản ghi. Các cột là các thành phần còn được gọi là các phần tử.

##### 1.2 - Khai báo kiểu dữ liệu bản ghi

Kiểu dữ liệu bản ghi có các phần tử liên kết với nhau. Phần tử được gọi là trường, mỗi trường có một tên, tên trường giống như tên biến. Mỗi trường thuộc một kiểu dữ liệu nào đó.

Khi báo kiểu dữ liệu bản ghi được viết trong cụm từ record ... end;

Trong cụm từ là danh sách tên các trường kèm theo sau là kiểu dữ liệu của nó.

Khai báo kiểu bản ghi như sau:

```
Type Tên_kiểu= Record
 Tên_trường1 : kiểu;
 Tên_trường2 : kiểu;
 ...
 Tên_trườngN : kiểu;
End;
```

**Ví dụ 1:** Bảng lương trong ví dụ ở mục 1 được khai báo như sau:

```
Type bang_luong = record
 Stt : Integer;
 Hoten : String[25];
 Ns : String[10];
 Heso,Luong,Bhxx,Tong : Real;
End;
Var luong1,luong2:bang_luong;
```

**Ví dụ 2:** Danh sách khách hàng bao gồm các dữ liệu như họ và tên, số nhà, phố, quận, Thành phố, số điện thoại.

```
Type khách_hang = record
 hoten : string[25];
 sonha : string[20];
 Pho, quan, thanhpho : string[30];
 tel : longint;
end;
Var Bangkh1,bangkh2: khách_hang;
```

### 1.3 - Sử dụng bản ghi

- Một trường của bản ghi coi như 1 biến, được sử dụng trong các biểu thức và các thủ tục vào ra.
- Một trường của bản ghi được chỉ định bằng cách viết sau:  
Tên\_biến.Tên\_trường

Ví dụ

```
luong1.stt:=1;
luong1.hoten:='Le Thu Ha';
luong1.luong:=luong1.heso * 210000;
bangkh1.pho:='Hang Dao';
bangkh1.tel:=8573980;
Readln(luong2.hoten);
Writeln(bangkh2.pho);
```

Biến bản ghi không được tham gia vào các biểu thức, các thủ tục vào ra, các toán tử logic, các toán tử quan hệ >, >=, <, <=.

Nếu hai biến bản ghi có cùng kiểu thì có thể tham gia các phép so sánh = (bằng) và <> (khác).

Nếu hai biến bản ghi có cùng kiểu thì có thể gán giá trị cho nhau.

```
luong1:=luong2;
bangkh1:=bangkh2;
```

### 1.4 - Câu lệnh With ... do

Khi chỉ định một trường của bản ghi ta phải viết cả tên biến và tên trường, do vậy viết các tên sẽ dài. Để không phải viết tên biến, chỉ viết tên trường ta dùng câu lệnh With ... do như sau:

```
With Tên_biến do Chuỗi_lệnh;
```

Trong chuỗi lệnh nếu chỉ ra một trường nào đó của biến đã nêu tên thì không phải viết tên biến kèm theo.

Ví dụ thay cho cách viết ở ví dụ phần 3 ta có thể sử dụng câu lệnh With...do

```
With luong1 do
begin
stt:=1;
hoten:='Le Thu Ha';
luong:=heso * 210000;
end;
```

```
With bangkh1 do
begin
pho:='Hang Dao';
tel:=8573980;
end;
```

```
With luong2 do Readln(hoten);
With bangkh2 do Writeln(pho);
```

### 1.5 - Các chương trình sử dụng bản ghi

**Bài toán 1:** Bài toán thống kê hàng nhập. Mỗi mặt hàng có các dữ liệu như tên hàng, ngày nhập, số lượng, đơn giá. Hãy nhập dữ liệu vào và in ra một bảng bao gồm các cột : tên hàng,

ngày nhập, số lượng, đơn giá, tiền của tất cả các mặt hàng đã nhập. Sau cùng in ra tổng số tiền đã nhập.

```
Chương trình
Program Thong_ke_hang;
uses crt;
Type hang=record
 ten: string[20];
 Ngay: string[10];
 sl,gia,tien : real;
end;
Var bang:array[1..30] of hang; i,n: integer; tong: real;
begin
 clrscr;
 Write)'Nhap so luong mat hang n ? ');
 readln(n);
 tong:=0;
 for i:=1 to n do
 with bang[i] do
 begin
 Write(' Tên hàng '); readln(ten);
 Write(' Ngày nhập '); readln(ngay);
 Write(' Số lượng '); readln(sl);
 Write(' Giá '); readln(gia);
 tien:=sl * gia;

 tong:=tong+tien;
 end;
 WriteLn(' Bảng thông kê hàng nhập');
 WriteLn(' Tên hàng':20,' Ngày nhập':12,' Số lượng':12,' Đơn giá':12,' Tiền':12);
 for i:= 1 to n do
 with bang[i] do
 WriteLn(ten:20, ngay:12, sl:12:2, gia:12:2, tien:12:2);
 writeln;
 writeln('Tong so tien la: ', tong: 15:2);
 readln;
 end.
```

## 2 - Kiểu tệp (File)

### 2.1 - Khái niệm tệp

Tệp dữ liệu là tập hợp các dữ liệu có liên quan với nhau và được nhóm lại với nhau tạo thành một dãy, được lưu trữ trên bộ nhớ ngoài ví dụ như đĩa từ.

Các phân tử của tệp cùng kiểu, được lưu trữ kế tiếp nhau, khi làm việc với các phân tử của tệp có con trỏ tệp. Khi mới mở tệp con trỏ tệp trỏ vào phân tử đầu tiên của tệp. Cuối tệp có dấu kết thúc tệp kí hiệu là eof(tệp).

Các phân tử của tệp f như sau:

Mỗi ô là một phân tử của tệp. Cuối tệp là dấu kết thúc tệp eof(f) (end of file)



- Có thể có các loại tệp sau:

- Tệp định kiểu.
- Tệp văn bản ( Text)
- Tệp không định kiểu

Trong các phần sau chỉ xét tệp định kiểu và tệp văn bản.

- Tệp và mảng có những điểm giống và khác nhau sau đây:

- \* Điểm giống nhau giữa tệp và mảng : tập hợp các phần tử cùng kiểu.
- \* Điểm khác nhau giữa tệp và mảng : Mảng khai báo với số phần tử xác định, còn tệp với số phần tử không xác định, tệp có thể chứa số phần tử tùy ý theo dung lượng trên đĩa.

## 2.2 - Các cách truy nhập tệp

- \* Truy nhập tuần tự và truy nhập ngẫu nhiên:

- Truy nhập tuần tự: Việc đọc một phần tử bất kỳ của tệp bắt buộc phải tuần tự đi qua các phần tử trước đây. Còn muốn thêm một phần tử vào tệp phải thêm vào cuối tệp.

Kiểu truy nhập này đơn giản trong việc tạo tệp, xử lý tệp, song nó kém linh hoạt

- Truy nhập tệp trực tiếp( direct access ): Có thể truy nhập vào bất kỳ phần tử nào trong tệp thông qua chỉ số thứ tự của phần tử trong tệp. Tùy theo từng bộ nhớ ngoài mà có thể truy nhập trực tiếp được hay không, như đĩa từ có thể truy nhập trực tiếp được, còn băng từ chỉ có thể truy nhập tuần tự không truy nhập trực tiếp được. Như vậy trong truy nhập trực tiếp có thể đọc bất kỳ phần tử nào, thêm phần tử mới thì phải thêm vào cuối tệp.

