

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.

Bài tập lập trình C++

Mục lục

Trang 1

1. MÃ HÓA THÔNG ĐIỆP
2. GIẢI PHƯƠNG TRÌNH BẬC NHẤT
3. TÍNH CĂN BẬC HAI THEO PHƯƠNG PHÁP LẬP NEWTON
4. CẤU TRÚC VÀ CÁC HÀM THAO TÁC TRÊN SỐ PHỨC
5. DÃY TĂNG DẦN
6. DÃY TĂNG CÓ TỔNG DÀI NHẤT
7. QUẢN LÝ SINH VIÊN
8. GIẢI PHƯƠNG TRÌNH BẬC HAI
9. MA PHƯƠNG
10. FILE VÀ HỆ THỐNG

Trang 2

1. SẮP XẾP MẢNG
2. Một ví dụ về Đa hình
3. Tiếp một ví dụ về Đa hình
4. Tổng hai ma trận
5. Một ví dụ về sử dụng template và quá tải toán tử Nhập xuất
6. Ví dụ về quá tải toán tử
7. Đếm số lần xuất hiện của các ký tự trong chuỗi
8. Bài toán Ancarokhi
9. Chứng minh đẳng thức An Casi
10. Hiện bảng mã ASCII
11. In ra năm âm lịch tương ứng với năm nhập vào.
12. In ra bảng cửu chương
13. Nhập chuỗi và in chuỗi
14. Giải hệ phương trình bậc nhất.
15. Tính thứ của ngày

Trang 3

1. Chuyển số La Mã sang số Ả rập
2. Chuyển năm sang số La Mã
3. Thuật toán sắp xếp bằng Radix sort
4. Danh sách liên kết đơn (Thuật toán vừa chèn vừa sắp xếp)
5. Quá tải toán tử nhập xuất và sử dụng template
6. Chương trình đếm số ký tự trong một chuỗi ASCII
7. Biểu diễn số dưới dạng bit
8. Đảo chuỗi

9. Chương trình xem tập tin
10. Giải bài toán trâu ăn cỏ
11. Loại bỏ khoảng trống thừa trong chuỗi
12. Tìm tất cả các ước của một số N
13. Bội số chung và ước số chung
14. Trộn 2 dãy giảm thành một dãy tăng
15. Tính tích 2 ma trận:
16. In danh sách các số hoàn hảo nhỏ hơn số N nhập từ user

Trang 4

1. Bài in ra lịch của một năm bất kỳ lớn hơn 1700
2. Bài tập kiểm tra dấu ngoặc đúng.
3. Bài toán Tám Hoàng Hậu
4. In ra số Hex tương ứng với một số nguyên dương
5. Liệt kê các hoán vị của N phần tử
6. In chuỗi theo các từ mỗi từ một dòng
7. In ra chữ số hàng trăm hàng chục hàng đơn vị
8. Tìm phần tử lớn nhất nhỏ nhất trong mảng một chiều
9. Tính tổ hợp chập K của N phần tử
10. Chương trình đọc số có 1,2 hoặc 3 chữ số.
11. Tính số ngày trong một tháng trong một năm bất kỳ
12. Bài kiểm tra số nguyên tố
13. Tìm max min của 4 số
14. Tìm n số Fibonacci đầu tiên

Trang 5

1. (Ngân hàng)Tìm số tiền nhận trong n tháng khi biết lãi suất
2. In ra dãy số ngược so với dãy số nhập vào
3. Trò chơi 8 hòn bi
4. Kiểm tra số đối xứng
5. Điền giá trị cho một mảng vuông theo chiều kim đồng hồ
6. In hình tam giác
7. Trộn hai mảng tăng dần thành một mảng tăng dần
8. Tìm vị trí đầu và vị trí cuối của một số trong một dãy số
9. Tính $x^1/1! + x^2/2! + x^3/3! + \dots + x^n/n!$
10. Trình bày các bước chuyển n đĩa từ cọc A sang cọc C trong bài toán Tháp Hà Nội dùng 3 đĩa
11. Trình bày các bước chuyển n đĩa từ cọc A sang cọc C trong bài toán Tháp Hà Nội dùng 4 đĩa

MÃ HÓA THÔNG điệp

C code:

[Lựa chọn code](#) | [Ẩn/Hiện code](#)

```
#include <stdio.h>
#include <ctype.h>
#include <alloc.h>

char *crypt(char *tdiep, int column)
{
    char tam[255], *result;
    int i = 0, k = 0, n, j=0;

    while(tdiep[i] != 0)
    {
        if (isalnum(tdiep[i]))
            tam[k++] = tdiep[i];
        i++;
    }
    tam[k] = 0;
    result = (char *)malloc(k+1);
    for (i=0; i<column; i++)
    {
        n = 0;
        while(n+i < k)
        {
            result[j++] = tolower(tam[n+i]);
            n += column;
        }
    }
    result[k] = 0;
    return result;
}

void main()
{
    char thongdiep[255], *mahoa;
    int col;

    printf("\nNhap thong diep can ma hoa : ");
    gets(thongdiep);
    printf("\nCho biet so cot : ");
    scanf("%d", &col);
    mahoa = crypt(thongdiep, col);
    printf("\nThong diep da duoc ma hoa thanh : %s", mahoa);
    getch();
}
```

GIẢI PHƯƠNG TRÌNH BẬC NHẤT

C code:

```
#include <stdio.h>

void main()
{
    float a, b;

    printf("\nGiai phuong trinh bac nhat AX + B = 0");
    printf("\nCho biet ba he so A B : ");
    scanf("%f%f", &a, &b);

    if (a==0)
    if (b!=0)
    printf("Phuong trinh vo nghiem");
    else
    printf("Phuong trinh co nghiem khong xac dinh");
    else
    printf("Dap so cua phuong trinh tren = %f", -b/a);
    getch();
}
```

TÍNH CĂN BẬC HAI THEO PHƯƠNG PHÁP LẶP NEWTON

```
#include <stdio.h>
#include <math.h>

void main()
{
    double a, xn, ketqua;

    printf("\nNhap vao so muon tinh can bac hai : ");
    scanf("%lf", &a);
    xn = (a+1)/2;
    do {
        ketqua = xn;
        xn = 0.5 * (xn + a/xn);
    } while (fabs(xn-ketqua) > 0.0001);
    printf("\nKet qua = %lf", xn);
    getch();
}
```

CÁU TRÚC VÀ CÁC HÀM THAO TÁC TRÊN SỐ PHỨC

```
#include <math.h>

typedef struct tagcomplex {
    float thuc, ao;
} complex;

complex tong(complex a, complex
{
    complex c;
    c.thuc = a.thuc + b.thuc;
    c.ao = a.ao + b.ao;
    return c;
}

complex hieu(complex a, complex
{
    complex c;
    c.thuc = a.thuc - b.thuc;
    c.ao = a.ao - b.ao;
    return c;
}

complex tich(complex a, complex
{
    complex c;
    c.thuc = a.thuc*b.thuc - a.ao*b.ao;
    c.ao = a.thuc*b.ao + a.ao*b.thuc;
    return c;
}

complex thuong(complex a, complex
{
    complex c;
    float tongbp;
    tongbp = b.thuc*b.thuc + b.ao*b.ao;
    c.thuc = (a.thuc*a.ao + b.thuc*b.ao)/tongbp;
    c.ao = (a.ao*b.thuc - a.thuc*b.ao)/tongbp;
    return c;
}

float argument(complex a)
{
    return acos(a.thuc/sqrt(a.thuc*a.thuc + a.ao*a.ao));
}

float modul(complex a)
```

```

{
return sqrt(a.thuc*a.thuc + a.ao*a.ao);
}

void print_complex(complex a)
{
printf("%.2f + %.2fi", a.thuc, a.ao);
}

void main()
{
complex a, b, c;
printf("\nNhap he so thuc va phuc cua A : ");
scanf("%f%f", &a.thuc, &a.ao);
printf("\nNhap he so thuc va phuc cua B : ");
scanf("%f%f", &b.thuc, &b.ao);
printf("\nSo phuc A = ");
print_complex(a);
printf("\nSo phuc B = ");
print_complex( ;
printf("\nTong cua chung = ");
c = tong(a, ;
print_complex©;
printf("\nHieu cua chung = ");
c = hieu(a, ;
print_complex©;
printf("\nTich cua chung = ");
c = tich(a, ;
print_complex©;
printf("\nThuong cua chung = ");
c = thuong(a, ;
print_complex©;
printf("\nArgument cua a = %f", argument(a));
printf("\nModul cua a = %f", modul(a));
getch();
}

```

DÃY TĂNG DẦN

```
#include <stdio.h>

void main()
{
    int a[10], i, maxstart, maxend, maxlen, tmpstart, tmpend,
    tmpflen;

    printf("\nNhap vao 10 phan tu nguyen cua day :");
    for (i=0; i<10; i++)
        scanf("%d", &a[i]);
    printf("Day da cho :\n");
    for (i=0; i<10; i++)
        printf("%6d", a[i]);

    maxstart = maxend = tmpstart = tmpend = 0;
    maxlen = tmpflen = 1;
    for (i=1; i< 10; i++)
    {
        if (a[i] < a[tmpend])
        {
            if (maxlen < tmpflen)
            {
                maxstart = tmpstart;
                maxend = tmpend;
                maxlen = tmpflen;
            }
            tmpstart = tmpend = i;
            tmpflen = 1;
        }
        else
        {
            tmpflen++;
            tmpend++;
        }
    }
    if (maxlen < tmpflen)
    {
        maxstart = tmpstart;
        maxend = tmpend;
    }
    printf("\nDay tang co so phan tu nhieu nhat la : \n");
    for (i=maxstart; i<=maxend; i++)
        printf("%6d", a[i]);
    getch();
}
```

DÃY TĂNG CÓ TỔNG DÀI NHẤT `#include <stdio.h>`

```
void main()
{
    int a[10], i, maxstart, maxend, maxtotal, tmpstart, tmpend,
    tmptotal;

    printf("\nNhap vao 10 phan tu nguyen cua day :");
    for (i=0; i<10; i++)
        scanf("%d", &a[i]);
    printf("Day da cho :\n");
    for (i=0; i<10; i++)
        printf("%6d", a[i]);

    maxstart = maxend = tmpstart = tmpend = 0;
    maxtotal = tmptotal = a[0];
    for (i=1; i< 10; i++)
    {
        if (a[i] < a[tmpend])
        {
            if (maxtotal < tmptotal)
            {
                maxstart = tmpstart;
                maxend = tmpend;
                maxtotal = tmptotal;
            }
            tmpstart = tmpend = i;
            tmptotal = a[i];
        }
        else
        {
            tmptotal += a[i];
            tmpend++;
        }
        if (maxtotal < tmptotal)
        {
            maxstart = tmpstart;
            maxend = tmpend;
        }
    }
    printf("\nDay tang co tong nhieu nhat la : \n");
    for (i=maxstart; i<=maxend; i++)
        printf("%6d", a[i]);
    getch();
}
```

QUẢN LÝ SINH VIÊN

```
#include <stdio.h>
#include <ctype.h>
#include <mem.h>
#include <string.h>

#define MAX 100
#define TOAN 0
#define LY 1
#define HOA 2

struct sinhvien {
    char mslop[5];
    char hoten[35];
    float diem[3];
} danhsach[MAX];
int n = 0;

void nhapmoi()
{
    char mslop[5], tmp[3];
    int i;
    float diem[3];
    do {
        printf("\nCho biet ma so lop : ");
        gets(mslop);
        if (strlen(mslop))
        {
            strcpy(danhsach[n].mslop, mslop);
            printf("\nCho biet ho ten : ");
            gets(danhsach[n].hoten);
            printf("\nCho biet diem so : ");
            for (i=0; i<3; i++)
            {
                scanf("%f", &diem[i]);
                danhsach[n].diem[i] = diem[i];
            }
            gets(tmp);
            n++;
        }
    } while (strlen(mslop));
}

void timkiem()
{
    char mslop[5];
    int i = 0, found = 0;
```



```

printf("\nCho biet ma so lop : ");
gets(mslop);
if (strlen(mslop))
    while (i<n)
        if (strcmp(danh sach[i].mslop, mslop) == 0)
        {
            printf("\nMa so lop : %s", danh sach[i].mslop);
            printf("\nHo va ten : %s", danh sach[i].hoten);
            printf("\nDiem Toan : %f",
danh sach[i].diem[TOAN]);
            printf("\nDiem Ly : %f", danh sach[i].diem[LY]);
            printf("\nDiem Hoa : %f", danh sach[i].diem[HOA]);
            found = 1;
            break;
        }
        else
            i++;
if (!found)
    printf("\nKhong tim thay!!!");
}

void xoa()
{
    char mslop[5], traloi;
    int i = 0, j;
    printf("\nCho biet ma so lop : ");
    gets(mslop);
    if (strlen(mslop))
        while (i<n)
            if (strcmp(danh sach[i].mslop, mslop) == 0)
            {
                printf("\nMa so lop : %s", danh sach[i].mslop);
                printf("\nHo va ten : %s", danh sach[i].hoten);
                printf("\nDiem Toan : %f",
danh sach[i].diem[TOAN]);
                printf("\nDiem Ly : %f", danh sach[i].diem[LY]);
                printf("\nDiem Hoa : %f", danh sach[i].diem[HOA]);
                printf("\nCo muon xoa khong (C/K)? ");
                do {
                    traloi = toupper(getch());
                } while (traloi != 'C' && traloi != 'K');
                putchar(traloi, stdout);
                if (traloi == 'C')
                {
                    n--;
                    memcpy(&danh sach[i], &danh sach[i+1],
sizeof(struct sinh vien) * (n-i));

```

```

        break;
    }
}
else
    i++;
}

void menu()
{
    printf("\n*****");
    printf("\n* 1. Them    *");
    printf("\n* 2. Xoa     *");
    printf("\n* 3. Tim kiem *");
    printf("\n* 0. Thoat    *");
    printf("\n*****");
    printf("\nChon lua ? ");
}

void main()
{
    char traloi;
    do {
        menu();
        do {
            traloi = getch();
        } while (traloi < '0' || traloi > '3');
        putchar(traloi, stdout);
        switch (traloi)
        {
            case '1' : nhapmoi();
                       break;
            case '2' : xoa();
                       break;
            case '3' : timkiem();
                       break;
        }
    } while (traloi != '0');
}

```

GIẢI PHƯƠNG TRÌNH BẬC HAI

```

#include <stdio.h>
#include <math.h>
void main()
{
    float a, b, c, delta;

    printf("\nGiai phuong trinh bac hai  $AX^2 + BX + C = 0$ ");
    printf("\nCho biet ba he so A B C : ");
}

```

```

scanf("%f%f%f", &a, &b, &c);

delta = b * b - 4 * a * c;
if (delta<0)
printf("Phuong trinh vo nghiem");
else if (delta == 0)
printf("Phuong trinh co nghiem kep x1 = x2 = %f", -
b/(2*a));
else
{
printf("Phuong trinh co hai nghiem phan biet\nx1 = %f", (-b
+ sqrt(delta))/(2*a));
printf("\nx2 = %f", (-b - sqrt(delta))/(2*a));
}
getch();
}

```

MA PHƯƠNG

```

#include <stdio.h>
#include <conio.h>

// func declaration
void matrix( int n );

// main()
int main(void)
{
    int n;

    // input until it's valid.
    do
    {
        printf("\n Plz input size of matrix [ odd size & n <
20 ]: n = ");
        scanf("%d",&n);
        if ( n % 2 == 0 ) printf("\n Invalid input value .. Plz
re-input ... \n");
    }
    while ( n % 2 == 0 );

    if ( n > 20 ) { n = 19 ; // in case of n is greater
than 20
        printf("\n %d is greater than 20 & set to be default as
19 .",n ); } // end if

    // call matrix()
    matrix(n);
    // stop to watch
}

```

```

    getch();
    return 0;
}

// function matrix(int n)
void matrix( int n )
{
    int a[20][20];
    int i, j, row, col, count = 1;
    int old_row, old_col, sum = 0;

    // set starting value of array
    for ( i = 0 ; i < n ; i++ )
    for ( j = 0 ; j < n ; j++ )
        a[i][j] = 0;

    // set the 1st value to start
    row = 0; col = (n-1) / 2;

    while ( count < n*n + 1 )
    {
        a[row][col] = count++ ; // set value for
elements
        old_row = row ; old_col = col; // save the last
addresses
        // define whether going out of array
        row -- 1; if ( row == -1 ) row = n - 1;
        col += 1; if ( col == n ) col = 0;
        // in case of already having number
        if ( a[row][col] != 0 )
        {
            row = old_row + 1;
            col = old_col;
        } // end if
    } // end while
    // print result
    printf("\n");
    for ( i = 0 ; i < n ; i++ )
    {
        for ( j = 0 ; j < n ; j++ )
            printf("%4d",a[i][j]);
        printf("\n");
    } // end for

    // calculate sum
    for ( j = 0 ; j < n ; j++ )
        sum += a[0][j];
}