## 2.3 - Khai báo tệp định kiểu

Khai báo tệp định kiểu dùng cụm từ sau:

File of kiểu\_phần\_tử;

- Khai báo kiểu tệp:  
Type tên\_kiểu = File of kiểu\_phần\_tử;
- Khai báo biến tệp:  
Var tên\_biến : File of kiểu\_phần\_tử;

Ví dụ 1 type t= file of integer;  
var f1,f2 : t;

Ví dụ 2: type bang= record  
ten: string[25];  
Ns: string[10];  
Que: string[30];  
luong,bhxh:real;  
end;  
var f1,f2,f3: file of bang;

## 2.4 - Tạo tệp để ghi dữ liệu

- \* **Mở tệp để ghi dữ liệu**

Dùng 2 thủ tục đi liền nhau theo thứ tự như sau:

- Thủ tục Assign  
Assign(biến\_tệp, tên\_tệp);

Thủ tục này gán tên\_tệp cho biến\_tệp. Tên\_tệp theo đúng qui tắc đặt tên trong DOS mà ta đã học ở phần trên.

- Thủ tục Rewrite

```
Rewrite(biến_tệp);
```

Thủ tục này thực hiện việc mở tệp để ghi.

Ví dụ: Mở tệp có tên là 'songuyen.dat' gán cho biến tệp f để ghi dữ liệu ta viết như sau:

```
Assign(f,'songuyen.dat');
```

```
Rewrite(f);
```

Sau khi mở tệp xong thì tệp chưa có phần tử nào, tệp rỗng. Con trỏ tệp ( cửa sổ tệp ) trở vào cuối tệp (eof) . Nếu tên\_tệp trùng với tệp đã có thì tệp đó sẽ bị xoá.

\* **Ghi dữ liệu vào tệp dùng thủ tục Write**

```
Write(biến_tệp, biểu_thức1, biểu_thức2, . . ., biểu_thức n);
```

Các biểu\_thức phải có giá trị cùng kiểu với kiểu của tệp. Giá trị của các biểu thức sẽ được ghi vào tệp theo như thứ tự đã viết.

```
Write(f, 2, 4, 6, i*j+3);
```

\* **Đóng tệp bằng thủ tục Close**

```
Close(biến_tệp);
```

\* **Các ví dụ chương trình tạo tệp để ghi dữ liệu**

**Bài toán 1:** Tạo tệp có tên là 'songuyen.dat' ghi các số nguyên dương <200 mà chia hết cho 3.

Chương trình

```
Program Tao_tep_so_nguyen;
```

```
uses crt;
```

```
var i : integer; f : file of integer ;
```

```
Begin
```

```
assign(f, 'songuyen.dat');
```

```
rewrite(f);
```

```
for i:=3 to 199 do
```

```
if (i mod 3) = 0 then write(f, i);
```

```
close(f);
```

```
readln;
```

```
end.
```

**Bài toán 2:** Tạo tệp 'sach.dat' để ghi các thông tin cho các cuốn sách bao gồm các dữ liệu như tên sách, năm xuất bản, số trang, tác giả.

Chương trình

```
Program Tao_tep_sach;
```

```
uses crt;
```

```
Type tin = record
```

```
ten: string[25];
```

```
nam: integer;
```



```

 trang: longint;
 tacgia: string[25];
end;
Var i,n : integer; f: file of tin; nhap: tin;
Begin
 clrscr;
 assign(f, 'sach.dat');
 rewrite(f);
 write(' Nhap so sach n : '); readln(n);
 for i:= 1 to n do
 begin
 with nhap do
 begin
 write('Ten sach : '); readln(ten);
 write('Nam xuất ban : '); readln(nam);
 write('So trang : '); readln(trang);
 write('Ten tác gia : '); readln(tacgia);
 end;
 write(f, nhap);
 end;
 close(f);
end.

```

## 2.5 - Đọc dữ liệu từ tệp đã có

### \* **Mở tệp để đọc**

Mở tệp để đọc dùng 2 thủ tục đi liền nhau theo thứ tự sau:

- Thủ tục Assign  
assign(biến\_tệp, tên\_tệp);
- Thủ tục Reset  
Reset(biến\_tệp) ;

Thủ tục này thực hiện mở tệp để đọc.

Ví dụ 1: Mở tệp 'songuyen.dat' gắn với biến tệp f để đọc dữ liệu.

```

assign(f, 'songuyen.dat');
reset(f);

```

Ví dụ 2: Mở tệp 'sach.dat' gắn với biến tệp f1 để đọc dữ liệu.

```

assign(f1, 'sach.dat');
reset(f1);

```

### \* **Đọc dữ liệu từ tệp**

Đọc dữ liệu từ tệp được thực hiện bằng thủ tục Read như sau:

```

Read(biến_tệp, biến1, biến2, . . . , biến n);

```

Thủ tục này thực hiện đọc giá trị ở vị trí con trỏ gán cho các biến tương ứng như thứ tự đã viết, khi đọc xong con trỏ tệp lại chuyển sang phần tử tiếp theo đọc và gán cho biến khác, cứ thế đọc cho đến biến n .

Việc đọc chỉ được thực hiện khi tệp vẫn còn phần tử, tức là con trỏ chưa tới eof ( cuối tệp). do vậy trước khi đọc phải kiểm tra xem đã kết thúc tệp chưa, dùng hàm chuẩn eof như sau:

eof(biến\_tệp); hàm này cho giá trị True nếu con trỏ ở cuối tệp, ngược lại hàm cho giá trị False.

Có thể dùng 2 cấu trúc sau:

- Kiểm tra nếu tệp chưa kết thúc thì đọc  
if not eof(biến\_tệp) then read(biến\_tệp, biến);
- Đọc tất cả các phần tử của tệp  
While not eof(biến\_tệp) do  
  Begin  
    read(biến\_tệp, biến);  
    ...  
  end;

Nếu con trỏ ở cuối tệp mà vẫn đọc thì máy sẽ báo lỗi, sau đó chương trình dừng lại. Do vậy phải kiểm tra trước khi đọc.

Ví dụ while not eof(f) do

```
 begin
 read(f,x);
 writeln(x);
 end;
```

\* **Đóng tệp**

```
 Close(biến_tệp);
```

\* **Các ví dụ chương trình đọc dữ liệu từ tệp**

**Bài toán 1:** Đọc dữ liệu từ tệp 'songuyen.dat' đã tạo ở trên và hiện kết quả trên màn hình.

Chương trình

```
program Doc_tep_songuyen;
uses crt;
var i: integer; f: file of integer;
begin
 clrscr;
 assign(f, 'songuyen.dat');
 reset(f);
 while not eof(f) do
 begin
 read(f, i);
 writeln(i);
 end;
 close(f);
 readln;
end.
```

**Bài toán 2:** Viết chương trình thực hiện tạo tệp 'diem.dat' ghi lại điểm thi của thí sinh, dữ liệu bao gồm: họ và tên thí sinh, điểm toán, điểm lý, điểm hoá. Đồng thời thực hiện đọc tệp và in ra thí sinh trúng tuyển, điểm chuẩn đã được nhập vào từ bàn phím.

Chương trình

```
program Tao_doc_tep_diemts;
uses crt;
type hs = record
 ten: string[25];
 toan,ly,hoa : real;
end;
```

```

var i,n: integer; f: file of hs; nhap: hs; diemc: real;
{*****}
procedure tao; { thu tuc tao }
begin
 clrscr;
 assign(f, 'diem.dat');
 rewrite(f);
 write(' So thi sinh: '); readln(n);
 for i:=1 to n do
 begin
 with hs do
 begin
 write(' Ho va ten: '); readln(ten);
 write(' Diem toan: '); readln(toan);
 write(' Diem ly : '); readln(ly);
 write(' Diem hoa : '); readln(hoa);
 end;
 write(f, hs);
 end;
 close(f);
end; { ket thuc thu tuc tao }
{*****}
Procedure doc; { thu tuc doc }
begin
 clrscr;
 Assign(f, 'Diem.dat');
 reset(f);
 write(' Diem chuan : '); readln(diemc);
 writeln(' Danh sach thi sinh trung tuyen dai hoc ');
 while not eof(f) do
 begin
 read(f,hs);
 with hs do
 if toan+ly+hoa >= diemc then writeln(ten:25,toan:10:1,ly:10:1,hoa:10:1);
 end;
 close(f);
end; { ket thuc thu tuc doc }
{*****}
{ than chuong trinh chinh }
repeat
 clrscr;
 writeln(' 1- Tao tep');
 writeln(' 2- Doc tep');
 writeln(' 3- Ket thuc');
 write(' Hay chon mot viec ? '); readln(i);
 case i of
 1: tao;
 2: doc;
 end;

```

```

end;
until i=3;
readln;
end.

```

## 2.6 - Truy nhập tệp trực tiếp

Các phần đã xét ở trên là truy nhập tuần tự tệp có định kiểu. Trong phần này ta xét cách truy nhập trực tiếp tệp có định kiểu.

Sử dụng tất cả các thủ tục và lệnh đã nêu ở trên, ngoài ra để truy nhập trực tiếp tệp còn sử dụng một số thủ tục và hàm sau.

\* **Thủ tục Seek để dịch chuyển con trỏ tệp**

```
Seek(biến_tệp, n);
```

n có kiểu longint. Thủ tục này thực hiện chuyển con trỏ tệp tới phần tử thứ n.

Trong tệp phần tử đầu được đánh thứ tự là 0.

\* **Hàm Filepos**

```
Filepos(biến_tệp)
```

Hàm này cho vị trí hiện thời của con trỏ tệp. Vị trí đầu là 0.

\* **Hàm Filesize**

```
Filesize(biến_tệp)
```

Hàm này cho số lượng phần tử của tệp. Hàm cho giá trị 0 khi tệp rỗng.

Để thêm 1 phần tử vào tệp phải thêm vào cuối tệp. Như vậy phải dịch con trỏ tới cuối tệp bằng thủ tục seek như sau:

```
seek(biến_tệp, Filesize(biến_tệp)-1);
```

\* **Ví dụ chương trình truy nhập tệp trực tiếp**

**Bài toán 1:** Tạo tệp 'sochan.dat' ghi các số nguyên dương chẵn  $\leq 20$ . Truy nhập để sửa một phần tử bất kỳ và thêm một phần tử vào tệp.

Chương trình

```

Program truy_nhap_truc_tiep_tep;
uses crt;
var i,j : integer; f: file of byte;
{ thu tuc tao tep }
procedure tao;
begin
 clrscr;
 assign(f, 'sochan.dat');
 rewrite(f);
 for i:=1 to 20 do
 if (i mod 2) =0 then write(f, i);
 close(f);
 readln;
end; { ket thuc thu tuc tao }
{ thu tuc sua }
procedure sua;
begin
 clrscr;
 reset(f);

```

```

write(' sua phan tu thu ? '); readln(i);
seek(f, i-1);
read(f,j);
writeln(' gia tri cu: ', j);
write(' nhap gia tri moi : ');
readln(j);
seek(f, i-1);
write(f, j);
close(f);
end; { ket thuc thu tuc sua }
{ thu tuc them phan tu }
procedure them;
begin
 clrscr;
 reset(f);
 write(' gia tri moi them: ');
 readln(j);
 seek(f, filesize(f)-1);
 write(f,j);
 close(f);
 readln;
end; { ket thuc thu tuc sua }
{ thu tuc doc }
procedure doc;
uses crt;
clrscr;
reset (f);
while not eof(f) do
 begin
 read(f,i);
 writeln(i);
 end;
close(f);
end; { ket thuc thu tuc doc}

{*****}
{ than chuong trinh chinh}
repeat
 clrscr;
 writeln(' 1- Tao tep');
 writeln(' 2- Sua tep');
 writeln(' 3- Them phan tu');
 writeln(' 4- Doc tep');
 writeln(' 5- ket thuc ');
 write(' Hay chon mot viec ? '); readln(i);

 case i of
 1: tao;

```

```

2: sua;
3: them;
4: doc;
end;
until i=5;
readln;
end.

```

## 2.7 - Tập văn bản

### \* Khai báo tập văn bản

Trong Pascal có một kiểu tập đã được định nghĩa trước đó là tập văn bản, được định nghĩa với từ chuẩn TEXT.

```

Khai báo kiểu tập văn bản;
Var Ten_bien: TEXT;

```

Các phần tử của TEXT là các kiểu kí tự, được tổ chức thành các dòng với độ dài của các dòng khác nhau, nhờ có thêm các dấu hết dòng (End of Line). Đó là cặp kí tự điều khiển CR (Carriage Return, nhảy về đầu dòng, mã ASCII là 13) và LR (Line Feed: xuống dòng, mã ASCII là 10), chúng dùng để ngăn cách các dãy kí tự tương ứng với 2 dòng khác nhau.

Ví dụ: Đoạn văn bản sau:

```

HANOI
123
THUDO

```

Được chứa trong tập văn bản thành một dãy như sau:

|       |       |     |       |       |       |     |
|-------|-------|-----|-------|-------|-------|-----|
| HANOI | CR LF | 123 | CR LF | THUDO | CR LF | EOF |
|-------|-------|-----|-------|-------|-------|-----|

Do tập văn bản được tổ chức thành từng dòng nên việc ghi và đọc tập văn bản có thể tục ghi và đọc theo dòng.

### \* Ghi vào tập văn bản

Có thể ghi các giá trị kiểu Integer, real, boolean, string vào tập văn bản bằng lệnh Write hoặc Writeln, các lệnh này cho phép chuyển các giá trị kiểu nói trên sang dạng kí tự.

Có 3 dạng sau:

Dạng 1:

```

Write(biến_tập, biểu_thức_1, biểu_thức_2,..., biểu_thức_n);

```

Ghi các giá trị của biểu thức vào tập không có dấu hết dòng

Dạng 2:

```

Writeln(biến_tập, biểu_thức_1, biểu_thức_2,..., biểu_thức_n);

```

Ghi các giá trị của biểu thức vào tập có dấu hết dòng.

Dạng 3:

```

Writeln(biến_tập);

```

Ghi dấu hết dòng vào tập.

### \* Đọc dữ liệu từ tập văn bản

Chúng ta có thể đọc các kí tự, các số nguyên, số thực, boolean từ tập văn bản thông qua các thủ tục sau:

Dạng 1:

```

read(biến_tập, biến1, biến2, biến3,..., biếnn);

```

Đọc các giá trị từ tập gán cho các biến tương ứng và không sang đầu dòng tiếp theo.

Dạng 2:

```
readln(biến_tệp, biến1, biến2, biến3,..., biếnn);
```

Đọc các giá trị từ tệp gán cho các biến tương ứng và đưa con trỏ sang đầu dòng tiếp theo.

Dạng 3:

```
readln(biến_tệp);
```

Đưa con trỏ sang đầu dòng tiếp theo.

Hàm chuẩn EOLN(f) sẽ phát hiện ra dấu hết dòng của tệp f.

Hàm này cho giá trị là True nếu kết thúc dòng, ngược lại là False.

### \* Các chương trình xử lý tệp văn bản

**Bài toán 1:** Tạo tệp văn bản ghi lại n dòng văn bản nhập vào từ bàn phím. Đọc tệp đã tạo và hiện ra trên màn hình.

```
Program Tao_doc_tep_van_ban;
uses crt;
var i,n: integer; f: text; s: string;
{*****}
procedure tao; { thu tuc tao }
begin
 clrscr;
 assign(f, 'vanban.txt');
 rewrite(f);
 write(' So dong van ban: '); readln(n);
 for i:=1 to n do
 begin
 write(' Nhap dong ', i); readln(s);
 writeln(f, s);
 end;
 close(f);
end; { ket thuc thu tuc tao }
{*****}
Procedure doc; { thu tuc doc }
begin
 clrscr;
 Assign(f, 'vanban.txt');
 reset(f);
 Writeln(' Tep van ban da tao');
 while not eof(f) do
 begin
 readln(f,s);
 writeln(s);
 end;
 close(f);
end; { ket thuc thu tuc doc }
{*****}
{ than chuong trinh chinh }
repeat
 clrscr;
```

```

writeln(' 1- Tao tep');
writeln(' 2- Doc tep');
writeln(' 3- Ket thuc');
write(' Hay chon mot viec ? '); readln(i);
case i of
 1: tao;
 2: doc;
end;
until i=3;
readln;
end.