```

```
printf("\n Sum of each row - column - diagonal line  
is : %d " , sum);
```

```
return;
```

```
}
```

FILE VÀ HỆ THỐNG

1. Xóa 1 file dùng Remove

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
remove("d:/urls1.dat");
```

```
return 0;
```

```
}
```

2. Xóa 1 File dùng Unlink

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
remove("C:/pete.txt");
```

```
return 0;
```

```
}
```

3. Cho biết thông tin FAT#include <stdio.h>

```
#include <dos.h>
```

```
void main(void)
```

```
{
```

```
struct fatinfo fat;
```

```
getfatd(&fat);
```

```
printf("Sectors per cluster %d\n", fat.fi_sclus);
```

```
printf("Clusters per disk %u\n", fat.fi_nclus);
```

```
printf("Bytes per cluster %d\n", fat.fi_bysec);
```

```
printf("Disk type %x\n", fat.fi_fatid & 0xFF);
```

```
}
```

4. Đếm tần suất 1 kí tự trong 1 file

```
# include <stdio.h>
```

```
# include <string.h>
```

```
main()
```

```
{
```

```
FILE *fp;
```

```
char in[100];
```

```
long int freq[257];
```

```
int i;
```

```

printf("\nFile frequency table generator\n\n");

printf("\nInput file:");
scanf("%s", in);
fp=fopen(in, "rb");
if(fp==NULL)
{
    printf("\nCould not open input file.Aborting\n");
    return 1;
}
for(i=0;i<257;i++)
    freq[i]=0;
while(i=fgetc(fp), i!=EOF)
{
    freq[i]++;
}
fcloseall();
fp=fopen("count.txt", "w");
fprintf(fp, "\nCharacter frequency table of %s\n", in);
fprintf(fp, "\nCharacter ASCII frequency\n\n");
for(i=0;i<256;i++)
{
    if(i==26)
    {
        fprintf(fp, "\t    26\t    %ld\n", freq[26]);
    }
    else if(i==9)
    {
        fprintf(fp, "\t    9\t    %ld", freq[9]);
    }
    else if(i<10)
    {
        fprintf(fp, "%c\t    %d\t    %ld\n", i, i, freq[i]);
    }
    else if(i<100)
    {
        fprintf(fp, "%c\t    %d\t    %ld\n", i, i, freq[i]);
    }
    else
    {
        fprintf(fp, "%c\t    %d\t    %ld\n", i, i, freq[i]);
    }
}

fcloseall();

```

```

    printf("\nFrequency table copied to count.txt\n");
}

```

5. Đọc nội dung 1 file

```
#include <stdio.h>
```

```
void main(void)
```

```
{
```

```
FILE *fp;
```

```
char ch;
```

```
fp = fopen("websites.txt", "r");
```

```
ch = getc(fp);
```

```
while (ch != EOF)
```

```
{
```

```
    putchar(ch);
```

```
    ch = getc(fp);
```

```
}
```

```
printf("\n\n");
```

```
}
```

6. Chọn ổ đĩa trong DOS

```
#include <stdio.h>
```

```
#include <dir.h>
```

```
void main(void)
```

```
{
```

```
int drives;
```

```
drives = setdisk(3);
```

```
printf("The number of available drives is %d\n", drives);
```

```
}
```

7. Chọn ổ đĩa trong WINS

```
#include <windows.h>
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
void main(void)
```

```
{
```

```
    char szBuffer[MAX_PATH+100];
```

```
    UINT nDrive, AvailDrive = 0;
```

```
    int dwLogicalDrives = GetLogicalDrives();
```

```
    DWORD Success;
```

```
    printf("Number of logical drives: %d\n",
dwLogicalDrives);
```

```
        for (nDrive = 0; nDrive < 32; nDrive++)
```

```
        {
```

```
            if (dwLogicalDrives & (1 << nDrive))
```

```
                { // Is drive available?
```

```

        AvailDrive++;
        // Get disk information.
        wsprintf(szBuffer, "%c:\\", nDrive+'A', '\\0');
        // Print out information.
        if(SetCurrentDirectory(szBuffer))
            printf("%s Is Now Current\n", szBuffer);
        else
            printf("Could not set %s as the current
drive\n", szBuffer);
    }
}
    printf("Number of drives available: %d\n",
AvailDrive);

}

```

8. Cho biết kích thước 1 file

```

#include <stdio.h>
#include <io.h>
#include <fcntl.h>
#include <sys\stat.h>

int main()
{
    int fp;

    long file_size;

    if ((fp = open("f:/cprojects/urls.txt", O_RDONLY)) == -1)
        printf("Error opening the file \n");
    else
    {
        file_size = filelength(file_handle);
        printf("The file size in bytes is %ld\n", file_size);
        close(fp);
    }
    return 0;
}

```

Tính thứ của ngày

```

#include <stdio.h>
#include <conio.h>

struct date
{
    int    month;
    int    day;
    int    year;
} date_1;

```



```

long int funct1 (int y,int m)
{
    long int result;
    if ( m <= 2 )
        y -= 1;
    result = y;
    return (result);
}

long int funct2 (int m)
{
    long int result;
    if ( m <= 2 )
        result = m + 13;
    else
        result = m + 1;
    return(result);
}

long int day_count (int m, int d, int y)
{
    long int number;
    number = 1461 * funct1(y,m) / 4 + 153 * funct2(m) / 5 + d;

    return (number);
}

void main ()
{
    long int number_of_days1;
    int day_of_week;

    printf ("Nhap vao mot ngay (dd mm yyyy), vd 12 03 1999 \n");
    scanf ("%d %d %d", &date_1.day, &date_1.month, &date_1.year);
    number_of_days1 = day_count (date_1.month, date_1.day,
date_1.year);
    printf ("\nNgay la : " );

    day_of_week = (number_of_days1 - 621049) % 7;
    switch (day_of_week)
    {
        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();
}

```

Giải hệ phương trình bậc nhất.

```

#include <stdio.h>
#include <conio.h>

void main()
{
    int a, b, c, d, e, f, dthuc;
    float x, y;

    printf("\nNhap vao cac he so a,b,c,d,e,f : ");
    scanf("%d%d%d%d%d%d", &a, &b, &c, &d, &e, &f);
    dthuc = b*d - e*a;
    if (dthuc != 0)
    {
        y = (float) (c*d-a*f)/dthuc;
        x = (float) (b*f-c*e)/dthuc;
        printf("Nghiem x = %f, y = %f", x, y);
    }
    else
        printf("\nHe phuong trinh vo ngiem.");

    getch();
}

```

Nhập chuỗi và in chuỗi

```

#include <stdio.h>
#include <conio.h>

void main()
{
    char name[80];

    printf("\nXin cho biet ten cua ban : ");
    gets(name);

    printf("Chao %s\n", name);
    getch();
}

```

In ra bảng cửu chương

```

#include <stdio.h>
#include <conio.h>
#include <string.h>

void main()

```

```

{
  int i, j;
  char chuoi[] = "B A N G   C U U   C H U O N G";
  char ten[10][5] = {"", "", "Hai", "Ba", "Bon", "Nam",
                   "Sau", "Bay", "Tam", "Chin"};

  clrscr();
  textcolor(YELLOW);
  gotoxy((80 - strlen(chuoi)) / 2, 1);
  cprintf("%s\n\n", chuoi);

  for (i=2; i<=9; i++)
  {
    gotoxy(10*(i-2) + (10 - strlen(ten[i]))/2, 4);
    textcolor(i);
    cprintf("%s", ten[i]);
  }
  for (j=1; j<=10; j++)
    for (i=2; i<=9; i++)
    {
      gotoxy(10*(i-2) + 1, j+4);
      textcolor(i);
      cprintf("%dx%2d = %2d", i, j, i*j);
    }
  getch();
}

```

In ra năm âm lịch tương ứng với năm nhập vào.

```

#include <stdio.h>
#include <conio.h>

void main()
{
  unsigned nam;
  char can[][5] = {"Giap", "At", "Binh", "Dinh", "Mau", "Ky",
                 "Canh", "Tan", "Nham", "Quy"};
  char chi[][5] = {"Ty", "Suu", "Dan", "Meo", "Thin", "Ty",
                 "Ngo", "Mao", "Than", "Dau", "Tuat", "Hoi"};

  printf("\nNhap nam can biet : ");
  scanf("%d", &nam);
  printf("Nam am lich cua %d la %s %s", nam, can[(nam+6)%10],
        chi[(nam+8)%12]);
  getch();
}

```

Hiện bảng mã ASCII

```

#include <stdio.h>
#include <conio.h>

void main()
{
  int i, j;

  clrscr();
  printf("   ");

```

```

for (j=0; j<16; j++)
    printf("%3d", j);
for (i=2; i<16; i++)
    for (j=0; j<16; j++)
    {
        if (j == 0)
            printf("\n%2d ", i);
        printf("  %c", i*16+j);
    }
getch();
}

```

Chứng minh đẳng thức An Casi

```

#include <stdio.h>

unsigned long vetrai(unsigned long n)
{
    unsigned long tmp=0, i;
    for (i=1; i<=n; i++)
        tmp += i*i*i*i;
    return tmp;
}

unsigned long vephai(unsigned long n)
{
    unsigned long tmp;
    tmp = (long)6*n*n*n*n*n + (long)15*n*n*n*n + (long)10*n*n*n*n - n;
    tmp = tmp / (long)30 ;
    return tmp;
}

void main()
{
    unsigned long tong1, tong2, n;

    for (n=1; n<=50; n++)
    {
        tong1 = vetrai(n);
        tong2 = vephai(n);
        if (tong1 == tong2)
        {
            printf("\nSo %d thoa man dang thuc An Casi. ", n);
            printf("Tong1 = %ld - Tong2 = %ld", tong1, tong2);
        }
        else
        {
            printf("\nSo %d khong thoa man dang thuc An Casi. ", n);
            printf("Tong1 = %ld - Tong2 = %ld", tong1, tong2);
        }
    }
    getch();
}

```

Bài toán Ancarokhi

```

#include <stdio.h>

```

```

void main()
{
    int dai, rong;

    printf("\nBai toan Ancarokhi : Tim dien tich hinh chu nhat co chieu
dai gap hai");
    printf("\nchieu rong va dien tich = chu vi");
    for (dai = 1; dai < 100; dai ++)
        for (rong=1; rong < 100; rong++)
            if (dai == 2 * rong && (dai + rong)*2 == dai*rong)
                printf("\nDai = %d; Rong = %d", dai, rong);
    getch();
}

```

Đếm số lần xuất hiện của các ký tự trong chuỗi

```

#include <stdio.h>
#include <ctype.h>

void main()
{
    char chuoi[80];
    int i = 0, count = 0;

    printf("\nNhap vao mot chuoi bat ky : ");
    gets(chuoi);

    while (chuoi[i] != 0)
    {
        if (isalpha(chuoi[i++]))
            count++;
    }

    printf("So ky tu trong chuoi = %d", count);
    getch();
}

```

Ví dụ về quá tải toán tử

```
#include <iostream.h>
#include <conio.h>
#include <math.h>
class PS
{
public:
    long tu, mau;

    PS()
    {
        tu=0;
        mau=0;
    }
    ~PS(){};
    int uscln(long a, long b);
    void rutgon();
    void nhap();
    void xuat();
    PS operator+(PS &a);
    PS operator-(PS &a);
    PS operator*(PS &a);
    PS operator/(PS &a);
};

int PS::uscln(long a, long b)
{
    if(a!=0 && b!=0)
    {
        a=abs(a);
        b=abs(b);
        while(a!=b)
        {
            if(a>b)
                a=a-b;
            else
                b=b-a;
        }
        return a;
    }
    else
        return 1;
}

void PS::rutgon()
{
    int u;
    u=uscln(tu, mau);
    tu=tu/u;
    mau=mau/u;
}

void PS::nhap()
{
    Nhap:
    cout<<"Nhập tu số ";
```

```

    cin>>tu;
    cout<<"Nhap mau so ";
    cin>>mau;
    if(mau==0)
    {
        cout<<"Mau phai khac 0"<<endl;
        goto Nhap;
    }
}

void PS::xuat()
{
    rutgon();
    if(mau<0)
    {mau=-mau; tu=-tu;}
    if(tu==0)
        cout<<"0"<<endl;
    else
        if(mau==1)
            cout<<tu<<endl;
        else
            cout<<tu<<"/"<<mau<<endl;
}

PS PS::operator+(PS &a)
{
    a.tu=tu*a.mau+mau*a.tu;
    a.mau=mau*a.mau;
    return a;
}

PS PS::operator-(PS &a)
{
    a.tu=tu*a.mau-mau*a.tu;
    a.mau=mau*a.mau;
    return a;
}

PS PS::operator*(PS &a)
{
    a.tu=tu*a.tu;
    a.mau=mau*a.mau;
    return a;
}

PS PS::operator/(PS &a)
{
    a.tu=tu*a.mau;
    a.mau=mau*a.tu;
    return a;
}

```

Một ví dụ về sử dụng template và quá tải toán tử Nhập xuất

```
#include <iostream.h>
#include <stdlib.h>
#include <conio.h>
class sv
{
    private :
        char ten[100];
        float Diem;
    public:
        sv()
        {
            Diem=0;
        }
        sv(char a[],float D)
        {
            strcpy(ten,a);
            Diem=D;
        }
        sv(sv&a)
        {
            Diem = a.Diem;
            strcpy(ten,a.ten);
        }
        void set_sv(char a[],float D)
        {
            strcpy(ten,a);
            Diem=D;
        }
        float get_diem() const
        {
            return Diem;
        }
        char* get_ten()
        {
            return ten;
        }
        friend ostream&operator <<(ostream&out,sv&);
        friend istream&operator>>(istream&in,sv&);
        operator float()
        {
            return float(Diem);
        }
};

ostream&operator <<(ostream&out,sv&a)
{
    cout<<"\n\n\t\t\tTen " <<a.ten<<endl;
    cout<<"\t\t\tDiem " <<a.Diem<<endl;
}

istream&operator>>(istream&in,sv&a)
{
    cout<<"\t\t\tNhập ten ";
    cin.ignore();
    cin.getline(a.ten,50);
}
```



```

        cout<<"\t\t\tNhap diem ";
        cin>>a.Diem;
    }
    int ucln(int a,int b)
    {
        int r;
        while(b)
        {
            r = a%b;
            a = b;
            b=r;
        }
        return a;
    }
class phanso
{
    private:
        float tu,mau;
    public:
        phanso(float a=1,float b=1)
        {
            if(b)
            {
                tu = a;
                mau = b;
            }
            else
            {
                tu =1;
                mau=1;
            }
        }
        void set_phanso(float a,float b)
        {
            tu =a;
            mau = b;
        }
        void nhap()
        {
            cout<<"\t\t\tNhap du lieu cho phan so "<<endl;
            cout<<"\t\t\tTu ";
            cin>>tu;
            cout<<"\t\t\tMau ";
            cin>>mau;
            toigian();
        }
        void toigian()
        {
            int t=ucln(tu,mau);
            tu = tu/t;
            mau = mau/t;
        }
        operator float()
        {
            return float(tu/mau);
        }

```

```

        }
        friend ostream&operator <<(ostream&out,phanso&a);
        friend istream&operator >>(istream&in,phanso&a);
};
ostream&operator<<(ostream&out,phanso&a)
{
    out<<a.tu<<"/"<<a.mau<<"->";
}
istream&operator >>(istream&in,phanso&a)
{
    cout<<"\t\tTu ";
    cin>>a.tu;
    cout<<"\t\tMau ";
    cin>>a.mau;
}
template <class T,int n>
class set
{
    private:
        T data[n];
        int spt;
    public:
        set()
        {
            spt=0;
        }
        set(const set&a)
        {
            for(int i=0;i<a.spt;i++)
                data[i]=a.data[i];
            spt = a.spt;
        }
        void them(T&a);
        bool search(T&a);
        friend ostream& operator<<(ostream&out,set<T,n>&a);
        friend set operator +(set&a,set&b);
        friend set operator *(set&a,set&b);
        friend set operator -(set&a,set&b);
        set operator =(const set&b)
        {
            for(int i=0;i<b.spt;i++)
                data[i]=b.data[i];
            spt=b.spt;
            return (*this);
        }
};
template <class T,int n>
void set<T,n>::them(T&a)
{
    if(spt<n)
        data[spt++]=a;
    else
        cout<<"\t\tMang da day rui khong them duoc nua dau "<<endl;
}
template <class T,int n>