```

**Bài toán 2:** Đọc xâu văn bản từ tệp, kiểm tra xem xâu đó có đối xứng không, sau đó sắp xếp các chữ số trong xâu theo thứ tự tăng dần.

chương trình

```

PROGRAM DOC_XAU_SAP_TT_SO;
USES CRT;
TYPE MANG=ARRAY[1..200] OF CHAR;
VAR I,K,L,N:INTEGER;T:BOOLEAN;S:STRING;A:MANG;F:TEXT;
PROCEDURE SAPTANG(VAR X:MANG;M:INTEGER);
VAR J,Z:INTEGER;P:CHAR;
BEGIN
 FOR J:=1 TO M-1 DO
 FOR Z:=1 TO M-J DO IF X[Z]>X[Z+1] THEN BEGIN P:=X[Z];
 X[Z]:=X[Z+1];
 X[Z+1]:=P;
 END;
 END;
END;
BEGIN { thân chương trình chính }
CLRSCR;
ASSIGN(F,'VB.TXT');
RESET(F);
READLN(F,S);
WRITELN(' XAU DOC DUOC');
WRITELN(S);
N:=LENGTH(S);
{ tìm xau doi xung }
K:=1;L:=N;T:=TRUE;
WHILE (K<L) AND T DO
 IF S[K]=S[L] THEN BEGIN K:=K+1; L:=L-1; END
 ELSE T:=FALSE;
IF T THEN WRITELN(' XAU DOI XUNG ') ELSE WRITELN(' XAU KHONG DOI XUNG ');
{ sap cac so trong xau tang dan }
K:=0;
FOR I:=2 TO N DO IF (ORD(S[I])>=48) AND (ORD(S[I])<=57) THEN
 BEGIN
 K:=K+1;

```



```
 A[K]:=S[I];
 END;
SAPTANG(A,K);
L:=0;
FOR I:=1 TO N DO IF (ORD(S[I])>=48) AND (ORD(S[I])<=57) THEN
 BEGIN
 L:=L+1;
 S[I]:=A[L];
 END;
WRITELN('XAU DA SAP THU TU CAC SO ');
WRITELN(S);
READLN;
CLOSE(F);
END.
```

## Bài tập chương VI

1. Viết chương trình gồm 2 CT con làm các công việc sau: Ghi 1 bài thơ có n dòng, nội dung nhập vào từ bàn phím lên tệp văn bản có tên 'BAITHO.TXT' và đọc tệp văn bản đã tạo từ đĩa và cho hiện nội dung lên màn hình. Dùng câu lệnh lựa chọn để chọn công việc: 1=Ghi, 2=Doc.

2. Viết chương trình tạo tệp dữ liệu 'DIEM.DAT' chứa bảng điểm của n người và in kết quả ra màn hình. Bảng điểm gồm các cột: Họ tên, Điểm Toán, Điểm Lý, Điểm Hoá, Tổng, Loại.

Nhập vào: Họ tên, Điểm toán, Điểm lý, Điểm hoá.

Tính Tổng, Loại :

Tổng = Điểm toán + Điểm lý + Điểm hoá .

Loại = ' DO '            nếu Tổng > 19

Loại = ' TRUOT '      nếu Tổng <= 19

3. Viết chương trình tạo tệp dữ liệu 'LUONG.DAT' chứa bảng lương của n người và in kết quả ra màn hình.

Bảng lương gồm các cột: Họ tên, Ngày công, Lương ính, Thưởng, Tổng

Nhập vào : Họ tên, Ngày công, Lương chính .

Tính Thưởng, Tổng :

Thưởng = Lương chính x 2      nếu Ngày công > 25

Thưởng = Lương chính          nếu Ngày công <= 25

Tổng = Lương chính + Thưởng .

4. Viết chương trình tạo tệp dữ liệu 'HANG.DAT' chứa bảng Thống kê hàng nhập của n mặt hàng và in kết quả ra màn hình. Bảng Thống kê hàng nhập gồm các cột:

Tên, Số lượng, Đơn giá, Tiền, Loại.

Nhập vào: Tên, Số lượng, Đơn giá.

Tính Tiền, Loại :    Tiền = Số lượng x Đơn giá .

Loại = 'Cao'      nếu Tiền > 500000.

Loại = 'THAP'    nếu Tiền <= 500000.

5. Viết chương trình tạo tệp dữ liệu 'SACH.DAT' chứa bảng danh mục của n quyển sách và in kết quả ra màn hình. Bảng danh mục sách bao gồm các cột:

Tên, Năm, Tác giả, Giá, Loại.

Nhập vào : Tên, Năm, Tác giả, Giá.

Tính Loại theo công thức sau:

Loại = 'Cao'      nếu Tiền > 50000

Loại = 'TB'      nếu 20000 <= Tiền <= 50000

Loại = 'THAP'    nếu Tiền < 20000.

# CHƯƠNG VII

## ĐỒ HOẠ VÀ ÂM THANH

### 1- Đồ hoạ

#### 1.1 - Khái niệm chung

Màn hình máy vi tính hiện nay có nhiều loại khác nhau, các máy AT-386, AT-486, AT-586... thường dùng loại màn hình VGA (Video Graphic Adapter), TVGA (Top of VGA) hoặc SVGA (Supper VGA), đây là loại màn hình có thể dùng ở một trong hai chế độ: chế độ TEXT - hiển thị văn bản và chế độ GRAPHIC - hiển thị đồ hoạ.

Trong chế độ TEXT màn hình được chia thành 25 dòng và 80 cột, nếu viết kín màn hình ta có thể viết được 2000 ký tự. Chúng ta có thể thay đổi chế độ phân giải để viết ra 25 dòng x 40 cột hoặc 80 dòng x 50 cột....

Muốn vẽ hình, tô màu các hình ta phải chuyển sang chế độ đồ hoạ, trong chế độ này màn hình được xem là một ma trận điểm, tùy thuộc độ phân giải ta có thể có ma trận 640x480 điểm hoặc 1024x720 điểm.... Mỗi điểm trên màn hình được gọi là 1 Pixel tức là một phần tử ảnh

( Picture Element ), ta có thể hoàn toàn chủ động trong việc thay đổi màu sắc, cường độ sáng của từng điểm để tạo ra một bức tranh theo ý muốn. Vị trí của mỗi điểm trên màn hình được biểu diễn bởi hai tọa độ: Hoành độ và Tung độ, gốc tọa độ (0,0) là điểm ở góc trên bên trái màn hình còn điểm góc dưới bên phải có tọa độ là 639,479.

Muốn chuyển sang làm việc ở chế độ đồ hoạ, trong thư mục hiện hành (thư mục chứa chương trình Pascal) phải có các tệp GRAPH.TPU, \*.BGI và \*.CHR. Lệnh gọi đơn vị chương trình đồ hoạ phải đặt ở đầu chương trình sau từ khoá USES như ví dụ 43.

Ví dụ 43

```
Program Ve_hinh;
Uses GRAPH;
.....
```

#### 1.2 - Khởi động chế độ đồ hoạ

Trong phần thân chương trình cần phải đưa vào các thông báo về kiểu màn hình, chế độ đồ hoạ (MODE) tương ứng. Những người làm tin học ứng dụng thường không quan tâm lắm đến các thông số này do vậy dễ lúng túng khi cần khai báo. Để khắc phục nhược điểm đó trong Turbo Pascal đã có sẵn một thủ tục khởi tạo chế độ đồ hoạ là **Initgraph(var GD,GM: Integer, DP:string[n]);** Khi gọi thủ tục này với các tham số hợp lệ Initgraph sẽ tự xác định kiểu màn hình và Mode đồ hoạ tối ưu .

Người lập trình có thể tự mình khai báo các tham số GD, GM (Trong đó GD: Graph Driver - là một số nguyên xác định kiểu màn hình; GM: Graph Mode - cũng là một số nguyên xác định Mode đồ hoạ).