```

```

bool set<T,n>::search(T&a)
{
    for(int i=0;i<spt;i++)
        if(data[i]==a)
            return true;
        return false;
}
template <class T,int n>
ostream&operator<<(ostream&out, set<T,n>&a)
{
    if(a.spt==0)
        out<<" rong "<<endl;
    for(int i=0;i<a.spt;i++)
    {
        out<<a.data[i];
        if(i<a.spt-1)
            cout<<"->";
    }
}
template <class T,int n>
set<T,n> operator +(set<T,n>&a, set<T,n>&b)
{
    set<T,n> r(a);

    for(int i=0;i<b.spt;i++)
        if(!a.search(b.data[i]))
            r.them(b.data[i]);

    return r;
}
template <class T,int n>
set<T,n> operator -(set<T,n>&a, set<T,n>&b)
{
    set<T,n> r;
    for(int i=0;i<a.spt;i++)
        if(!b.search(a.data[i]))
            r.them(a.data[i]);
    return r;
}
template <class T,int n>
set<T,n> operator *(set<T,n>&a, set<T,n>&b)
{
    set<T,n> r;

    for(int i=0;i<a.spt;i++)
        if(b.search(a.data[i]))

            r.them(a.data[i]);

    return r;
}
void main()
{
    set<float,100> a;
    set<float,100> c;
}

```

```

set<float,100> d;
set<float,100> e;
set<float,100> f;
set<sv,100> g;
set<phanso,100> b;
int n,m,l;
float r;
sv A;
phanso s;
cout<<"\t\t\tNhap so luong cac so thu ";
cin>>n;
for(int i=0;i<n;i++)
{
    cout<<" nhap so thu "<<(i+1)<<":";
    cin>>r;
    a.them(r);
}clrscr();
cout<<"\t\t\tNhap so luong phan so ";
cin>>m;
for(int i=0;i<m;i++)
{
    cout<<"\t\t\tNhap phan so thu "<<(i+1)<<endl;
    cin>>s;
    b.them(s);
    c.them(s);clrscr();
}

clrscr();
cout<<"\t\t\tNhap so luong cac sinh vien ";
cin>>l;
for(int i=0;i<l;i++)
{
    cout<<"\t\t\tNhap du lieu cho sinh vien thu "<<(i+1)<<endl;
    cin>>A;
    g.them(A);
    clrscr();
}
clrscr();
textcolor(YELLOW+RED);
cprintf("%s","\t\t\tchuong trinh da gan cac so 1 cach tu dong ta duoc ");
cout<<"\n\nday so thuc vua nhap "<<endl;
cout<<a;
cout<<"\n\nday phan so vua nhap "<<endl;
    cout<<b;
cout<<"\n\tDay sinh vien vua nhap "<<endl;
cout<<g;
getch();clrscr();
d = a+c;
cout<<"\n\n hop cua hai tap hop phan so va so thuc la "<<endl;;
cout<<d;
e=a*c;
cout<<"\n\n giao cua hai tap so thuc va phan so la "<<endl;
cout<<e;
cout<<"\n\nhieu cua hai tap so thuc va phan so la "<<endl;
f=a-c;
cout<<f;

```

```

    getch();
}

```

Tổng hai ma trận

```

#include <iostream.h>
#include <conio.h>
#include <stdlib.h>
void congmt(float a[][10],float b[][10],float c[][10],int hang,int cot)
;
void nhapmt(float a[][10],int hang,int cot);
void inmt(float a[][10],int hang,int cot);
void main()
{
    system("color 3e");
    float a[10][10],b[10][10],c[10][10];
    int hang1,cot1;

    cout<<"Moi ban nhap vao ma tran a: \n";
    cout<<"Nhap vao so hang cua ma tran a: ";
    cin>>hang1;
    cout<<"Nhap vao so cot cua ma tran a: ";
    cin>>cot1;
    nhapmt(a,hang1,cot1);
    inmt(a,hang1,cot1);

    int hang2,cot2;
    cout<<"Moi ban nhap vao ma tran b: \n";
    do
    {
        cout<<"Nhap vao so hang cua ma tran b: ";
        cin>>hang2;
    }while(hang2 != hang1);
    do
    {
        cout<<"Nhap vao so cot cua ma tran b: ";
        cin>>cot2;
    }while(cot2 != cot1);
    nhapmt(b,hang2,cot2);
    inmt(b,hang2,cot2);

    cout<<"\nVay tong cua hai ma tran a,b la: \n";
    congmt(a,b,c,hang1,cot1);
    inmt(c,hang1,cot1);
    getch();
}

void congmt(float a[][10],float b[][10],float c[][10],int hang,int cot)
{
    for (int i=0; i<hang; i++)
        for (int j=0; j<cot; j++)
            c[i][j] = a[i][j] + b[i][j];
}

void nhapmt(float a[][10],int hang,int cot)
{
    for(int i = 0;i < hang;i++)

```

```

    {
        for(int j = 0; j < cot; j++)
        {
            cout<<"Nhap vao phan tu ["<<i<<";"<<j<<"]": ";
            cin>>a[i][j];
        }
    }
}

```

```

void inmt(float a[][10],int hang,int cot)
{
    for(int i = 0; i < hang; i++)
    {
        for(int j = 0; j < cot; j++)
        {
            cout<<a[i][j]<<"\t";
        }
        cout<<endl;
    }
}

```

Tiếp một ví dụ về Đa hình

```

#include <iostream.h>
#include <conio.h>
#include <math.h>

```

```

class Point
{
    private:
        int x; int y;
    public:
        Point()
        {
            x = 0; y = 0;
        }
        Point(int a,int b)
        {
            x = a; y = b;
        }
        virtual void set(int a,int b)
        {
            x = a; y = b;
        }
        float gettung()
        {
            return y;
        }
        float gethoanh()
        {
            return x;
        }
        float kc(Point t)
        {
            return ((x - t.x)*(x - t.x) + (y - t.y)*(y - t.y));
        }
        virtual void in()
        {

```

```

        cout<<"\n(" <<x<<" "; <<y<<" ) ";
    }
};
class Cpoint : protected Point
{
    private:
        int mau;
    public:
        Cpoint() : Point()
        {
            mau = 0;
        }
        Cpoint(int a,int b,int mau_) : Point(a,b)
        {
            mau = mau_;
        }
        void set(int a,int b,int mau_)
        {
            Point::set(a,b);
            mau = mau_;
        }
        void in()
        {
            Point::in();
            cout<<" Co mau la " <<mau;
        }
};

void main()
{
    Point *p;
    p = new Cpoint(3,5,6);
    p->in();
    delete p;
    p = new Point(5,2);
    p->in();
    getch();
}

```

Một ví dụ về Đa hình

```

#include <iostream.h>
#include <conio.h>
#include <math.h>
class hinhve
{
    public:
        virtual float dientich() = 0;
        virtual char *ten() = 0;
        virtual void in()=0;
};

class haichieu : public hinhve
{
    public:
        virtual float chuvi() = 0;
        void in()

```

```

        {
            cout<<"ten cua hinh: "<<ten()
                <<" ,dien tich la: "<<dientich()
                <<" ,chu vi la: "<<chuvi()<<endl;
        }
};

class bachieu : public hinhve
{
public:
    virtual float thetich() = 0;
    void in()
    {
        cout<<"ten cua hinh: "<<ten()
            <<" ,dien tich la: "<<dientich()
            <<" ,the tich la: "<<thetich()<<endl;
    }
};

class hinhtron : public haichieu
{
private:
    float r;
public:
    hinhtron() { r = 0;}
    hinhtron(float bk) {r = bk;}
    float chuvi()
    {
        return 2*3.14*r;
    }
    float dientich()
    {
        return 3.14*r*r;
    }
    char *ten()
    {
        return "Hinh Tron";
    }
};

class hinhvuong : public haichieu
{
private:
    float a;
public:
    hinhvuong(float x)
    {
        a = x;
    }
    float chuvi()
    {
        return a*4;
    }
    float dientich()

```



```

        {
            return a*a;
        }
        char *ten()
        {
            return "Hinh Vuong";
        }
};

class tgdeu : public haichieu
{
    private:
        float a;
    public:
        tgdeu(float x) : a(x){}
        float chuvi()
        {
            return 3*a;
        }
        float dientich()
        {
            return a*a*sqrt(3)/2;
        }
        char *ten()
        {
            return "Hinh tam giac deu";
        }
};

class cau: public bachieu
{
    private:
        float r;
    public:
        cau(float bk): r(bk){}
        float thetich() { return r*r*r*3.14;}
        float dientich() { return 4*3.14*r*r; }
        char *ten()
        {
            return "Hinh Cau";
        }
};

class lapphuong : public bachieu
{
    private:
        float a;
    public:
        lapphuong(float x) : a(x) {}
        float thetich() { return a*a*a; }
        float dientich() { return 6*a*a; }
        char * ten() { return "Hinh Lap Phuong"; }
};

void main()
{
    hinhve *p;
    p = new hinhtron(3);
}

```

```

    p->in();
    delete p;
    p = new lapphuong(3);
    p -> in();
    delete p;
    p = new cau(3);
    p -> in();
    delete p;
    p = new tgdeu(5);
    p -> in();
    delete p;
    p = new hinhvuong(6);
    p -> in();
    getch();
}

```

SẮP XẾP MẢNG

```

#include<alloc.h>
#include<stdio.h>
#include<conio.h>
//=====
void taolap(int *A,int n)
{
    int i;
    printf("\n Tao lap day so:\n");
    for(i=0;i<n;i++)
    {
        printf("\n A[%d]=",i);
        scanf("%d",&A[i]);
    }
}
void dayso(int *A,int n)
{
    int i;
    for(i=0;i<n;i++)
        printf("%5d",A[i]);
}
void select(int *A,int n)
{
    int i,j,temp;
    for(i=0;i<n-1;i++)
    {
        for(j=i+1;j<n;j++)
        {
            if(A[i]>A[j])
            {
                temp=A[i];
                A[i]=A[j];
                A[j]=temp;
            }
        }
    }
    printf("\n Ket qua thu duoc la:");
    dayso(A,n);
}
void luachon()

```

```

{
    clrscr();
    int *A,n;
    printf("\n \t SAP XEP KIEU LUA CHON\n");
    printf("\n Nhap so phan tu n=");
    scanf("%d",&n);
    A=(int*)malloc(n*sizeof(int));
    taolap(A,n);
    select(A,n);
    free(A);
    getch();
}

//=====
void in2(int *A,int n)
{
    int i;
    for(i=0;i<n;i++)
        printf("%5d",A[i]);
}
void tructiep()
{
    clrscr();
    int *A,i,j,n,temp;
    printf("\n SAP XEP KIEU TRUC TIEP\n");
    printf("\n\t SAP XEP KIEU TRUC TIEP\n");
    printf("\n Nhap so phan tu n=");
    scanf("%d",&n);
    A=(int*)malloc(n*sizeof(int));
    printf("\n Tao lap day so:\n");
    for(i=0;i<n;i++)
    {
        printf("\n A[%d]=",i);
        scanf("%d",&A[i]);
    }
    for(i=1;i<n;i++)
    {
        temp=A[i];
        for(j=i-1;j>=0&&temp<A[j];j--)
            A[j+1]=A[j];
        A[j+1]=temp;
        printf("\n\nKet qua lan thu %d:",i);
        in2(A,i+1);
    }
    free(A);
    getch();
}
//=====
void tlap(int *A,int n)
{
    int i;
    printf("\n");
    printf("\n Tao lap day so:\n");
    for(i=0;i<n;i++)
    {

```

```

        printf("\n A[%d]=", i);
        scanf("%d", &A[i]);
    }
}
void in1(int *A, int n)
{
    int i;
    for(i=0; i<n; i++)
        printf("%5d", A[i]);
}

void bubble(int *A, int n)
{
    int i, j, temp;
    for(i=1; i<n; i++)
    {
        for(j=n-1; j>=i; j--)
        {
            if(A[j-1]>A[j])
            {
                temp=A[j-1];
                A[j-1]=A[j];
                A[j]=temp;
            }
        }
        printf("\n\n Ket qua lan %d:", i);
        in1(A, n);
    }
}

void suibot()
{
    clrscr();
    int *A, n;
    printf("\n SAP XEP KIEU SUI BOT\n");
    printf("\n Nhap so phan tu n=");
    scanf("%d", &n);
    A=(int*)malloc(n*sizeof(int));
    tlap(A, n);
    bubble(A, n);
    free(A);
    getch();
}

//=====
void qs(int *A, int left, int right)
{
    int i, j, x, y;
    i=left;
    j=right;
    x=A[(left+right)/2];
    do
    {
        while(A[i]<x&& i<right) i++;
        while(A[j]>x&& j>left) j--;
        if(i<=j)
        {
            y=A[i];

```

```

        A[i]=A[j];
        A[j]=y;
        i++;
        j--;
    }
    }while(i<=j);
    if(left<j)qs(A,left,j);
    if(i<right)qs(A,i,right);
}
void quick(int *A,int n)
{
    qs(A,0,n-1);
}

void in3(int *A,int n)
{
    int i;
    for(i=0;i<n;i++)
        printf("%5d ",A[i]);
}
void nhanh()
{
    clrscr();
    int *A,n;
    printf("\n SAP XAP NHANH\n");
    printf("\n So phan tu n=");
    scanf("%d",&n);
    A=(int*)malloc(n*sizeof(int));
    printf("\n\n Tao lap day so:\n");
    for(int i=0;i<n;i++)
    {
        printf("\n A[%d]=",i);
        scanf("%d",&A[i]);
    }
    quick(A,n);
    printf("\n\n");
    printf("Ket qua thu duoc la:\n\n");
    in3(A,n);
    getch();
    free(A);
}
//=====
void in4(int *A,int n)
{
    for(int i=0;i<n;i++)
        printf("%5d",A[i]);
}
void merge(int *A,int n)
{
    int i,k,j,low1,up1,low2,up2,size;
    int *ds;
    size=1;
    ds=(int*)malloc(n*sizeof(int));
    while(size<n)
    {
        low1=0;

```

```

    k=0 ;
    while(low1+size<n)
    {
        low2=low1+size;
        up1=low2-1;
        if(low2+size-1<n)
            up2=low2+size-1;
        else
            up2=n-1;
        for(i=low1,j=low2;i<=up1 && j<=up2;k++)
        {
            if(A[i]<=A[j])
                ds[k]=A[i++];
            else
                ds[k]=A[j++];
        }
        for(;i<up1;k++)
            ds[k]=A[i++];
        for(;j<up2;k++)
            ds[k]=A[j++];
        low1=up2+1;
    }
    for(i=low1;k<n;i++)
        ds[k++]=A[i];
    for(i=0;i<n;i++)
        A[i]=ds[i];
    size*=2;
}
printf("\n \n Ket qua thu duoc la:\n\n");
in4(A,n);
free(ds);
}
void hoanhap()
{
    clrscr();
    int *A,n,i;
    printf("\n \t SAP XEP KIEU HOA NHAP\n");
    printf("\n So phan tu n=");
    scanf("%d",&n);
    A=(int*)malloc(n*sizeof(int));
    printf("\n Tao lap day so:\n");
    for(i=0;i<n;i++)
    {
        printf("\nA[%2d]=",i);
        scanf("%d",&A[i]);
    }
    merge(A,n);
    printf("\n");
    getch();
    free(A);
}

//=====
void in5(int *A,int n)
{
    for(int i=0;i<n;i++)

```

```

        printf("%5d",A[i]);
    }
void shaker(int *A,int n)
{
    int i,j,temp,tdoi;
    do
    {
        tdoi=0;
        for(i=n-1;i>0;i--)
            {
                if(A[i-1]>A[i])
                {
                    temp=A[i-1];
                    A[i-1]=A[i];
                    A[i]=temp;
                    tdoi=1;
                }
            }
        for(j=1;j<n;j++)
            {
                if(A[j-1]>A[j])
                {
                    temp=A[j-1];
                    A[j-1]=A[j];
                    A[j]=temp;
                    tdoi=1;
                }
            }
    }while(tdoi);
    printf("\n\n Ket qua la :",tdoi);
    in5(A,n);
}
void shaker()
{
    clrscr();
    int *A,n,i;
    printf("\n \tSHAKER_SORT\n");
    printf("\n So phan tu n=");
    scanf("%d",&n);
    A=(int*)malloc(n*sizeof(int));
    printf("\n \n Tao lap day so:\n");
    for(i=0;i<n;i++)
        {
            printf("\n A[%2d]=",i);
            scanf("%d",&A[i]);
        }
    shaker(A,n);
    getch();
    free(A);
}

//=====
void main()
{