Trước hết trong phần khai báo biến các tham số này phải khai thuộc kiểu Integer, sau đó trong phần thân chương trình phải chọn các giá trị thích hợp cho kiểu màn hình và Mode đồ hoạ.

Thông thường chúng ta không biết chính xác kiểu màn hình và Mode đồ hoạ của máy tính đang sử dụng nên sẽ gặp lúng túng khi khai báo. Trong trường hợp này tốt nhất là để Pascal tự xác định giúp chúng ta các tham số này.

Nếu ngay sau từ khoá Begin của phần thân chương trình chúng ta khai báo

GD:= DETECT;

thì Initgraph hiểu là nó phải tự đi xác định kiểu màn hình và Mode đồ họa sao cho đạt kết quả tối ưu. Nói chung trừ những trường hợp đặc biệt, chúng ta không nên tự xác định những thông số này làm gì.

Bảng .... cho biết giá trị của các tham số GD,GM và độ phân giải màn hình của ba loại màn hình đã có, loại thông dụng nhất hiện nay là VGA.

Bảng ...

| Kiểu màn hình |         | Mode đồ họa |         | Độ phân giải      |
|---------------|---------|-------------|---------|-------------------|
| Tên           | Giá trị | Tên         | Giá trị |                   |
| CGA           | 1       | CGAC0       | 0       | 320x200           |
|               |         | CGAC1       | 1       | 320x200           |
|               |         | CGAC2       | 2       | 320x200           |
|               |         | CGAC3       | 3       | 320x200           |
|               |         | CGACHi      | 4       | 640x200           |
| EGA           | 3       | EGALO       | 0       | 640x200           |
|               |         | EGAHi       | 1       | 640x350           |
| IBM8514       | 6       | IBM8514L    | 0       | 640x480, 256 màu  |
|               |         | O           | 1       | 1024x768, 256 màu |
|               |         | IBM8514Hi   |         |                   |
| VGA           | 9       | VGALO       | 0       | 640x200           |
|               |         | VGAMed      | 1       | 640x350           |
|               |         | VGAHi       | 2       | 640x480           |

Ví dụ 44 trình bày cách sử dụng thủ tục **Initgraph** để thiết kế chương trình vẽ một đường tròn có tâm tại chính giữa màn hình và bán kính là 50 Pixel. Các tham số GD,GM sẽ do Pascal tự xác định, các tệp điều khiển đồ họa đặt trong thư mục TP trên đĩa cứng C.

Ví dụ 44

```
Program Ve_hinh_tron;
Uses graph;
Var
GD,DM: Integer;
BEGIN
GD:= detect;
Initgraph(GD,GM,'C:\tp\bgi');
If graphresult <> grok then halt(1);
Circle(320,240,50);
Readln;
CloseGraph;
END.
```

Nếu chúng ta muốn tự mình khai báo GD và GM (Ví dụ: màn hình VGA, kiểu đồ họa là độ phân giải cao, tệp tin đồ họa chứa trong thư mục TP5 trên đĩa A) thì bỏ lệnh GD:=DETECT Và viết lại đoạn đầu của ví dụ 44 như sau:

.....

GD:=VGA;  
 GM:=VGAHi;  
 INITGRAPH(GD,GM,'C:\TP\BGI');

**Chú ý:** Có thể thay thế tên màn hình bằng giá trị tương ứng đã có trong bảng.

Tham số DP ( Driver Path ) là đường dẫn tới thư mục chứa các tệp điều khiển kiểu màn hình đồ họa, trong ví dụ trên được khai là 'c:\tp\bgi' nghĩa là ổ đĩa C thư mục TP, còn BGI là đuôi tệp điều khiển. Trong trường hợp chúng ta không khai báo gì cho DP, nghĩa là để một chuỗi rỗng "" thì Pascal sẽ tự tìm các tệp điều khiển đồ họa trong thư mục chủ (là nơi mà Pascal được khởi động).

### 1.3 - Kiểm tra lỗi đồ họa

Dòng thứ 8 trong ví dụ 44

**If graphresult <> grok then halt(1);**

Là câu lệnh kiểm tra lỗi đồ họa. Việc kiểm tra được thực hiện thông qua hàm của hàm **graphresult**. Lỗi đồ họa thường phát sinh khi không tìm thấy các trình điều khiển đồ họa, hoặc khi chúng ta khai báo không đúng các tham số GD, GM, DP. Nếu phát hiện ra lỗi hàm Graphresult sẽ nhận một trong các giá trị cho trong bảng sau, mỗi giá trị ứng với một hằng chuỗi ký tự:

**Bảng lỗi của hàm Graphresult**

| Giá trị | Hằng chuỗi       | ý nghĩa                                           |
|---------|------------------|---------------------------------------------------|
| 0       | Grok             | Tốt, không có lỗi                                 |
| -1      | Grnoinitgraph    | Không tìm thấy đơn vị đồ họa                      |
| -2      | Grnotdetected    | Không có phần cứng đồ họa                         |
| -3      | GrFilenotfound   | Không tìm thấy các tệp điều khiển màn hình đồ họa |
| -4      | GrInvalidDriver  | Trình điều khiển hỏng, không hợp lệ               |
| -5      | GrNoloadMem      | Bộ nhớ động (RAM) bị thiếu                        |
| -6      | GrNoscanMem      | Không thực hiện được việc kiểm tra bộ nhớ         |
| -7      | GrNoFloodMem     |                                                   |
| -8      | GrFontNoFound    | Không tìm thấy các tệp chứa font chữ              |
| -9      | GrNoFontMem      | Không đủ bộ nhớ để nạp Font chữ                   |
| -10     | GrInvalidMode    | Kiểu đồ họa không hợp lệ                          |
| -11     | GrError          | Lỗi đồ họa (tổng quát )                           |
| -12     | GrIOError        | Lỗi các đường nhập, xuất đồ họa                   |
| -13     | GrInvalidFont    | Kiểu chữ không hợp lệ                             |
| -14     | GrInvalidFontNum | Số hiệu kiểu chữ không hợp lệ                     |

Như vậy nếu giá trị của hàm Graphresult khác Grok nghĩa là có một lỗi nào đó thì chương trình sẽ bị dừng bằng lệnh Halt(1), còn nếu tốt thì tiếp tục làm việc.

#### 1.4 - Một số thủ tục cơ bản để vẽ hình

- \* **MOVETO(x,y)** : Di chuyển con trỏ đến tọa độ x,y (x là hoành độ, y là tung độ)  
x,y là các giá trị kiểu Integer, với màn hình VGA thì  $0 \leq x \leq 639$ ,  $0 \leq y \leq 479$
- \* **LINETO(x,y)**: Vẽ một đường thẳng từ vị trí con trỏ hiện thời tới tọa độ x,y kết thúc quá trình vẽ con trỏ nằm tại tọa độ mới.  
Pascal có sẵn hai hàm để xác định tọa độ góc dưới bên phải màn hình đó là Getmaxx và Getmaxy. Để vẽ đường chéo của màn hình từ tọa độ 0,0 ta có thể viết các lệnh  
MOVETO(0,0);  
LINETO(Getmaxx,Getmaxy);
- \* **LINE(x1,y1,x2,y2)** : Thủ tục này vẽ một đường thẳng từ tọa độ x1,y1 đến tọa độ x2,y2
- \* **LINEREL(dX,dY)**: Vẽ đường thẳng từ vị trí hiện thời ( tọa độ x,y) tới tọa độ x+dx, y+dy.
- \* **CIRCLE(x,y,r)**: Vẽ đường tròn tâm tại tọa độ x,y bán kính bằng r Pixel
- \* **PUTPIXEL(x,y, n)**: Thủ tục này sẽ vẽ một điểm sáng tại tọa độ x,y với màu là n. Giá trị n lấy trong khoảng 0-15 hoặc viết trực tiếp tên màu theo tiếng Anh.

#### • Thiết lập màu đồ họa

Để định màu trong chế độ đồ họa ta có thể dùng hai thủ tục sau đây :

**g. SETCOLOR(n)** : Định màu cho các nét vẽ

**h. SETBKCOLOR(n)** : Định màu nền cho nét vẽ

Tham số n cho các thủ tục 7 và 8 xem trong bảng sau

Bảng .. Mã màu và tên màu

| Mã màu | Tên tiếng Anh | Tên tiếng Việt    |
|--------|---------------|-------------------|
| 0      | Black         | Đen               |
| 1      | Blue          | Xanh da trời      |
| 2      | Green         | Xanh lá mạ        |
| 3      | Cyan          | Xanh lơ           |
| 4      | Red           | Đỏ                |
| 5      | Magenta       | Tím               |
| 6      | Brown         | Nâu               |
| 7      | Lightgray     | Xám nhạt          |
| 8      | Darkgray      | Xám sẫm           |
| 9      | Lightblue     | Xanh da trời nhạt |
| 10     | Lightgreen    | Xanh lá mạ nhạt   |
| 11     | Lightcyan     | Xanh lơ nhạt      |
| 12     | Lightred      | Đỏ nhạt           |
| 13     | Lightmagenta  | Tím nhạt          |
| 14     | Yellow        | Vàng              |
| 15     | White         | Trắng             |

#### Chú ý:

- Lệnh định màu phải đặt trước lệnh vẽ hình
- Một màu đã định sẽ có tác dụng cho đến khi có lệnh định màu mới

- **Vẽ hình khép kín và tô màu**

\* **RECTANGLE(x1,y1,x2,y2)** : Vẽ khung hình chữ nhật toạ độ góc trên bên trái là x1,y1 , toạ độ góc dưới bên phải là x2,y2.

\* **BAR(x1,y1,x2,y2)** : Vẽ một hình chữ nhật góc trên bên trái có toạ độ x1,y1 góc dưới bên phải có toạ độ x2,y2. Khi dùng kết hợp với thủ tục số 11 sẽ đồng thời cho phép kẻ các vân hoa trên nền và tô màu cho nền.

\* **SETFILLSTYLE(n1,n2)** : Thủ tục định vân hoa và màu nền.

n1 là một giá trị nguyên với  $0 \leq n1 \leq 11$  : định kiểu vân hoa (xem bảng ...)

n2 là số hiệu mã màu đã giới thiệu  $0 \leq n2 \leq 15$

**Chú ý:**

- Thủ tục 11 chọn màu nền và vân hoa chỉ có hiệu lực khi chúng ta sử dụng kèm theo các thủ tục vẽ hình phẳng sau đây:

Hình chữ nhật (Bar), hình hộp chữ nhật (Bar3D), hình quạt (Pieslice), đa giác có toạ độ nhập vào mảng (Fillpoly), Tô màu cho miền (Floodfill).

- Khi vẽ đường tròn hoặc khung chữ nhật thì không thể tạo hoa văn và tô màu trong nền.

- Thủ tục SETFILLSTYLE(n1,n2) cần đưa vào trước các thủ tục vẽ hình đã nêu trong mục chú ý. Giá trị của màu và kiểu vân hoa sẽ được giữ cho đến khi ta định nghĩa lại.

Bảng... Giá trị và ý nghĩa của tham số n1

| Giá trị | Tên tiếng Anh  | ý nghĩa                                   |
|---------|----------------|-------------------------------------------|
| 0       | EmptyFill      | Không tạo hoa văn bên trong hình          |
| 1       | SolidFill      | Tạo hoa văn bằng nét liền                 |
| 2       | LineFill       | Tạo bằng nét gạch nối ----                |
| 3       | LTSlashFill    | Tạo hoa văn bằng nét ///                  |
| 4       | SlashFill      | Tạo hoa văn bằng nét /// đậm              |
| 5       | BKSlashFill    | Tạo hoa văn bằng nét \\\\ đậm             |
| 6       | LTBKSlashFill  | Tạo hoa văn bằng nét \\\                  |
| 7       | HatchFill      | Kẻ lưới hình ô vuông                      |
| 8       | XHatchFill     | Kẻ lưới hình ô vuông xoay 90 độ           |
| 9       | InterleaveFill | Tô bằng nét đứt quãng                     |
| 10      | WideDotFill    | Tạo hoa văn bằng dấu chấm thưa            |
| 11      | CloseDotFill   | Tạo hoa văn bằng dấu chấm dày (liền nhau) |

### 1.5 - Viết chữ trong chế độ đồ hoạ

Khi đã chuyển sang làm việc ở chế độ đồ hoạ ta không thể viết chữ bình thường như trong chế độ văn bản. Muốn viết chữ trong các hình vẽ ta sử dụng một số thủ tục sau đây:

\* **OUTTEXT(chuỗi)** : Thủ tục này sẽ cho hiện chuỗi ký tự tại vị trí con trỏ hiện thời. Chuỗi có thể viết trực tiếp hoặc thông qua biến chuỗi như trong ví dụ 45 sau đây:

Ví dụ 45

.....

Var

chuviet : string[30]

Begin

outtext('cong hoa xa hoi chu nghia ...');

```
chuviet:='Viet nam dan chu cong hoa';
outtext(chuviet);
```

.....

\* **OUTTEXTXY(x,y,chuoi)** : thủ tục này sẽ viết ra chuỗi ký tự tại tọa độ x,y.

\* **SETTEXTSTYLE(Kiểu chữ, Chiều viết, Kích thước);**

**Kiểu chữ** là một tham số nguyên nhận giá trị trong khoảng 0-4

**Chiều viết** chỉ nhận 1 trong hai giá trị : 0 nằm ngang; 1 thẳng đứng

**Kích thước** Là hệ số phóng to chữ có thể chọn từ 0-10

Để chấm dứt chế độ đồ họa trở về chế độ văn bản ta dùng thủ tục CLOSEGRAPH. Sau đó muốn quay lại chế độ đồ họa ta lại phải gọi lại INITGRAPH.

Trong một số trường hợp để chuyển nhanh giữa chế độ đồ họa và văn bản chúng ta có thể dùng hai thủ tục sau đây:

- RESTORECRTMODE; Tạm ngừng chế độ đồ họa chuyển sang chế độ văn bản.

- SETGRAPHMODE(n); Ngắt chế độ văn bản đã tạo ra bởi Restorecrtmode thiết lập trở lại chế độ đồ họa. Tham số n có thể lựa chọn trong khoảng 0-2. Ví dụ 46 dưới đây trình bày cách sử dụng các thủ tục này.

```
Program dohoa_text;
uses crt,graph;
var
gd,gm:integer;
begin
gd:=detect;
initgraph(gd,gm,'a:\tp5\bgi');
if graphresult<>grok then halt(1);
moveto(0,0); setcolor(5);
lineto(300,300); delay(2500);
circle(400,300,100); delay(1500);
restorecrtmode; (* Chuyển về chế độ văn bản *)
gotoxy(20,20);textcolor(9);
write('Happy New Year');
readln;
setgraphmode(2); (* Trở về chế độ đồ họa với n=2 cho màn hình VGA *)
setcolor(blue);
circle(100,100,50);
delay(2000);
restorecrtmode; (* Chuyển sang chế độ văn bản lần thứ hai *)
textcolor(3);
gotoxy(20,0);write('DAI HOC NONG NGHIEP I HA NOI');
readln;
closegraph; (* Kết thúc chế độ đồ họa *)
end.
```

Việc sử dụng các thủ tục đồ họa không có gì phức tạp, với một chút cố gắng bạn có thể vẽ được những hình rất đẹp theo mong muốn. Dưới đây là một chương trình vẽ đồ thị hình sin. Chạy chương trình ta sẽ thấy ba đường hình sin với các biên độ và màu sắc khác nhau.

Ví dụ 46



```

Program Do_thi_hinh_sin;
uses graph,crt;
const m=0.1;
Var t3,t4,t1,n,t2,gd,gm:integer; t,x,y,z:real;
Begin
gd:=detect;
Initgraph(gd,gm,'a:\tp5\bgi');
if graphresult<>gok then Halt(1);
x:=0; t3:=100; n:=0; t2:=10;
 while t2<=600 do
 Begin
setcolor(green);
y:=sin(x);
t1:=round(y*50);
t3:=round(y*70);
t4:=round(y*100);
t1:=200-t1;
t3:=200-t3;
t4:=200+t4;
moveto(10,200);
lineto(620,200);
line(10,80,10,300);
setttextstyle(3,0,3);
outtextxy(610,205,'x');
setttextstyle(3,0,3);
outtextxy(15,75,'y');
setttextstyle(4,0,3);
setcolor(5);
outtextxy(200,300,'do thi ham sin(x)');
setcolor(12);
moveto(10,200);
putpixel(t2,t1,11);
putpixel(t2,t3,14);
setcolor(red);
putpixel(t2,t4,random(14));
setcolor(12);
delay(5);
x:=x+0.07;
t2:=t2+1;
end;
repeat until keypressed;
t1:=1;
t2:=200;
while t1<=220 do
begin
line(340,240,round(sqrt(440*440-t1*t1)),t1);
t1:=t1+1;
delay(15);

```

```

end;
repeat until keypressed;
closegraph;
End.

```

Chương trình dưới đây thiết kế một đồng hồ ba kim , tốc độ chạy của kim giây tùy thuộc vào lệnh DELAY(n) , nếu chọn DELAY(1000) thì cứ 1 giây kim giây chuyển một vị trí. Khi nhập chương trình vào máy cần lưu ý khai báo lại đường dẫn đến thư mục chứa các tệp \*.BGI

```

Program VEDONGHO;
uses crt,graph;
var
x,y, maxx,maxy, gd,gm,color,i,j,t:integer;
N:real;
LAM,TT:CHAR;
begin
gd:=detect;
initgraph(gd,gm,'c:\tp5\BGI');
setcolor(5);
rectangle(30,20,610,450);
rectangle(31,21,609,449);
rectangle(32,22,608,448);
setfillstyle(9,2);
bar(33,23,607,447);
setcolor(red);
setbkcolor(red);
for i:=1 to 10 do circle(320,240,i);
setcolor(11);
setbkcolor(white);
for i:=11 to 80 do circle(320,240,i);
setcolor(14);
setbkcolor(white);
for i:=80 to 160 do circle(320,240,i);
setcolor(white);
for i:=160 to 200 do circle(320,240,i);
setcolor(11);
circle(320,240,79);
circle(320,240,80);
setcolor(4);
circle(320,240,159);
circle(320,240,160);
setttextstyle(3,0,4);
outtextxy(310,40,'XII');
outtextxy(405,60,'I');
outtextxy(470,120,'II');
outtextxy(490,200,'III');
outtextxy(480,290,'IV');

```

```

outtextxy(410,370,'V');
outtextxy(310,400,'VI');
outtextxy(210,370,'VII');
outtextxy(135,290,'VIII');
outtextxy(130,210,'IX');
outtextxy(155,130,'X');
outtextxy(220,60,'XI');
setcolor(blue);
Settextstyle(4,0,5);
outtextxy(230,100,'DIAMON');
setcolor(random(14));
for i:=1 to 20 do
circle(320,360,i);
setttextstyle(1,0,2);
setcolor(5);
outtextxy(200,450,'Copyright by Dr. Duong Xuan Thanh');
 for i:=1 to 20 do
 begin
 setcolor(random(14));
 circle(320,360,i);
 end;
for i:=1 to 20 do
begin
setcolor(random(14));
circle(320,360,i);

end;
for t:=0 to 12 do {----- Kim gio -----}
begin
setcolor(12);
moveto(320,240);
setlinestyle(0,0,3);
SetWriteMode(xorput);
linerel(round(110*cos((t*30-89)*pi/180)),round(110*sin((t*30-89)*pi/180)));
 moveto(320,240);
linerel(round(110*cos((t*30-90)*pi/180)),round(110*sin((t*30-90)*pi/180)));
 moveto(320,240);
linerel(round(110*cos((t*30-91)*pi/180)),round(110*sin((t*30-91)*pi/180)));
 moveto(320,240);
linerel(round(110*cos((t*30-92)*pi/180)),round(110*sin((t*30-92)*pi/180)));
 for i:=0 to 60 do { -----Kim phut -----}
 begin
 setcolor(12);
 moveto(320,240);
 setlinestyle(0,0,3);
 SetWriteMode(xorput);
 linerel(round(130*cos((i*6-89)*pi/180)),round(130*sin((i*6-89)*pi/180)));
 moveto(320,240);

```

```

linerel(round(130*cos((i*6-90)*pi/180)),round(130*sin((i*6-90)*pi/180)));
moveto(320,240);
linerel(round(130*cos((i*6-91)*pi/180)),round(130*sin((i*6-91)*pi/180)));
(*-----Kim giay-----*)
for j:=0 to 360 do
begin
moveto(320,240);
setlinestyle(0,0,3);
SetWriteMode(XORPut);
setcolor(12);
linerel(round(150*cos((j-90)*pi/180)),round(150*sin((j-90)*pi/180)));
moveto(320,240);
linerel(round(150*cos((j-91)*pi/180)),round(150*sin((j-91)*pi/180)));
delay(1000); moveto(320,240);
linerel(round(150*cos((j-90)*pi/180)),round(150*sin((j-90)*pi/180)));
moveto(320,240);
linerel(round(150*cos((j-91)*pi/180)),round(150*sin((j-91)*pi/180)));
end;
moveto(320,240);
linerel(round(130*cos((i*6-89)*pi/180)),round(130*sin((i*6-89)*pi/180)));
moveto(320,240);
linerel(round(130*cos((i*6-90)*pi/180)),round(130*sin((i*6-90)*pi/180)));
moveto(320,240);
linerel(round(130*cos((i*6-91)*pi/180)),round(130*sin((i*6-91)*pi/180)));
end;
moveto(320,240);
linerel(round(110*cos((t*30-89)*pi/180)),round(110*sin((t*30-89)*pi/180)));
moveto(320,240);
linerel(round(110*cos((t*30-90)*pi/180)),round(110*sin((t*30-90)*pi/180)));
moveto(320,240);
linerel(round(110*cos((t*30-91)*pi/180)),round(110*sin((t*30-91)*pi/180)));
moveto(320,240);
linerel(round(110*cos((t*30-92)*pi/180)),round(110*sin((t*30-92)*pi/180)));
end;
repeat until keypressed;
END.

```

## 2 – Âm thanh

Trong các máy PC thông dụng công suất của loa rất nhỏ do đó việc tạo âm thanh chỉ có tính chất biểu diễn. Muốn tạo ra âm thanh cao thấp khác nhau ta chỉ cần đưa vào loa các xung điện với tần số khác nhau. Turbo Pascal đã có ba thủ tục thiết kế sẵn để làm việc này

SOUND(n) : tạo ra âm thanh với tần số n , ở đây n phải là một số nguyên dương

DELAY(n) : Kéo dài tín hiệu âm thanh trong khoảng thời gian n miligiây nếu sau đó có thủ tục ngắt âm Nosound.

NOSOUND : ngắt tín hiệu âm thanh

Cần chú ý rằng khi có thủ tục sound và Delay tín hiệu âm thanh sẽ được phát ra chừng nào chưa có Nosound mặc dù ta đã định khoảng thời gian trễ qua thủ tục Delay.

Để tạo một bản nhạc chúng ta cũng cần có một chút kiến thức về nhạc lý. Độ cao thấp trong âm nhạc được phân thành các quãng tám. Quãng tám trung tần thì nốt Đô có tần số 512 Hz, quãng tám trầm hơn tần số của nốt Đô sẽ là 256 Hz, còn quãng tám cao hơn nốt Đô có tần số 1024 Hz.

Tên các nốt nhạc theo quy ước là:

Quãng tám trầm : đồ, rê, mi, fa, son, la, xi

Quãng tám trung : đô, rê, mi, fa, son, la, xi

Quãng tám cao : đồ, rê, mi, fa, son, la, xi

Về cao độ thì các nốt Mi - Fa và Xi - Đô cách nhau nửa cung các nốt còn lại cách nhau một cung. Nếu tính từ nốt Đô (tần số 256) đến nốt Đô (tần số 512) khoảng cách là 6 cung như vậy sự chênh lệch về tần số trong mỗi cung sẽ là  $(512-256)/6 = 43$  Hz. Giữa các nốt Mi - Fa và Xi - Đô chênh lệch chỉ là 22 Hz (vì các nốt này cách nhau nửa cung). Ngoài ra cũng cần biết cách tạo ra các nốt thăng, giáng để có thể tạo ra một bản nhạc theo ý muốn.

Nếu chúng ta chỉ muốn nghe một bản nhạc phát ra từ máy thì có thể tạo nên một chương trình thiết kế các nốt nhạc sau đó ghép chúng lại thành bản nhạc tùy ý.

Chương trình dưới đây thiết kế bản nhạc Làng tôi, các chương trình con T0, T1...T5 tạo ra trường độ, ví dụ T0 là nốt tròn, T1 là nốt đen... các chương trình con khác tạo ra các nốt nhạc của hai quãng tám trầm và trung, trong đó có một số nốt thăng và giáng.

Program Vancao;

uses crt;

```
procedure T0; Begin delay(20000); Nosound; End;
procedure T1; Begin delay(10000); Nosound; End;
procedure T2; Begin delay(5000); Nosound; End;
procedure T3; Begin delay(2500); Nosound; End;
procedure T4; Begin delay(1250); Nosound; End;
procedure T5; Begin delay(30000); Nosound; End;
```

```
Procedure nt(i:integer);
```

```
Begin Sound(i); End;
```

```
Procedure do1; Begin nt(256); End; {quang tam tram}
```

```
Procedure do11; Begin nt(270); End; {not do thang}
```

```
Procedure re1; Begin nt(299); End;
```

```
Procedure mi1; Begin nt(342); End;
```

```
Procedure fa1; Begin nt(363); End;
```

```
Procedure son1; Begin nt(406); End;
```

```
Procedure la1; Begin nt(449); End;
```

```
Procedure si1; Begin nt(492); End;
```

```
Procedure do2; Begin nt(534); End; {quang tam trung}
```

```
Procedure re2; Begin nt(600); End; {not Re giang}
```

```
Procedure mi2; Begin nt(685); End;
```

```
Procedure fa2; Begin nt(634); End;
```

```
Procedure son2; Begin nt(677); End;
```

```
Procedure la2; Begin nt(712); End;
```

```
Procedure si2; Begin nt(755); End;
```

```
Procedure Langtoi;
```

```
Begin
```

```

clrscr; gotoxy(25,12); textcolor(14); textbackground(red);
Write(' LANG TOI * Nhạc tien chien ');
Repeat
do11;t0;mi1;t1;son1;t0;la1;t1;son1;t0;
son1;t1;do2;t1;si1;t1;la1;t1;son1;t0;
la1;t1;son1;t1;fa1;t1;mi1;t1;son1;t0;t1;
do11;t0;re1;t1;mi1;t0;son1;t1;do2;t0;re2;t1;mi2;t0;t1;
re2;t1;do2;t1;re2;t1;do2;t1;son1;t0;mi1;t1;son1;t1;do2;t5;t1;
do2;t1;do2;t2;la1;t1;la1;t1;si1;t1;la1;t1;son1;t0;t1;
fa1;t1;fa1;t1;la1;t0;t4;la1;t1;mi1;t0;re1;t1;son1;t5;t1;
do1;t0;re1;t1;mi1;t0;fa1;t1;son1;t0;mi1;t1;re1;t5;
do1;t1;do2;t0;si1;t1;re2;t0;son1;t1;do2;t5;

Until keypressed; Nosound;
End;
BEGIN
Langtoi;
END.

```

Dưới đây là ví dụ thiết kế bàn phím thành các phím của một chiếc đàn dương cầm. Các nốt đô, rê, mi, fa, son, la , xi đồ sẽ bấm các chữ cái tương ứng ( D, R, M, F, S, L, X, Z). Muốn dừng âm ta bấm phím P (PAUSE), còn muốn dừng chương trình bấm E (EXIT).

```

Program nhac;
uses crt,graph;
var
n :char; i,j:integer;
begin
clrscr;
textcolor(14); textbackground(white);
for i:=1 to 6 do
begin
gotoxy(16,i+3);
for j:=1 to 51 do write(chr(177));
end;
gotoxy(17,5);
textcolor(red);
write('Do-D, Re-R, Mi-M, Fa-F, Son-S, La-L, Xi-X, Do2-Z');
gotoxy(33,6); write(' Re2-W, Mi2-T ');
gotoxy(22,7);
textcolor(blue);
writeln(' P->Ngat am, E-> Dung chuong trinh ');
gotoxy(22,8);
textcolor(5);
writeln('Moi ban choi moi ban nhac minh ua thich ');
repeat
n:= readkey;
if n = 'd' then begin nosound; delay(3); sound(523); end;

```

```
if n = 'r' then begin nosound; delay(10); sound(587); end;
if n = 'm' then begin nosound; delay(10); sound(659); end;
if n = 'f' then begin nosound; delay(10); sound(698); end;
if n = 's' then begin nosound; delay(10); sound(784); end;
if n = 'l' then begin nosound; delay(10); sound(880); end;
if n = 'x' then begin nosound; delay(10); sound(988); end;
if n = 'z' then begin nosound; delay(3); sound(1050); end;
if n = 'w' then begin nosound; delay(3); sound(1190); end;
if n = 't' then begin nosound; delay(3); sound(1300); end;
if n = 'p' then nosound;
until upcase(n) = 'E';
nosound;
end.
```

## Bài tập chương VII

1. Viết chương trình vẽ n hình vuông lồng nhau.
2. Viết chương trình vẽ bàn cờ hình vuông có 64 ô ( mỗi chiều là 8), xen kẽ cứ một ô đỏ rồi đến 1 ô trắng.
3. Viết chương trình vẽ đồ thị hình Cosin trong lượng giác.

## Tài liệu tham khảo

1. Bùi Thế Tâm. Windows 2000, Word 2000, Excel 2000, Powerpoint 2000. Nhà xuất bản Giao thông vận tải , năm 2003.
2. Bùi Thế Tâm. Turbo Pascal 7.0. Nhà xuất bản thống kê, năm 1996
3. Charles rubin. Microsoft Word (bản dịch). Nhà xuất bản trẻ, năm 1999.
4. Đỗ Xuân Lôi. Cấu trúc dữ liệu và Giải thuật. Nhà xuất bản Khoa học và Kỹ thuật, năm 1998.
5. Đỗ Xuân Thụ - Hồ Khánh Lâm. Kỹ thuật vi xử lý và máy vi tính. Nhà xuất bản giáo dục, năm 2000.
6. Kain Richard. Advance Computer architecturre: A system Design Approach. Prentice Hall USA, 1996
7. Hoàng Hồng. Giáo trình tin học văn phòng. Nhà xuất bản Giao thông vận tải, năm 2002.
8. Hayes John. Computer Architecturre and Orgnizassion. McGaw Hill USA. 1998
9. Nguyễn Nam Trung. Cấu trúc máy tính và thiết bị ngoại vi. Nhà xuất bản Khoa học và kỹ thuật, năm 2003.
10. Nguyễn Xuân Phong - Trương Lê Ngọc Thường. Tin học đại cương. Nhà xuất bản Thanh niên, năm 2002.
11. Nguyễn Xuân Huy. Thuật toán . Nhà xuất bản Thống kê, năm 1988
12. Xuân Thành. Internet, Explorer 6.0. Nhà xuất bản Thống kê, năm 2002.