```

```

while(1) {
clrscr();
int key;
printf("\n\tSAP XEP VA TIM KIEM\n");
printf("\n 1.Selection_sort\n");;
printf("\n 2.Bubble_sort\n");
printf("\n 3.Insertion_sort\n");
printf("\n 4.Quick_sort\n");
printf("\n 5.Merge_sort\n");
printf("\n 6.Shaker_sort\n");
printf("\n 0.Tro ve");
printf("\nBam mot phim de chon chuc nang:");
scanf("%d",&key);
if(key==0) break;

switch(key)
{
case 1:
clrscr();
luachon();
printf("\n\n\tAn phim bat ky de tro lai menu chinh");
getch();
clrscr();
break;
case 2:
clrscr();
suibot();
printf("\n\n\tAn phim bat ky de tro lai menu chinh");
getch();
clrscr();
break;
case 3:
clrscr();
tructiep();
printf("\n");
printf("\nAn phim bat ky de tro lai menu chinh");
getch();
clrscr();
break;
case 4:
clrscr();
nhanh();
printf("\n");
printf("\nAn phim bat ky de tro lai menu chinh");
getch();
clrscr();
break;
case 5:
clrscr();
hoanhap();
printf("\n");
printf("\nAn phim bat ky de tro lai menu chinh");
getch();
clrscr();
break;
case 6:

```



```

        clrscr();
        shaker();
        printf("\n");
        printf("\nAn phim bat ky de tro lai menu chinh");
        getch();
        clrscr();
        break;
    }
}
// getch();
}

```

Chuyển số La Mã sang số Ả rập

```

#include <iostream>
#include <conio.h>
#include <string.h>
int doigt(char chuoi[],int i);
void loi();

void main()
{
    char chuoi[20];
    cout<<"Nhap chuoi:";
    cin.get(chuoi,20);
    int nam=0;
    for(int i=0;i<strlen(chuoi);i++)
        nam+=doigt(chuoi,i);
    cout<<nam<<endl;
    getch();
}
int doigt(char chuoi[],int i)
{
    if(!(chuoi[i]=='I' || chuoi[i]=='V' || chuoi[i]=='X' || chuoi[i]=='L' |
| chuoi[i]=='C' || chuoi[i]=='D' || chuoi[i]=='M'))
        loi();
    if(chuoi[i]==chuoi[i+1]&&chuoi[i]==chuoi[i+2]&&chuoi[i]==chuoi[i+3])
        loi();
    if(chuoi[i]=='I')
        if(chuoi[i+1]=='V' || chuoi[i+1]=='X')
        {
            if(chuoi[i+2]=='V' || chuoi[i+2]=='X')
                loi();
            else
                return -1;
        }
    else
        return 1;
    if(chuoi[i]=='V')
        return 5;
    if(chuoi[i]=='X')
        if(chuoi[i+1]=='L' || chuoi[i+1]=='C')
        {
            if(chuoi[i+2]=='L' || chuoi[i+2]=='C')
                loi();
            else

```

```

        return -10;
    }
    else
        return 10;
    if(chuoi[i]=='L')
        return 50;
    if(chuoi[i]=='C')
        if(chuoi[i+1]=='D' || chuoi[i+1]=='M')
        {
            if(chuoi[i+2]=='D' || chuoi[i+2]=='M')
                loi();
            else
                return -100;
        }
    else
        return 100;
    if(chuoi[i]=='D')
        return 500;
    if(chuoi[i]=='M')
        return 1000;
}
void loi()
{
    cout<<"Day so tren khong hop le."<<endl;
}

```

Chuyển năm sang số La Mã

```

#include <iostream>
#include <math.h>
void viet(char a,char b,char c,int so);
void kytu(char &a,char &b,char &c,int so);
void main()
{
    char x,y,z;
    int nam;
    do
    {
        cout<<"Nhap nam(nho hon 4000): ";
        cin>>nam;
    } while(nam>=4000 || nam<0);
    cout<<"Nam "<<nam<<" viet trong he La Ma la: ";
    int tam=nam;
    int i=3;
    while(tam!=0)
    {
        tam=tam/pow(10.0,i);
        kytu(x,y,z,i);
        viet(x,y,z,tam);
        tam=fmod(nam,pow(10.0,i));
        i--;
    }
    cout<<endl;
}
void kytu(char &a,char &b,char &c,int i)
{
    if(i==0)

```

```

    {
        a='I';b='V';c='X';
    }
else if(i==1)
{
    a='X';b='L';c='C';
}
else if(i==2)
{
    a='C';b='D';c='M';
}
else if(i==3)
a='M';
}
void viet(char a,char b,char c,int so)
{
    if(so==1||so==2||so==3)
    for(int i=1;i<=so;i++)
    cout<<a;
    if(so==4)
        cout<<a<<b;
    if(so==5)
        cout<<b;
    if(so==6||so==7||so==8)
    {
        cout<<b;
        for(int i=1;i<=so-5;i++)
            cout<<a;
    }
    if(so==9)
    cout<<a<<c;
}

```

Thuật toán sắp xếp bằng Radix sort

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <conio.h>
/////Radix sort
int getMax(int a[],int n)
{
    int max=a[0];
    for(int i=1;i<n;i++)
        if(max<a[i])
            max=a[i];
    return max;
}
int countDigit(int n)
{
    int count=0;
    while(n)
    {
        count++;
        n/=10;
    }
    return count;
}

```

```

}
int getDigit(int n,int t)
{
    int tt=1;
    for(int i=0;i<t;i++)tt*=10;
    return ((n/tt)%10);
}
void send2Box(int a[],int n,int *b[10],int num[10],int t)
{
    for(int i=0;i<n;i++)
    {
        int tt=getDigit(a[i],t);
        b[tt][num[tt]++]=a[i];
    }
}
void getValue(int a[],int *b[10],int nn[10])
{
    int j=0;
    for(int i=0;i<10;i++)
    {
        if(nn[i]!=0)
        {
            for(int k=0;
                k<nn[i];
                k++)a[j++]=b[i][k];
            nn[i]=0;
        }
    }
}
void radixsort(int a[],int n)
{
    int *Box[10];
    int number[10];
    for(int i=0;i<10;i++)
    {
        Box[i]=new int [n];
        if(Box[i]==NULL)
        {
            printf("Not enough");
            exit(0);
        }
        number[i]=0;
    }
    int nn=countDigit(getMax(a,n));
    for(int i=0;i<nn;i++)
    {
        send2Box(a,n,Box,number,i);
        getValue(a,Box,number);
    }
}
void xuat(int a[], int n)
{
    int i=0;
    printf("\n\t");
    for (i=0;i<n;i++)
    {

```

```

        printf("%-6d",a[i]) ;
        if ((i+1) % 10 == 0)
            printf("\n\t");
    }
}
void main()
{
    clrscr();
    int i,n,A[100];
    do
    {
        printf("\n\tNhap so phan tu mang\n\t( n>0 va n<=100) :
");
        scanf("%d",&n);
    }while (n<0||n>100);
    for (i=0;i<n;i++)
    {
        printf("Nhap A[%d]=",i);
        scanf("%d",&A[i]);
    }
    printf("\n\tMang moi nhap vao:\n");
    xuat(A,n);
    printf("\n\n\tMang da sap xep :\n");
    radixsort(A,n);
    xuat(A,n);
    getch();
}

```

Danh sách liên kết đơn (Thuật toán vừa chèn vừa sắp xếp)

```

#include <iostream.h>
#include <conio.h>

class Node
{
private:
    float data;
    Node *next;
public:
    Node()
    {
        data = 0;
        next = NULL;
    }
    Node(float x)
    {
        data = x;
        next = NULL;
    }
    void setnext(Node *p)
    {
        next = p;
    }
    Node * getnext()
    {
        return next;
    }
}

```

```

        void setdata(float x)
        {
            data = x;
        }
        float getdata()
        {
            return data;
        }
};

class List
{
private:
    Node *head;
public:
    List()
    {head = NULL;}
    List(Node *x)
    {head = x;}

    void chensapxep(float x)
    Node *p,*q;
    p = new Node(x);
    q = head;
    if(head == NULL)
    head = p;
    else
    {
        if(p->getdata() < head->getdata())
        {
            p->setnext(head);
            head = p;
        }
        else
        {
            while ( q->getnext() != NULL)
            {
                if ((q->getdata() < p->getdata() && q->getnext()-
>getdata() > p->getdata())
                    || (q->getdata() == p->getdata() && q->getnext()-
>getdata() >= p->getdata()))
                {
                    p->setnext(q->getnext());
                    q->setnext(p);
                    break;
                }
                else
                q = q->getnext();
            }
            if (q->getnext() == NULL)
                q->setnext(p);
        }
    }
}
}

```

```

void show()
{
    Node *p = head;
    p = p->getnext();
    if( head != NULL)
    {
        while(p != NULL)
        {
            cout<<p->getdata()<<" ";
            p = p->getnext();
        }
    }
};
void main()
{
    float so;
    List p;
    do
    {
        cout<<"Nhap vao so (nhap 0 de thoat):";
        cin>>so;
        p.chensapxep(so);
    }while(so != 0);
    p.show();
    getch();
}

```

Quá tải toàn tử nhập xuất và sử dụng template

/*Chương trình này được viết để phục vụ và ôn tập lại các kiến thức sau :
qua tải toàn tử nhập xuất , sử dụng template để nhận mọi kiểu trả về dù
ở
truyền vào , viết lớp tương ứng cho tất cả các lớp cần sử dụng không
cần phải gọi trực tiếp .
Để viết dưới dạng tổng quát và dễ hình dung hơn . Trước qua tải toàn
tử nhập xuất với đối tượng mảng giá trị thì không có gì nhưng nay qua t
ai
toán tử nhập xuất với một mảng kí tự thì lại khác và dưới đây là một b
i như thế . */

```

#include <iostream.h>
#include <conio.h>

//viết lớp cat
class cat
{
    private :
        char name[20];
        int age;
    public:
        cat ()
        {
            for (int i=0;i<20;i++)
                name[i]='\0';
            age=0;
        }
};

```

```

    }
    cat (char ten[] ,unsigned int tuoi)
    {
        for (int i=0;i<20;i++)
        {
            name[i]=ten[i];
            if (ten[i]=='\n')
                break;
        }
        age=tuoi;
    }
    friend ostream & operator << (ostream & ,cat &);
    friend istream & operator >> (istream & ,cat &);
};

ostream & operator << (ostream & out ,cat &a)
{
    out<<a.age<<endl;
    out<<a.name<<endl;
    return out;
}
istream & operator >> (istream & in,cat &a)
{
    cout<<"nhap tuoi :";
    in>>a.age;
    in.ignore();
    cout<<"nhap ten:";
    in.getline(a.name,20);
    return in;
}

//viet lop dog
class dog
{
    private :
        char name[20];
        unsigned int age;
    public:
        dog ()
        {
            for (int i=0;i<20;i++)
                name[i]='\0';
            age=0;
        }
        dog (char ten[],unsigned int tuoi)
        {
            for (int i=0;i<20;i++)
            {
                name[i]=ten[i];
                if (ten[i]=='\n')
                    break;
            }
            age=tuoi;
        }
    friend ostream & operator << (ostream & ,dog &);
    friend istream & operator >> (istream & ,dog &);
};

```



```

ostream & operator << (ostream & out ,dog &a)
{
    out<<a.age<<endl;
    out<<a.name<<endl;
    return out;
}
istream & operator >> (istream & in,dog &a)
{
    cout<<"moi nhap tuoi:";
    in>>a.age;
    in.ignore();
    cout<<"moi nhap ten:";
    in.getline(a.name,20);
    return in;
}

//viet lop tuong trung]
//thu qua tai toan tu nhap va xuat cho lop total nay
template <class T>
class total
{
    private :
        T data;
    public:
        total (total<T> &b)
        {
            data=b.data;
        }
        total (T b)
        {
            data=b;
        }
    friend ostream & operator << (ostream & ,total<T> &);
    friend istream & operator >> (istream & ,total<T> &);
};

template <class T>
ostream & operator << (ostream & out ,total<T> &x)
{
    out<<x.data<<endl;
    return out;
}
template <class T>
istream & operator >> (istream & in,total<T> & x)
{
    in>>x.data;
    return in;
}

//viet phan main() cho chuong trinh

void main()
{
    //truong hop 1 : thu truyen mot doi tuong khong thuc cung lop vao cho
    lop tuong trung
    //va su dung ham in va nhap cho no
    //khai bao 1 con cat va nhap du lieu ,truyen cho doi tuong lop tuong tr

```

```

ung sau do in ra man hinh
    cat a;
    cout<<"moi nhap du lieu cho cat :"<<endl;
        cin>>a;
    total<cat> b(a);
    cout<<b;
    //OK khong loi
    //truong kop 2: truyen vao mot doi la dog cung ra ket qua tuong tu
    //truong hop 3: truyen vao cung kieu la total
    cat x;
    cout<<"moi nhap du lieu cho cat:"<<endl;
    cin>>x;
    total<cat> c(x);
    cout<<"thuc hien phep sao chep :"<<endl;
    total<cat> d(c);
    cout<<d<<endl;
    //OK khong loi vay da thanh cong hoan toan
    getch();
}
//Kinh nghiem rut ra tu bai nay la:
/* khi su dung template de dinh nghia toan tu nhap va xuat ta can de y
rang luc xuat ra
la xuat du lieu cua doi tuong .Phai nho doi tuong duoc goi den .Khong d
uoc xuat ra ngay du lieu
du no la ham friend
-
    Khi su dung qua tai toan tu ta thay bien "in" su dung y het cin o ngoa
i khi goi duoc ham
get() or getline () de lay ca ki tu trong' ke ca ham ignore() cung the
-
    Khi qua tai ta van su dung duoc ham cout or cin o trong qua tai toan t
u xuat hay nhap
Dac biet la cho nhap ten can phai tao ra mot mang dinh san khong the de
mang dong ,khi khai bao mang
dong ta se bi sai khi xuat ra man hinh .Do khi do di lieu ta nhap vao n
am o vung nho buffer
khi chay chuong trinh se vet het tat ca va gan cho bien hoac doi tuong
khi ta goi do do'
khong lam gi duoc .
*/

```

Chương trình đếm số ký tự trong một chuỗi ASCII

```

#include <stdio.h>
#include <ctype.h>
#include<conio.h>
void main()
{
    char chuoi[80];
    int i = 0, count = 0;

    printf("\nNhap vao mot chuoi bat ky : ");
    gets(chuoi);

    while (chuoi[i] != 0)
    {
        if (isalpha(chuoi[i++]))

```

```

        count++;
    }

    printf("So ky tu trong chuoai = %d", count);
    getch();
}

```

Biểu diễn số dưới dạng bit

```

#include <stdio.h>
#include <conio.h>
void main()
{
    unsigned int mang[24], i;
    int bit[16], k, index;
    printf("\nNhap vao 23 gia tri nguyen : ");
    for (i=0; i<23; i++)
        scanf("%d",&mang[i]);
    printf("          FEDCBA9876543210");
    for (i=0; i<23; i++)
    {
        k = mang[i];
        for (index = 0; index < 16; index++)
            bit[index] = 0;
        index = 15;
        while (k)
        {
            bit[index--] = k%2;
            k /= 2;
        }
        printf("\n%5d  ",mang[i]);
        for (index=0; index<16; index++)
            if (bit[index] == 1)
                printf("*");
            else
                printf("-");
    }
    getch()
}

```

Đảo chuỗi

Code:

```

char *dnchuoi(char *s)
{
    char *tmp, i;

    i = 0;
    tmp = (char *)malloc(strlen(s)+1);
    while (i<strlen(s))
        *(tmp+i) = *(s + strlen(s) - i++ - 1);
        *(tmp+i) = 0;
    return tmp;
}

```

```

void main()
{
    char hello[] = "Hello World";
    char *s;

    printf("\nChuoi ban dau = %s", hello);
    s = dnchuoi(hello);
    printf("\nChuoi dao nguoc = %s", s);
    getch();
}

```

Chương trình xem tập tin

Code:

```

#include <stdio.h>
#include <conio.h>

void main()
{
    FILE *fp;
    char filename[50], s[255], *hang[1000], c;
    int nline = 0, line = 0, i;

    printf("\nNhap ten tap tin muon xem : ");
    gets(filename);
    if ((fp = fopen(filename,"r")) == NULL)
        printf("\nKhong the mo tap tin %s", filename);
    else
    {
        while (!feof(fp))
        {
            fgets(s, 255, fp);
            hang[nline] = (char *)malloc(strlen(s));
            strcpy(hang[nline], s+1);
            nline++;
        }
        do {
            clrscr();
            i = line;
            while (i<line+26 && i<nline)
                printf("%s", hang[i++]);
            c = getch();
            if (c == 0)
            {
                c = getch();
                switch(c)
                {
                    case 72 : if (line > 0)
                                line--;
                                break;
                    case 80 : if (line+25 < nline)
                                line++;
                                break;
                }
            }
        }
    }
}

```

```

    }
    } while (c != 27);
    fclose(fp);
}
}

```

Giải bài toán trâu ăn cỏ

PHP Code:

```

/* Giai bai toan co :
   Tram trau tram co
   Trau dung an nam
   Trau nam an ba
   Trau gia an mot
*/

#include <stdio.h>
#include <conio.h>

void main()
{
    int tdung, tnam, tgia, phuongan=0;

    for (tdung = 1; tdung <= 98; tdung ++)
        for (tnam = 1; tnam < 99 - tdung; tnam ++)
            for (tgia = 1; tgia < 99 - (tdung + tnam); tgia++)
                if ((tdung*5 + tnam*3 + tgia) == 100)
                {
                    printf("\nTrau dung : %5d ; Trau nam : %5d ; Trau gia : %5d",
                        tdung, tnam, tgia);
                    phuongan++;
                }
    printf("\nTong cong co %d phuong an.", phuongan);
    getch();
}

```

Loại bỏ khoảng trống thừa trong chuỗi

PHP Code:

```

#include <stdio.h>
#include <string.h>
#include <conio.h>
#pragma warn -pia

char *trim(char *chuoi)
{
    char *p;
    while (p = strstr(chuoi, " "))
        memmove(p, p+1, strlen(chuoi) - (p - chuoi));
    if (chuoi[0] == ' ')
        memmove(chuoi, chuoi+1, strlen(chuoi) - 1);
    if (chuoi[strlen(chuoi)-1] == ' ')
        chuoi[strlen(chuoi)-1] = 0;
    return chuoi;
}

```

```

void main()
{
    char chuoi[125];

    printf("\nNhap chuoi mau : ");
    textattr(0x1e);
    gets(chuoi);

    trim(chuoi);
    printf("\nChuoi sau khi da trim : ");
    textattr(0x1e);
    cputs(chuoi);
    getch();
}

```

Tìm tất cả các ước của một số N

PHP Code:

```

#include <stdio.h>
#include <conio.h>

void main()
{
    int n, i;

    printf("Cho gia tri N = ");
    scanf("%d", &n);

    printf("Cac uoc so cua %d la :\n", n);
    for (i=1; i<n; i++)
        if ((n % i) == 0)
            printf("%5d", i);
    getch();
}

```

Bội số chung và ước số chung

PHP Code:

```

#include <stdio.h>
#include <conio.h>
unsigned USCLN (unsigned n, unsigned m)
{
    while (n != 0 && m != 0)
        if (n>m)
            n -= m;
        else
            m -= n;
    if (n == 0)
        return m;
    else
        return n;
}

```

```

unsigned BSCNN (unsigned n, unsigned m)
{
    return n * m / USCLN(n, m);
}

void main()
{
    unsigned n, m;

    printf("\nNhap hai vao so nguyen duong : ");
    scanf("%u%u", &n, &m);
    printf("\nUSCLN cua %u va %u = %u", n, m, USCLN(n,m));
    printf("\nBSCNN cua %u va %u = %u", n, m, BSCNN(n,m));
    getch();
}

```

Trộn 2 dãy giảm thành một dãy tăng

PHP Code:

```

#include <stdio.h>
#include <conio.h>
#define MAX 10

void main()
{
    int a[MAX], b[MAX], c[2*MAX], n1, n2, i, i1, i2;

    printf("\nCho biet so phan tu cua mang thu nhat : ");
    scanf("%d", &n1);
    printf("Nhap vao cac phan tu (giam dan) cua mang thu nhat : ");
    for (i=0; i<n1; i++)
        scanf("%d", &a[i]);
    printf("\nCho biet so phan tu cua mang thu hai : ");
    scanf("%d", &n2);
    printf("Nhap vao cac phan tu (giam dan) cua mang thu hai : ");
    for (i=0; i<n2; i++)
        scanf("%d", &b[i]);
    i1 = n1-1;
    i2 = n2-1;
    for (i=0; i<n1 + n2; i++)
    {
        if (i1 < 0 || i2 < 0)
            break;
        if (a[i1] < b[i2])
        {
            c[i] = a[i1];
            i1--;
        }
        else
        {
            c[i] = b[i2];
            i2--;
        }
    }
    if (i1 >= 0)
        while (i1 >= 0)

```

```

        c[i++] = a[i1--];
    if (i2 >= 0)
        while (i2 >= 0)
            c[i++] = b[i2--];
    printf("\nCac phan tu cua mang tron : ");
    for (i=0; i<n1+n2; i++)
        printf("%d ", c[i]);
    getch();
}

```

Tính tích 2 ma trận:

PHP Code:

```

#include<conio.h>
#include<stdio.h>
#include<alloc.h>
void main()
{
    int *a,*b,*c;
    int m,n;
    int i,j;
    clrscr();
    //Nhap so hang so cot
    printf("Nhap vao m:");scanf("%d",&m);
    printf("Nhap vao n:");scanf("%d",&n);
    //Cap phat bo nho
    a=(int*)calloc(m*n,sizeof(int));
    b=(int*)calloc(m*n,sizeof(int));
    c=(int*)calloc(m*n,sizeof(int));
    // Nhap so lieu va tinh toan
    for(i=1;i<=m;i++)
    for(j=1;j<=n;j++)
    {
        printf("Nhap vao a[%d,%d]=",i,j);scanf("%d",&a[(i-1+j)+((i-1)*(n-1))]);
    }
    for(i=1;i<=m;i++)
    for(j=1;j<=n;j++)
    {
        printf("Nhap vao b[%d,%d]=",i,j);scanf("%d",&b[(i-1+j)+((i-1)*(n-1))]);
        c[(i-1+j)+((i-1)*(n-1))]=a[(i-1+j)+((i-1)*(n-1))]+b[(i-1+j)+((i-1)*(n-1))];
    }
    // xuat cac mang a,b,c ra man hinh
    for(i=1;i<=m;i++)
    for(j=1;j<=n;j++)
    {
        printf("\t%d",a[(i-1+j)+((i-1)*(n-1))]);
        if(j==n)printf("\n");
    }
    printf("\n=====\\n");
    for(i=1;i<=m;i++)
    for(j=1;j<=n;j++)
    {
        printf("\t%d",b[(i-1+j)+((i-1)*(n-1))]);

```



```

        if(j==n)printf("\n");
    }
    printf("\n=====n");
    for(i=1;i<=m;i++)
    for(j=1;j<=n;j++)
    {
        printf("\t%d",c[(i-1+j)+((i-1)*(n-1))]);
        if(j==n)printf("\n");
    }
    getch();
}

```

In danh sách các số hoàn hảo nhỏ hơn số N nhập từ user

Code:

```

#include <stdio.h>

int main(void)
{
    int i , j , n , tong ;

    printf("\n Nhập vào N = "); scanf("%d",&n);

    for ( i = 2 ; i <=n ; i++ )
    {
        tong = 1 ;
        for ( j = 2; j <= i / 2 ; j++ )
            if ( i % j == 0 )    tong += j;
        if ( tong == i ) printf(" %10d ", i );
    }
    getch();
    return 0;
}

```

Bài in ra lịch của một năm bất kỳ lớn hơn 1700 (Bài này tớ làm đấy 🤖)

PHP Code:

```

#include <iostream.h>
#include <conio.h>
#include <stdlib.h>
int songay(int,int);
bool namnhuan(int nam);
void InLich(int,int);
int ThuDauTien(int,int);
void main()
{
    int nam;
    char chon;
    do
    {
        do

```

```

        {
            cout<<"Nhap vao nam(> 1700): ";
            cin>>nam;
        }while(nam < 1700);
        for(int i = 1;i <= 12;i++)
        {
            cout<<"Thang " <<i<<endl;
            InLich(i,nam);
            cout<<endl;
        }
        cout<<"\nBan co muon tiep tục khong(y/n): ";
        cin>>chon;
        system("cls");
    }while(chon == 'y');
    cout<<"Thanh Nam cam on ban da su dung chương trình {an Enter de tho
at)";
    getch();
}
int songay(int thang,int nam)
{
    switch(thang)
    {
        case 1:
        case 3:
        case 5:
        case 7:
        case 8:
        case 10:
        case 12:
            return 31;
        case 2:
            if(namnhuan(nam))
                return 29;
            else
                return 28;
        case 4:
        case 6:
        case 9:
        case 11:
            return 30;
    }
    return 0;
}
bool namnhuan(int nam)
{
    return ((nam % 4 == 0 && nam % 100 != 0) || (nam % 400 == 0));
}
void InLich(int thang,int nam)
{
    int ngayd = ThuDauTien(thang,nam);
    if(ngayd == 0)
        ngayd = 7;
    int i;
    int sn = songay(thang,nam);
    cout<<"\nMon\tTUE\tWED\tTHU\tFRI\tSAT\tSun\n";
    for(i = 1;i <= ngayd - 1;i++)
        cout<<"\t";
}

```

```

    for(i = 1;i <= sn;i++)
    {
        cout<<i<<"\t";
        if((i + ngayd - 1) % 7 == 0)
            cout<<endl;
    }
}

long int funct1 (int nam,int thang)
{
    long int result;
    if ( thang <= 2 )

        nam -
= 1;    //
        result = nam;

    return (result);
}

long int funct2 (int thang)
{
    long int result;
    if ( thang <= 2 )
        result = thang + 13;
    else
        result = thang + 1;
    return(result);
}

long int day_count (int thang, int nam)
{
    long int number;

    number = 1461 * funct1(nam,thang) / 4 + 153 * funct2(thang) / 5 +
1;
    return (number);
}

int ThuDauTien(int thang,int nam)
{
    long int number_of_days1;
    int day_of_week;
    number_of_days1 = day_count (thang, nam);
}

```

```

    day_of_week = (number_of_days1 -
621049) % 7;

    return day_of_week;
}

```

In code we trust

Bài tập kiểm tra dấu ngoặc đúng.

Ví dụ: $(5*8) + (4*7)$ có đủ 2 cái ngoặc mở và 2 cái ngoặc đóng, suy ra đây là chuỗi ngoặc đúng, hoặc $(6*6)+(7*12$ --> đây là chuỗi ngoặc sai.

Code:

```

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<string.h>
#define max 100
int stack[max],Top=-1;
void push(int stack[],int &Top,int X)
{
    if (Top==max-1)
    {
        puts("Stack day");
        exit(3);
    }
    else
    {
        Top++;
        stack[Top]=X;
    }
}
int pop(int stack[],int &Top)
{
    int tam;
    if (Top==--1) return Top;
    else
    {
        tam=stack[Top];
        Top--;
        return tam;
    }
}
void DauNgoac(char s[],int n)
{
    int i;
    for(i=0;i<n;i++)
    {
        if (s[i]=='(') push(stack,Top,1);
        if (s[i]==')' && pop(stack,Top)==-1)
        {
            Top=-2;

```

```

                                break;
                                }

                                }
                                if (Top==-1) printf("\nDung");
                                else printf("\nSai");
                                }
void main()
{
    clrscr();
    char s[max];
    int n=
    printf("Nhap chuoi:");
    gets(s);
    n=strlen(s);
    DauNgoac(s,n);
    getch();
}

```

Bài toán Tám Hoàng Hậu

Code:

```

#include <stdio.h>
#include<conio.h>
int dong[8], cot[8], cheoxuoi[15], cheonguoc[15];

void print ()
{
    int i;
    printf("\n");
    for (i=0; i<8; i++)
        printf("%3d", dong[i]);
}

void thu(int i)
{
    int j;
    for (j=0; j<8; j++)
    {
        if (cot[j] == 1 && cheoxuoi[i+j] ==1 && cheonguoc[i-j+7] == 1)
        {
            dong[i] = j;
            cot[j] = 0;
            cheoxuoi[i+j] = 0;
            cheonguoc[i-j+7] = 0;
            if (i<7)
                thu(i+1);
            else
                print();
            cot[j] = 1;
            cheoxuoi[i+j] = 1;
            cheonguoc[i-j+7] = 1;
        }
    }
}

```

```

    }
}

void tim()
{
    int i, q;

    for (i=0; i<8; i++)
    {
        cot[i] = 1;
        dong[i] = -1;
    }
    for (i=0; i<15; i++)
    {
        cheoxuoi[i] = 1;
        cheonguoc[i] = 1;
    }
    thu(0);
}

void main()
{
    tim();
    getch();
}

```

In ra số Hex tương ứng với một số nguyên dương

Code:

```

#include <stdio.h>
#include <conio.h>
void main()
{
    unsigned number;
    char hex[] = "0123456789ABCDEF";

    printf("\nNhap vao mot gia tri nguyen duong 16 bit : ");
    scanf("%u", &number);
    printf("Gia tri Hex tuong ung = %c%c%c%c",
           hex[number/0x1000], hex[(number/0x100)%0x10],
           hex[(number/0x10)%0x10], hex[number%0x10]);
    getch();
}

```

Liệt kê các hoán vị của N phần tử

Code:

```

#include <stdio.h>
#include <conio.h>
#define MAX 10
int mang[MAX], n;

```

```

void swap (int *x, int *y)
{
    int tmp;
    tmp = *x;
    *x = *y;
    *y = tmp;
}

void hoanvi(int k)
{
    int j;

    if (k==1)
    {
        printf("\n");
        for (j=0; j<n; j++)
            printf("%d ", mang[j]);
    }
    else
        for (j=k-1; j>=0; j--)
        {
            swap(&mang[k-1], &mang[j]);
            hoanvi(k-1);
            swap(&mang[j], &mang[k-1]);
        }
}

void main()
{
    int i;

    printf("\nCho biet so phan tu (N < 10) : ");
    scanf("%d", &n);
    for (i=0; i<n; i++)
        mang[i] = i;
    hoanvi(n);
    getch();
}

```

In chuỗi theo các từ mỗi từ một dòng

Code:

```

#include <stdio.h>
#include <conio.h>
#include <string.h>
void main()
{
    char s[50];
    int i, len;

    printf("\nNhap vao mot chuoi : ");
    gets(s);
    len = strlen(s);
    i = 0;
    while (i<len)

```

```

{
    while (s[i] == ' ' && i<len)
        i++;
    while (s[i] != ' ' && i<len)
        putchar(s[i++], stdout);
    putchar('\n', stdout);
}
getch();
}

```

In ra chữ số hàng trăm hàng chục hàng đơn vị

Code:

```

#include <stdio.h>
#include <conio.h>

void main()
{
    int n, tram, chuc, donvi;
    clrscr();
    printf("\nNhap vao mot so tu 100 - 999 : ");
    scanf("%d", &n);

    tram = n;
    donvi = tram % 10;
    tram /= 10;
    chuc = tram % 10;
    tram /= 10;

    printf("\nSo hang tram    = %d", tram);
    printf("\nSo hang chuc    = %d", chuc);
    printf("\nSo hang don vi   = %d", donvi);

    getch();
}

```

Tìm phần tử lớn nhất nhỏ nhất trong mảng một chiều

Code:

```

#include <conio.h>
#include <stdlib.h>

void main()
{
    int mang[20];

    int i, minval, maxval;

    /* Khoi tao mang ngau nhien */
    randomize();
    for (i=0; i<20; i++)
        mang[i] = random(100);

    /* Tim gia tri lon nhat va nho nhat */
    minval = maxval = mang[0];
}

```



```

for (i=1; i<20; i++)
{
    if (maxval < mang[i])
        maxval = mang[i];
    else if (minval > mang[i])
        minval = mang[i];
}

/* In mang */
clrscr();
for (i=0; i<20; i++)
{
    if (mang[i] == maxval)
        textcolor(YELLOW);
    else if (mang[i] == minval)
        textcolor(RED);
    else
        textcolor(WHITE);
    cprintf("%3d", mang[i]);
}

getch();
}

```

Tính tổ hợp chập K của N phần tử

Code:

```

#include <stdio.h>
#include <conio.h>

unsigned long giaithua(int n)
{
    unsigned long ketqua = 1;
    int i;

    for (i=2; i<=n; i++)
        ketqua *= i;

    return ketqua;
}

unsigned long to_hop_chap(int k, int n)
{
    return giaithua(n) / (giaithua(k) * giaithua(n-k));
}

void main()
{
    int n, k;

    printf("\nNhap vao gia tri N va K : ");
    scanf("%d%d", &n, &k);

    printf("Top hop chap %d cua %d = %lu", k, n, to_hop_chap(k, n));
    getch();
}

```

Chương trình đọc số có 3 chữ số.

Code:

```
/* Day la chuong trinh doc so co 3 chu so hay mot so co 2 cu so tham
chi so co 1 chu so
   Duoc viet boi Vu Thanh Nam */
#include<iostream.h>
#include<conio.h>
void main()
{
    int a;
    cout<<"Nhap vao mot so co 3 hay 2 tham chi 1 chu so: ";
    cin>>a;
    int hangtram=a/100;
    int hangchuc=((a/10)%10);
    int hangdonvi=a%10;
    switch(hangtram)
    {
        case 1 :cout<<"Mot tram ";break;
        case 2 :cout<<"Hai tram ";break;
        case 3 :cout<<"Ba tram ";break;
        case 4 :cout<<"Bon tram ";break;
        case 5 :cout<<"Nam tram ";break;
        case 6 :cout<<"Sau tram ";break;
        case 7 :cout<<"Bay tram ";break;
        case 8 :cout<<"Tam tram ";break;
        case 9 :cout<<"Chin tram ";break;
    }
    switch(hangchuc)
    {
        case 0:
            if(hangdonvi==0)
                cout<<" ";
            else
                {if(hangchuc==0&&hangtram==0)
                    cout<<" ";
                    else
                        cout<<"ninh";}
                break;
        case 1:cout<<"muoi";break;
        case 2:cout<<"hai muoi";break;
        case 3:cout<<"ba muoi";break;
        case 4:cout<<"bon muoi";break;
        case 5:cout<<"nam muoi";break;
        case 6:cout<<"sau muoi";break;
        case 7:cout<<"bay muoi";break;
        case 8:cout<<"tam muoi";break;
        case 9:cout<<"chin muoi";break;
    }
    switch(hangdonvi)
    {
        case 0:cout<<"Khong";break;
        case 1:
```

```

        if (hangchuc==1 || (hangtram==0 && hangchuc==0))
            cout<<" mot";
        else
            cout<<" mo't";
        break;
    case 2: cout<<" hai"; break;
    case 3: cout<<" ba"; break;
    case 4:
        if (hangchuc==1)
            cout<<" bon";
        else
            {if (hangchuc==0 && hangtram==0)
                cout<<"bon";
            else
                cout<<" tu";}
        break;
    case 5:
        if (hangchuc==0 && hangtram==0)
            cout<<" nam";
        else
            cout<<" lam";
        break;
    case 6: cout<<" sau"; break;
    case 7: cout<<" bay"; break;
    case 8: cout<<" tam"; break;
    case 9: cout<<" chin"; break;
}
    getch();
}

```

Tính số ngày trong một tháng trong một năm bất kỳ.

Code:

```

/* Day la chuong trinh tinh so ngay trong bat ky mot thang trong nam
bat ky nao do
   Duoc viet boi Vu Thanh Nam */
#include<iostream.h>
#include<conio.h>
#include<math.h>
void main()
{
    long int thang, nam;
    cout<<"Nhap thang va nam: ";
    cin>>thang>>nam;
    if ((thang>12 || thang<1) || (nam<0))
        cout<<"Nhap thang hoac nam sai";
    else
    {
        bool namnhuan= ((nam%4==0 && nam%100!=0) || (nam%400==0 && nam%100==0));
        int songay;
        if (thang==4 || thang==6 || thang==9 || thang==11)
            songay = 30;
        else
        {
            if (thang==2)

```

```

    songay = namnhuan?29:28;
    else
        songay=31;
    }
    cout<<"So ngay cua thang "<<thang<<" trong nam "<<nam<<" la:
"<<songay<<endl;
    }
    getch();
}

```

Bài kiểm tra số nguyên tố

Code:

```

/*Ham tinh so nguyen to */
#include<iostream.h>
#include<conio.h>
#include<math.h>
bool LaNguyenTo(int n);
void main()
{
    int n;
    cout<<"Nhap vao mot so bat ky: ";cin>>n;
    if(LaNguyenTo(n))
        cout<<"La so nguyen to";
    else
        {
            cout<<"Khong la so nguyen to vi no chia het cho ";
            for(int j=2;j<n;j++)
            {
                if(n%j==0)
                    cout<<j<<" va ";
            }
            cout<<"The thoi";
        }
    getch();
}
bool LaNguyenTo(int n)
{
    bool co=true;
    if(n<2)
    {cout<<"Du lieu nhap sai va so do ";
    return !co;}
    else
    {
        for(int i=2;i<=sqrt(n)+1;i++)
        {
            if(n%i==0)
            {
                co=false;
                break;
            }
        }
    }
    return co;
}

```

Bài tập tìm max min của 4 số.

Code:

```
#include<iostream.h>
#include<conio.h>
#include<math.h>
float max(int a, int b);
float min(int a, int b);
void main()
{
    int a,b,c,d;
    cout<<"Nhap vao so thu 1: ";cin>>a;
    cout<<"Nhap vao so thu 2: ";cin>>b;
    cout<<"Nhap vao so thu 3: ";cin>>c;
    cout<<"Nhap vao so thu 4: ";cin>>d;
    cout<<"max= "<<max(max(max(a,b),max(b,c)),max(c,d))<<" min="
"<<min(min(min(a,b),min(b,c)),min(c,d))<<endl;
    cout<<"Vay trung binh cong cua 4 so do la: "<<(a+b+c+d)/4;
    getch();
}
float max(int a, int b)
{
    if(a>b)
        return a;
    else
        return b;
}
float min(int a, int b)
{
    if(a>b)
        return b;
    else
        return a;
}
```

Tìm n số Fibonacci đầu tiên

Code:

```
#include<iostream.h>
#include<conio.h>
#include<math.h>
long Fibonacci(long n);
void main()
{
    long n;
    cout<<"Nhap vao mot so nguyen duong: ";
    cin>>n;
    cout<<"Vay "<<n<<" so Fibonacci dau tien la: ";
    Fibonacci(n);
    getch();
}
long Fibonacci(long n)    //sach giao khoa 253
```

```

{
    int f1=0,f2=1,fi;
    fi=f2;
    for(int i = 1;i <= n ;i ++)
    {
        cout<<fi<<" ";
        fi=f1+f2;
        f1=f2;
        f2=fi;
    }
    return fi;
}

```

(Ngân hàng) Tìm số tiền nhận trong n tháng khi biết lãi suất

Code:

```

#include<iostream.h>
#include<conio.h>
#include<math.h>
void main()
{
    double  sotiengui,tienlai,laixuat;
    int sothang;
    cout<<"Nhap vao so tien gui: ";cin>>sotiengui;
    cout<<"Nhap vao lai xuat(%): ";cin>>laixuat;
    cout<<"Nhap vao so thang gui: ";cin>>sothang;
    for(int i=1;i<=sothang;i++)
    {
        tienlai=laixuat*sotiengui;
        sotiengui=sotiengui+tienlai;
    }
    cout<<"Vay so tien nguoi do gui trong "<<sothang<<" thang la:
"<<sotiengui;
    getch();
}

```

In ra dãy số ngược so với dãy số nhập vào

Code:

```

#include <iostream.h>
#include <conio.h>
void main()
{
    long a;
    do
    {
        cout<<"a (Phai la so nguyen duong) = ";
        cin>>a;
    }while(a<0);
    cout<<"Vay "<<a<<" duoc viet nguoc lai la: ";
    if (a > 0)
    {
        while (a > 0)

```

```

        {
            cout<<a%10;
            a = a / 10;
        }
    }
    getch();
}

```

Trò chơi 8 hòn bi

Code:

```

/*tro choi 8 hon bi
vu thanh nam*/
#include <iostream.h>
#include <conio.h>
#include <stdlib.h>

void main()
{
    char chon;
do
{
    cout<<"\nCHUONG TRINH DOAN HON BI KHAC KHOI LUONG TRONG 8 HON
BI\n";
    cout<<"\n\t-Co 8 hon bi danh so tu 1 den 8\n";
    cout<<"\t 1    2    3    4    5    6    7    8\n";
    cout<<"\t-Co duy nhat 1 hon bi khac khoi luong khong biet nhe hay
nang hon so bi con lai\n";
    cout<<"\t-Chi su dung 1 chiec can 2 dia (can thang bang) de phat
hien ra hon bi do \n";
    cout<<"\t-Ban nghi trong dau va chon 1 trong nhung hon bi do\n";
    cout<<"\t-Tra loi trung thuc nhung cau hoi sau, may tinh se doan ra
hon bi ma ban da chon \n";
    cout<<"\t-Luu y: may tinh chi can toi da 3 lan\n";
    cout<<"-----
\n";
    cout<<"Chon -1 neu dia 1 nhe, 1 neu dia 1 nang va 0 neu 2 dia
can bang \n";
    cout<<"-----
\n";
    cout<<"1. CAN LAN 1:\n";
    cout<<"\t-Dia 1 gom hon 1,2,3\n";
    cout<<"\t-Dia 2 gom hon 4,5,6 \n";
    int kq1;
    cout<<"Moi ban nhap ket qua can: ";
    cin>>kq1;
    if(kq1==0)
    {
        cout<<"2. CAN LAN 2:\n";
        cout<<"\t-Dia 1 gom qua 1\n";
        cout<<"\t-Dia 2 gom qua 7 \n";
        int kq2;
        cout<<"Moi ban nhap ket qua can: ";cin>>kq2;
        if(kq2==0)

```

```

{
cout<<"3.CAN LAN 3:\n";
cout<<"\t-Dia 1 gom hon 1 \n";
cout<<"\t-Dia 2 gom hon 8 \n";
int kq3;
cout<<"Moi ban nhap ket qua can: ";cin>>kq3;
if(kq3==1||kq3==-1)
    cout<<((kq3==1)?"Hon 8 nhe!!!\n":"Hon 8 nang!!!\n");

else
    cout<<"BAN CHUA HIEU LUAT CHOI - AN GIAN KIA\n";

}
else if (kq2==1||kq2==-1)
    cout<<((kq2==1)?"Hon 7 nhe!!!\n":"Hon 7 nang!!!\n");

else
    cout<<"BAN CHUA HIEU LUAT CHOI - AN GIAN KIA\n";

}
else if (kq1==-1||kq1==1)
{
cout<<"2.CAN LAN 2:\n";
cout<<"\t-Dia 1 gom hon 1,2\n";
cout<<"\t-Dia 2 gom hon 3,4\n";
int kq2;
cout<<"Moi ban nhap ket qua can: ";cin>>kq2;
if(kq2==0)
{
    cout<<"3.CAN LAN 3:\n";
    cout<<"\t-Dia 1 gom hon 5\n";
    cout<<"\t-Dia 2 gom hon 6\n";
    int kq3;
    cout<<"Moi ban nhap ket qua can: ";cin>>kq3;
    if (kq3==1||kq3==-1)
    {
        if(kq1==1)
            cout<<((kq3==1)?"Hon 6 nhe!!\n":"Hon 5 nhe!!!\n");

        else
            cout<<((kq3==1)?"Hon 5 nang!!\n":"Hon 6 nang!!!\n");

    }
    else
        cout<<"BAN CHUA HIEU LUAT CHOI-AN GIAN KIA\n";

}
else if(kq2==1||kq2==-1)
{
    if(kq1!=kq2)
        cout<<((kq1==1)?"Hon 3 nang!!!\n":"Hon 3 nhe!!!\n");

    else
    {
        cout<<"3.CAN LAN 3:\n";
        cout<<"\t-Dia 1 gom hon 1\n";

```



```

cout<<"\t-Dia 2 gom hon 2\n";
int kq3;
cout<<"Moi ban nhap ket qua can: ";cin>>kq3;
if(kq3==0)
    cout<<((kq1==1)?"Hon 4 nhe!!!\n":"Hon 4 nang!!!\n");

else if(kq3==1||kq3==-1)
{
    if(kq1==1)
        cout<<((kq3==1)?"Hon 1 nang!!\n":"Hon 2
nang!!\n");

    else
        cout<<((kq3==1)?"}Hon 2 nhe!!!\n":"Hon 1 nhe!!!\n");

}
else
    cout<<"BAN CHUA HIEU LUAT CHOI-AN GIAN KIA\n";

}
else
    cout<<"BAN CHUA HIEU LUAT CHOI-AN GIAN KIA\n";

}
else
    cout<<"BAN CHUA HIEU LUAT CHOI\n";

cout<<"\nBan co muon choi tiep khong(c/k): ";
cin>>chon;
system("cls");
}while(chon == 'c');
}

```

Kiểm tra số đối xứng

Code:

```

/*Vu Thanh Nam
Bai tap 43 */
#include<iostream.h>
#include<conio.h>
#include<math.h>
long songuoc(long a);
void main()
{
    long n;
    do
    {
        cout<<"Nhap vao mot so bat ky: ";
        cin>>n;
    }while(n<0);
    if(n==songuoc(n))
        cout<<"La so doi xung";
    else

```

```

    cout<<"Khong la so doi xung";
    getch();
}
long songuoc(long a)
{
    int tg;
    if (a > 0)
    {
        while (a > 0)
        {
            tg= a%10;
            a = a / 10;
        }
    }
    return tg;
}

```

Điền giá trị cho một mảng vuông theo chiều kim đồng hồ

Code:

```

#include <stdio.h>
#include <conio.h>
void main()
{
    int n, w, hang, cot, chanhangtang, chanhanggiam, chancottang,
    chancotgiam;
    int matran[12][12], tang, giam;

    printf("\n\nChuong trinh in ma tran dac biet.");
    do {
        printf("\nNhap kích thước của ma tran (3 <= n <= 12) : ");
        scanf("%d%c", &n);
    } while (! (n >=3 && n<=12) );
    hang = cot = 0;
    chanhangtang = chancottang = n-1;
    chanhanggiam = chancotgiam = 0;
    /*
        tang có 4 giá trị :
        tang = 0 -> tang cot
        tang = 1 -> tang hang
        tang = 2 -> giam cot
        tang = 3 -> giam cot
    */
    tang = 0;
    for (w=0; w<n*n; w++)
    {
        matran[hang][cot]= w + 1;
        switch(tang)
        {
            case 0 : if (cot < chancottang)
                    cot++;
                    else
                    {
                        tang++;      // Chuyen sang tang hang
                        chanhanggiam++;
                    }
        }
    }
}

```

```

        hang++;
    }
    break;
case 1 : if (hang < chanhangtang)
        hang++;
    else
    {
        tang++;    // Chuyen sang giam cot
        chancottang--;
        cot--;
    }
    break;
case 2 : if (cot > chancotgiam)
        cot--;
    else
    {
        tang++;    // Chuyen sang giam hang
        chanhangtang--;
        hang--;
    }
    break;
case 3 : if (hang > chanhanggiam)
        hang--;
    else
    {
        tang=0;    // Chuyen sang tang cot
        chancotgiam++;
        cot++;
    }
    break;
    }
}
for (hang=0; hang < n; hang++)
{
    printf("\n");
    for (cot=0; cot < n; cot++)
        printf("%4d", matran[hang][cot]);
}
getch();
}

```

In hình tam giác

Code:

```

#include <stdio.h>
#include <conio.h>

void main()
{
    int n, i, j;

    printf("\nNhap chieu cao cua hinh tam giac : ");
    scanf("%d", &n);

```

```

for (i=0; i<n; i++)
{
    for (j=n-1; j>i; j--)
        printf(" ");
    for (j=0; j<i*2+1; j++)
        printf("*");
    printf("\n");
}

getch();
}

```

Trộn hai mảng tăng dần thành một mảng tăng dần

Code:

```

/* Tron hai mang tang dan thanh 1 mang tang dan */
#include <stdio.h>

#define MAX 10

void main()
{
    int a[MAX], b[MAX], c[2*MAX], n1, n2, i, i1, i2;

    printf("\nCho biet so phan tu cua mang thu nhat : ");
    scanf("%d", &n1);
    printf("Nhap vao cac phan tu (tang dan) cua mang thu nhat : ");
    for (i=0; i<n1; i++)
        scanf("%d", &a[i]);
    printf("\nCho biet so phan tu cua mang thu hai : ");
    scanf("%d", &n2);
    printf("Nhap vao cac phan tu (tang dan) cua mang thu hai : ");
    for (i=0; i<n2; i++)
        scanf("%d", &b[i]);
    i1 = i2 = 0;
    for (i=0; i<n1 + n2; i++)
    {
        if (i1 >= n1 || i2 >= n2)
            break;
        if (a[i1] < b[i2])
        {
            c[i] = a[i1];
            i1++;
        }
        else
        {
            c[i] = b[i2];
            i2++;
        }
    }
    if (i1 < n1)
        while (i1 < n1)
            c[i++] = a[i1++];
    if (i2 < n2)
        while (i2 < n2)

```

```

    c[i++] = b[i2++];
printf("\nCac phan tu cua mang tron : ");
for (i=0; i<n1+n2; i++)
    printf("%d ", c[i]);
getch();
}

```

Tìm vị trí đầu và vị trí cuối của một số trong một dãy số

Code:

```

#include<iostream.h>
#include<conio.h>
#include<math.h>
void main()
{
    int a,n,m,vtdau=0,vtcuoi=0;
    cout<<"Nhap vao so luong so: ";
    cin>>n;
    cout<<"Ban muon kiem tra so nao: ";
    cin>>m;
    do
    {
        cout<<"Nhap vao so thu 1: ";
        cin>>a;
    }while(a<0);
    for(int i=2;i<=n;i++)
    {
        do
        {
            cout<<"Nhap vao so thu "<<i<<": ";
            cin>>a;
        }while(a<0);
        if(a==m)
        {
            if(vtdau==0)
                vtdau=i;
            vtcuoi=i;
        }
    }
    cout<<"Vay vi tri dau cua so "<< m<< "la: "<<vtdau<<endl;
    cout<<"Vay vi tri cuoi cua so "<< m<< "la: "<<vtcuoi<<endl;
    getch();
}

```

Tính $x^1/1! + x^2/2! + x^3/3! + \dots + x^n/n!$

Code:

```

#include<iostream.h>
#include<conio.h>
#include<math.h>
void main()
{
    float x,tong=0.0,n,tg,mu=1.0,gt=1.0;

```

```

cout<<"Nhap vao mot so thuc x= ";cin>>x;
cout<<"Nhap vao n= ";cin>>n;
for(int i=1;i<=n;i++)
{
    mu=mu*x;
    gt=gt*i;
    tg=mu/gt;
    tong=tong+tg;
}
cout<<"Vay ";
for(int j=2;j<=n;j++)
{
    cout<<x<<"^"<<j<<"/"<<j<<"!"<<" + ";
}
cout<<x<<" = "<<(tong);
getch();
}

```

Trình bày các bước chuyển n đĩa từ cọc A sang cọc C trong bài toán Tháp Hà Nội dùng 3 đĩa

Code:

```

#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <dos.h>

#define A 0
#define B 1
#define C 2

void hanoi(int from, int to, int dia)
{
    int trunggian;
    if (dia == 1)
        printf("\nChuyen 1 dia tu coc %c sang coc %c", 'A'+from, 'A'+to);
    else
    {
        if ((from == A && to == C) || (from == C && to == A))
            trunggian = B;
        else if ((from == A && to == B) || (from == B && to == A))
            trunggian = C;
        else if ((from == C && to == B) || (from == B && to == C))
            trunggian = A;
        hanoi(from, trunggian, dia-1);
        hanoi(from, to, 1);
        hanoi(trunggian, to, dia-1);
    }
}

void main()
{
    int n;

    printf("\nCho biet so dia ( 3 -> 6 ): ");
}

```

```

do {
    scanf("%d", &n);
} while (n < 3 || n > 6);
hanoi(A, C, n);
getch();
}

```

Trình bày các bước chuyển n đĩa từ cọc A sang cọc C trong bài toán Tháp Hà Nội dùng 4 đĩa

Code:

```

#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <dos.h>

#define A 0
#define B 1
#define C 2
#define D 3

void hanoi(int from, int to, int dia)
{
    int tg1, tg2, dia2;
    if (dia <= 1)
        printf("\nChuyen 1 dia tu coc %c sang coc %c", 'A'+from, 'A'+to);
    else
    {
        if ((from == A && to == D) || (from == D && to == A))
        {
            tg1 = B;
            tg2 = C;
        }
        else if ((from == A && to == B) || (from == B && to == A))
        {
            tg1 = C;
            tg2 = D;
        }
        else if ((from == A && to == C) || (from == C && to == A))
        {
            tg1 = B;
            tg2 = D;
        }
        else if ((from == B && to == C) || (from == C && to == B))
        {
            tg1 = A;
            tg2 = D;
        }
        else if ((from == B && to == D) || (from == D && to == B))
        {
            tg1 = A;
            tg2 = C;
        }
        else if ((from == C && to == D) || (from == D && to == C))
        {

```

```

    tg1 = A;
    tg2 = B;
}
dia2 = dia / 2;
if (dia % 2 == 1)
{
    hanoi(from, tg1, dia2);
    hanoi(from, tg2, dia2);
    hanoi(from, to, 1);
    hanoi(tg2, to, dia2);
    hanoi(tg1, to, dia2);
}
else
{
    hanoi(from, tg1, dia2);
    if (dia2 > 1)
    {
        hanoi(from, tg2, dia2-1);
        hanoi(from, to, 1);
        hanoi(tg2, to, dia2-1);
    }
    else
        hanoi(from, to, 1);
    hanoi(tg1, to, dia2);
}
}
}

void main()
{
    int n;

    printf("\nCho biet so dia ( 3 -> 6 ): ");
    do {
        scanf("%d", &n);
    } while (n < 3 || n > 6);
    hanoi(A, D, n);
    getch();
}

```

câu 12 trang 3: các ước của số n, với n nhập từ bàn phím :

PHP Code:

```

#include<conio.h>
#include<stdio.h>
void LK_Uoc(int n)//liet ke cac uoc cua n
{
    for(int i=1;i<=n;i++)
        if(n%i==0)
            printf("%4d",i);
}
void main()
{
    int n;
    printf("\nMoi ban nhap so n bat ky n =");
    scanf("%d",&n);
}

```



```

printf("\nCac uoc so cua n la : \n ");
LK_Uoc(n);
getch();
}

```

Tìm vị trí đầu và vị trí cuối của một số trong một dãy số.

```

#include<conio.h>
#include<stdio.h>
void Nhap_Mang(int A[],int n)
{
    for(int i=0;i<n;i++)
    {
        printf("A[%d=",i);
        scanf("%d",&A[i]);
    }
}
void Xuat_Mang(int A[],int n)
{
    for(int i=0;i<n;i++)
        printf("%4d",A[i]);
}
int Tim_X_DT(int A[],int n,int X)
{
    for(int i=0;i<n;i++)
        if(A[i]==X)
            return i;
    return 0;
}
int Tim_X_CC(int A[],int n,int X)
{
    for(int i=n-1;i>=0;i--)
        if(A[i]==X)
            return i;
    return 0;
}
void main()
{
    int A[100];
    int n,X;
    printf("\nMoi ban nhap so phan tu n = ");
    scanf("%d",&n);
    Nhap_Mang(A,n);
    printf("\nMang sau khi nhap :");
    Xuat_Mang(A,n);
    printf("\nMoi ban nhap gia tri X =");
    scanf("%d",&X);
    int kq=Tim_X_DT(A,n,X);
    if(kq==0)
        printf("\n Mang vua nhap ko co gia tri X");
    else
        printf("\nGia tri X dau tien trong mang =%d",kq);
    int kq1=Tim_X_CC(A,n,X);
    printf("\nGia tri X cuoi cung trong mang =%d",kq1);
    getch();
}

```

bai 16 :In danh sách các số hoàn hảo(hoàn hảo =hoàn thiện) nhỏ hơn số N nhập từ user

```

#include<conio.h>
#include<stdio.h>
int La_HT(int n)//La_HT=La_HH
{
    int tong=0;
    for(int i=1;i<n;i++0)
        if(n%i==0)
            tong=tong+i;
    return(tong==n);
}
void LK_HT(int n)
{
    for(int i=0;i<=n;i++)
        if(La_HT(i)==1)
            printf("%4d",i);
}
void main()
{
    int n;
    printf("\nMoi ban nhap so n = ");
    scanf("%d",&n);
    printf("\nLiet ke Hoan hao nho hon n :");
    LK_HT(n);
    getch();
}

```

bài 12 trang 4 kiểm tra nguyên tố :

```

#include<conio.h>
#include<stdio.h>
int La_NT(int n)
{
    int dem=0;
    for(int i=1;i<=n;i++)
        if(n%i==0)
            dem++;
    return(dem==2);
}
void main()
{
    int n;
    printf("\nMoi ban nhap n =");
    scanf("%d",&n);
    int kq=La_NT(n);
    if(kq==0)
        printf("\nso vua nhap ko phai la NT");
    else
        printf("\nLa NT");
    getch();
}

```

Kiểm tra ma trận B có là ma trận con của ma trận A hay ko :

```

#include<stdio.h>
#include<conio.h>

#define N 2
#define M 5

```

```

void search(int b[N][N], int a[M][M])
{
    int i, j, k, l, m, x, y, dem, demx, demy;
    int timthay=1;

    for(k=0; k<=M-N; ++k)
    {
        for(l=0; l<=M-N; ++l)
        {
            dem=demx=demy=0;
            x=l; y=k;
            for(i=0; i<N; ++i)
            {
                for(j=0; j<N; ++j)
                {
                    if(b[i][j]==a[y][x]) dem++;
                    ++x; ++demx;
                    if(demx==N) {demx=0; x=l;}
                }
                ++y; ++demy;
                if(demy==N) {demy=0; y=k;}
            }
            if(dem==N*N) break;
        }
        if(dem==N*N) break;
    }

    if(dem!=N*N) timthay=0;
    if(timthay==0) printf("\nKo tim thay");
    else printf("\nTim thay");
}

void main()
{
    clrscr();
    int a[M][M]={ 1, 2, 3, 4, 5,
                  6, 7, 8, 9, 10,
                  11, 12, 13, 14, 15,
                  16, 17, 18, 19, 20,
                  21, 22, 23, 24, 25};

    int b[N][N]={4, 5,
                 9, 10};

    search(b, a);
}

```

```

    getch();
}

```

Bai 2: in hình sau ra màn hình (in hình tam giác cân)

```

-----1
-----121
-----12121
-----1212121
...

```

Code:

```

#include <stdio.h>
#include <conio.h>

void main() {
    clrscr();
    short d[]={1,2};
    printf("Nhap chieu cao tam giac: ");
    int i, num, space;
    scanf("%d", &i);
    num=2*i+1;
    for(int n=0; n<i; ++n) {
        space=(num-(2*n+1))/2;
        while(space-->0)
            printf(" ");
        int index;
        for(int j=0; j<2*n+1; ++j) {
            index=(j%2)?1:0;
            printf("%d", d[index]);
        }
        printf("\n");
    }
    getch();
}

```

Code:

```

#include<conio.h>
#include<iostream.h>
//by sakervista    http://th6b.net
main()
{
    int i, j, k, h;
    printf("Nhap Chieu cao h : "); scanf("%d", &h);
    for(i=1; i<=h; i++)
    {
        for(k=1; k<=h-i+1; k++)
        {
            printf(" ");           // in ra khoảng

```

trắng

```

    }
    for(j=1;j<=2*i-1;j++)
    {
        if(j%2!=0) // nếu j chẵn thì
        {
            printf("1");
        }
        else
            printf("2");
    }

    printf("\n");
}
getch();
}

```

Một số hàm thông dụng :

PHP Code:

```

long giaithua(unsigned n)
{
    return (n<=0)?1:n*giaithua(n-1);
}
long haicham(unsigned n)
{
    return (n<=0||n==1)?1:n*haicham(n-2);
}
int UCLN(int a,int b)
{
    return (b==0)?a:UCLN(b, a%b);
}
long BCNN(int a,int b)
{
    return long(a)*b/UCLN(a,b);
}
int tohop(unsigned k,unsigned n)
{
    if (k>n) return 0;
    return (k<=0||k==n)?1:tohop(k-1,n-1)+tohop(k,n-1);
}
long fibonacci(unsigned n)
{
    return (n==1||n==2)?1: fibonacci(n-1)+fibonacci(n-2) ;
}
char nguyento(unsigned n)
{
    if (n<2) return 0;
    for (int i=2; i<= sqrt(n) ; i++ )
        if (n%i==0) return 0;
    return 1;
}
char hoanhao(unsigned n)
{

```

```

        int s=0,i;
        for (i=1;i<n;i++)
            if (n%i==0) s+=i;
        if (s==n) return 1;
        else return 0;
    }
char chinhphuong(unsigned n)
{
    if (n==0) return 1;
    if (n==int(sqrt(n))*int(sqrt(n))) return 1;
    return 0;
}

```

PHP Code:

```

char doixung(float *mang,int n) //n la so phan tu ( nho nhat la 1)
{
    for (int i=0;i<n/2;i++)
        if (mang[i]!=mang[n-i-1]) return 0;
    return 1;
}
void xapxepngd(int mang[], int n) // xap xep nguyen giam dan
{
    int i,j;
    for (i=0;i<n-1;i++)
        for (j=i+1;j<n;j++)
        {
            if (mang[i]<mang[j])
                mang[i]^=mang[j]^=mang[i]^=mang[j];
        }
}
void xapxepntd(int mang[], int dodai) // xap xep nguyen tang dan
{
    int i,j;
    for (i=0;i<=dodai-2;i++)
        for (j=i;j<dodai;j++)
        {
            if (mang[i]>mang[j])
                mang[i]^=mang[j]^=mang[i]^=mang[j];
        }
}
void xapxepthgd(float mang[], int dodai) // xap xep thuc giam dan
{
    int i,j;
    float temp;
    for (i=0;i<=dodai-2;i++)
        for (j=i;j<dodai;j++)
        {
            if (mang[i]<mang[j])
            {
                temp=mang[i];
                mang[i]=mang[j];
                mang[j]=temp;
            }
        }
}

```

```

    }
void xapxepthd(float mang[], int dodai) // xap xep thuc tang dan
{
    int i,j;
    float temp;
    for (i=0;i<=dodai-2;i++)
        for (j=i;j<dodai;j++)
        {
            if (mang[i]>mang[j])
            {
                temp=mang[i];
                mang[i]=mang[j];
                mang[j]=temp;
            }
        }
    }
}

```

PHP Code:

```

int strlen(char *xau)
{
    int i=0;
    while (xau[i]!=NULL) i++;
    return i;
}

void xoakitu(int vitri , int soluong , char *xau)
{
    if (vitri+soluong-1>strlen(xau)) xau[0]=NULL;
    else
    {
        while (xau[vitri+soluong]!=NULL)
        {
            xau[vitri]=xau[vitri+soluong];
            vitri++;
        }
        xau[vitri]=NULL;
    }
}

void chenkitu(int vitri, char *chen , char xau*)
{
    int j=0,i=vitri;
    char temp[99];
    while (xau[i]!=NULL) temp[j++]=xau[i++];
    temp[j]=NULL;
    j=0;
    while (chen[j]!=NULL) xau[vitri++]=chen[j++];
    j=0;
    while (temp[j]!=NULL) xau[vitri++]=temp[j++];
    xau[vitri]=NULL;
}

//viết 1 xâu kí tự : viết từng chữ 1 ra màn hình
void viet(char *xau="Quang",int tre=15);
void viet(char xau[],int tre)
{
    int i;
    for ( i=0 ; i<strlen(xau) ; ++i)

```

```

        { putchar(xau[i]);delay(tre);}
    }

//cho 1 xâu kí tự chạy từ vị trí x y gốc đến đích
//hiệu ứng giống như trong video studio
//chưa hình thành hết
void PR(int,int,int,int,int,char *xau="Quang",int tre=15 ,int mau=GREEN,int
cach=1);
void PR( int xgoc , int ygoc , int xdich , int ydich , char xau[] , int
tre , int mau , int cach)
{
    int i,j=0;
    textcolor(mau);
    if (xgoc==xdich&&ygoc>ydich&&cach==1)
    {
        for (j=0 ; j<=strlen(xau) ; j++)
        {
            if (xau[j]!=32)
                for (i=ygoc ; i>= ydich ; i--)
                {
                    gotoxy(xgoc,i);putch(xau[j]);
                    gotoxy(xgoc,i+1);putch(' ');
                    delay(tre);
                }
            else {gotoxy(xgoc,ydich);putch(' ');}
            xgoc++;
        }
    }

    if (ydich==ygoc&&xgoc>xdich&&cach==1)
    {
        for (j=0 ; j<=strlen(xau) ; j++)
        {
            if (xau[j]!=32)
                for (i=xgoc ; i>= xdich ; i--)
                {
                    gotoxy(i,ygoc);cprintf("%c ",xau[j]);
                    delay(tre);
                }
            else {gotoxy(xgoc,ydich);putch(' ');}
            xdich++;
        }
    }
}
}

```

Bài toán giải phương trình bậc nhất và giải phương trình bậc 2

PHP Code:

```

void bai101(void)
{
    float a,b;
    printf("Nhập vào số (thực) a : ");scanf("%f", &a);
    printf("Nhập vào số (thực) b : ");scanf("%f", &b);
    if (a==0)
        if (b>0) printf("Bất phương trình có nghiệm với mọi x");
        else printf("Bất phương trình vô nghiệm");
    if (a>0) printf("Bất phương trình có nghiệm là x > %.2f" ,1.0*(

```



```

-b)/a);
    if (a<0) printf("Bất phương trình có nghiệm là x < %.2f" ,1.0*(
-b)/a);
    }
void bai102(void)

    {
        float a,b,c,s;
        printf("Nhập vào số (thực) a : ");scanf("%f", &a);
        printf("Nhập vào số (thực) b : ");scanf("%f", &b);
        printf("Nhập vào số (thực) c : ");scanf("%f", &c);
        if (a==0)
            { if (b==0) if (c==0) printf("Phương trình có nghiệm với mọi
i x");
                else printf("Phương trình vô nghiệm");
                else printf("Phương trình có nghiệm duy nhất là x = %.2f"
, (-c)/b);
            }
        else
        {
            s=b*b-4*a*c;
            if (s<0) printf("Phương trình vô nghiệm");
            else if (s==0) printf("Phương trình có nghiệm kép %.2f" ,(-
b)/(2*a) );
            else printf("Phương trình có 2 nghiệm phân biệt là %.2f và
%.2f" , ((-b)+sqrt(s))/(2*a) ,1.0*((-b)-sqrt(s))/(2*a) );
        }
    }

```

Bài toán nhập a,b,c kiểm tra xem có phải tam giác hay ko

```

void bai103(void)
    {
        float a,b,c,s;
        printf("Nhập vào số (thực) a : ");scanf("%f", &a);
        printf("Nhập vào số (thực) b : ");scanf("%f", &b);
        printf("Nhập vào số (thực) c : ");scanf("%f", &c);
        if ((a+b>c) && (a+c>b) && (b+c>a))
            {
                printf("Đây là ba cạnh của một tam giác ");
                if ((a*a+b*b==c*c) || (a*a+c*c==b*b) || (c*c+b*b==a*a)) printf(
"vuông ");
                if ((a==b) && (b==c)) printf("đều ");
                else if ((a==b) || (b==c) || (c==a)) {printf("cân ");}
                s=(a+b+c)/2;printf("có :");
                printf("\n\tchu vi là %.2f và diện tích là %.2f", s*2 , sqr
t(s*(s-a)*(s-b)*(s-b) ) );
            }
        else printf("Đây không phải ba cạnh của một tam giác");
    }

```

Bài 105: Kiểm tra xem n có phải là số nguyên tố ko ?

Bài 106: Nhập phân số (nhập trực tiếp ví dụ : 3/9 , 4/8 , 1/2 ...). Đưa ra dạng tối giản của phân số đó.

Bài 107: Tìm ước chung lớn nhất , bội chung nhỏ nhất của 2 số nhập từ bàn phím.

Bài 108: Kiểm tra xem n có phải là số hoàn hảo ko ?

Bài 109: Liệt kê các số Amstrong nhỏ hơn hoặc bằng 999.

Bài 110: Tính tổng cách chữ số của số nguyên n.

Bài 111: Bài 18: Tính $n!$ và $n!!$.

Bài 112: Giải bài toán cổ vừa gà vừa chó.

Bài 113: a. Tìm min max của 2 số b. Nhập chiều cao h , cạnh đáy a . Tính diện tích tam giác

Bài 114: Nhập vào 1 số nguyên từ 0 đến 7. Viết ra màn hình thứ tương ứng trong tuần

Bài 115: Nhập 2 cạnh 1 hình chữ nhật. Tính S, P

Bài 116: Tính lũy thừa n của x

Bài 117: Giải bài toán cổ 100 trâu 100 cỏ

Bài 118: Nhập R của hình tròn. Tính S, P

Bài 119: Tính tổng các số số lẻ $> n$.

Bài 120: Kiểm tra 1 ký tự nhập từ bàn phím

Bài 121: Tính tổ hợp chập k của n phần tử

Code:

```
void bail05(void)
{
    int n;
    printf("Nhập vào số n : ");scanf("%d", &n);
    if (nguyento(n)) printf("Bạn vừa nhập vào 1 số nguyên tố.");
    else printf("%d không phải là số nguyên tố",n);
}
void bail06(void)
{
    int m,n;

    printf("Nhập vào một phân số :\n\tTừ số là : ");scanf("%d",&m)
;
    printf("\tMẫu số là : ");scanf("%d",&n);
    if (n!=0)
    {
        int a=UCLN(m,n);
        m/=a;n/=a;
        printf("Dạng tối giản của phân số đó là %d/%d ",m,n);
    }
}
void bail07(void)
{
    int m,n;
    printf("Nhập vào số (nguyên) m : ");scanf("%d", &m);
    printf("Nhập vào số (nguyên) n : ");scanf("%d", &n);
    printf("Ước chung lớn nhất của 2 số đó là : %d ",UCLN(m,n) );
    printf("\nBội chung nhỏ nhất của 2 số đó là : %d ",BCNN(m,n) );
}
void bail08(void)
{
    int m;
    printf("Nhập vào số m : ");scanf("%d", &m);
    if (hoanhao(m)) printf("Bạn vừa nhập vào một số hoàn hảo.");
    else printf("%d không phải là số hoàn hảo",m);
}
void bail09(void)
{
    int a,b,c;
    printf("Các số Amstrong nhỏ 999 là :");
    for (a=1 ; a<=9 ; a++)
```

```

        for (b=1 ; b<=9 ; b++)
            for (c=1 ; c<=9 ; c++)
                if (a*a*a+b*b*b+c*c*c==a*100+b*10+c)
                    printf("\n %d%d%d " , a,b,c );
    }
void bai110(void)
{
    int n,i,j,m;
    printf("Nhập vào số (nguyên) n : ");scanf("%d", &n);
    j=m=0;
    for (i=1 ; i<=12 ; i++ )
        {if (n==0) break;else {m+=n%10;n/=10;j++;}}
    printf("Số các chữ số của n là : %d \n" , j );
    printf("Tổng các chữ số của n là : %d" ,m );
}
void bai111(void)
{
    int n;
    printf("Nhập vào số (nguyên) n : ");scanf("%d", &n);
    printf(" %d! = %ld\n",n,giaithua(n));
    printf("%d!! = %ld",n,haicham(n));
}
void bai112(void)
{
    int i;
    for (i=1 ; i<=24 ; i++)
        if ((4*i+2*(36-i))==100)
            printf("Chó %d cho %d gà.", i, 36-i);
}
void bai113(void)
{
    float a,b,h;
    printf("Nhập vào số (thực) a : ");scanf("%f", &a);
    printf("Nhập vào số (thực) b : ");scanf("%f", &b);
    printf("Số lớn là : %.2f   số bé là : %.2f" , (a>b)?(a):(b) , (a
<b)?(a):(b));
    printf("\nNhập vào chiều cao : ");scanf("%f", &a);
    printf("Nhập vào cạnh đáy : ");scanf("%f", &b);
    printf("Diện tích tam giác là : %.2f" , a*b/2);
}

void bai114(void)
{
    int n;
    printf("Nhập vào số (nguyên) n : ");scanf("%d", &n);
    switch (n)
    { case 1:case 0:printf("Chủ nhật.");break;
      case 2:printf("Thứ hai.");break;
      case 3:printf("Thứ ba.");break;
      case 4:printf("Thứ tư.");break;
      case 5:printf("Thứ năm.");break;
      case 6:printf("Thứ sáu.");break;
      case 7:printf("Thứ bảy.");break;
      default :printf("Bạn nhập vào số ` 8 rồi");break;
    }
}

```

```

    }
void bail15(void)
{
    float a,b;
    printf("Nhập vào cạnh a : ");scanf("%f", &a);
    printf("Nhập vào cạnh b : ");scanf("%f", &b);
    printf("Chu vi của hình chữ nhật là : %.2f\n",a+b+a+b);
    printf("Diện tích của hình chữ nhật là : %.2f",a*b);
}
void bail16(void) //*****
{
    float x;
    int n;
    printf("Nhập vào số (thực) x : ");scanf("%f", &x);
    printf("Nhập vào số (nguyên) n : ");scanf("%d", &n);

    printf("x^n = %.0lf", pow(x,double(n)));
}
void bail17(void) //*****
{
    int a,b;
    for (a=1 ; a<=20 ; a++)
        for (b=1 ; b<=33 ; b++)
            if ( a*5 + b*3 + (100-a-b)/3 == 100 )
                printf("%d trâu đực , %d trâu nài , %d trâu già.\n"
,a ,b,100-a-b);
}
void bail18(void)
{
    float r;
    printf("Nhập vào bán kính đường tròn : ");
    scanf("%f",&r);
    printf("\nChu vi của đường tròn là : %.2f" , 3.14*r);
    printf("\nDiện tích của đường tròn là : %.2f" , 3.14*r*r);
}
void bail19(void)
{
    int i,m=0,n;
    printf("Nhập vào số (nguyên) n : ");
    scanf("%d", &n);
    for (i=0; i<=n ; i++) if (i&1) m+=i;
    gotoxy(5,23);printf("Tổng các số lẻ " n la %d",m);
}
void bail20(void)
{
    char ch;
    printf("Nhập vào một kí tự.\n");
    ch=getch();
    if (ch==32) printf("Bạn vừa nhập vào một dấu cách.");
    else if (isalpha(ch)) printf("Bạn vừa nhập vào một chữ cái %c "
, ch);
    else if (isdigit(ch)) printf("Bạn vừa nhập vào một chữ số %c "
, ch);
    else printf("Bạn vừa nhập vào một kí tự đặc biệt : %c " , ch);
}
void bail21(void)

```

```

{
    int k,n;
    printf("Nhập vào số (nguyên) k : ");scanf("%d", &k);
    printf("Nhập vào số (nguyên) n : ");scanf("%d", &n);
    printf("Tổ hợp chập k của n phần tử : %d" ,tohop(k,n));
}

```

Một bài ví dụ về Hàm tạo và class

Tính Tổng , Hiệu ,Thương , Tích 2 phân số

```

#include<iostream>
using namespace std;

class PhanSo
{
public:
    PhanSo() // Ham khai tao gt mac dinh
    {
        tu = mau = 1;
    }
    PhanSo(int tu1, int mau1) // Ham tao de nhap vao phan so
    {
        tu = tu1;
        mau = mau1;
    }
    void rutGon(); // Ham rutgon phan so
    PhanSo cong(PhanSo p); // Cong
    PhanSo tru(PhanSo p); // Tru
    PhanSo nhan(PhanSo p); // Nhan
    PhanSo chia(PhanSo p); // Chia

    void output() // In ket qua va phan so ra man hinh
    {
        cout << tu << "/" << mau << endl;
    }

private:
    int tu;
    int mau;
};

PhanSo PhanSo::cong(PhanSo p) // Dinh nghia ham Cong phan so
{
    PhanSo kq;
    kq.tu = tu * p.mau + mau * p.tu; // Quy dong phan so
    kq.mau = mau * p.mau;
    kq.rutGon(); // Rut gon phan so Tong
    return kq; // Roi tra ve phan so
}

PhanSo PhanSo::tru(PhanSo p)
{
    PhanSo kq;
    kq.tu = tu * p.mau - mau * p.tu;
    kq.mau = mau * p.mau;
    kq.rutGon();
    return kq;
}

PhanSo PhanSo::nhan(PhanSo p)

```

```

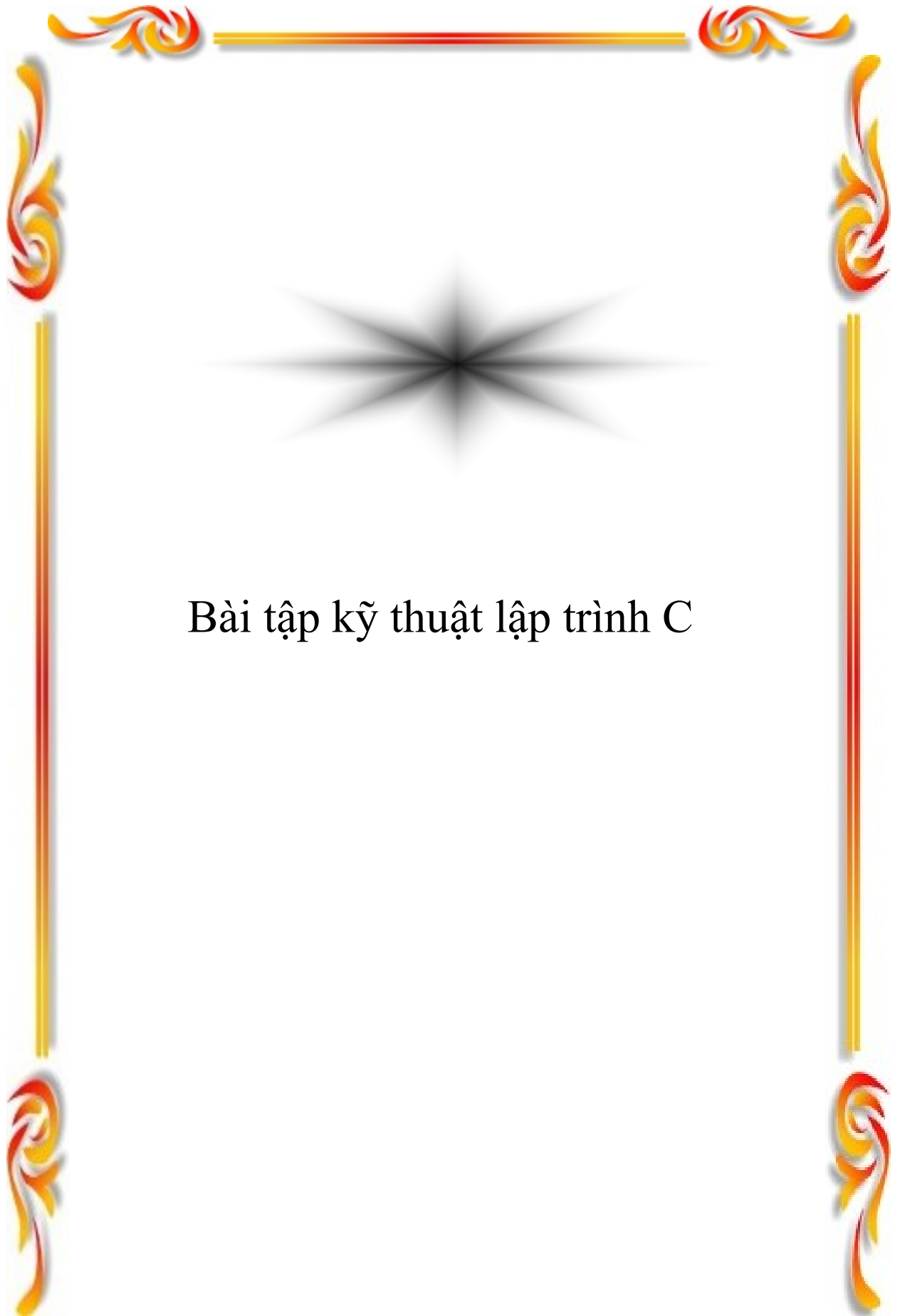
{
    PhanSo kq;
    kq.tu = tu * p.tu;
    kq.mau = mau * p.mau;
    kq.rutGon();
    return kq;
}

PhanSo PhanSo::chia(PhanSo p)
{
    PhanSo kq;
    kq.tu = tu * p.mau;
    kq.mau = mau * p.tu;
    kq.rutGon();
    return kq;
}

void PhanSo::rutGon() // Ham rut gon Phan so + Tim uoc so chung lon n
hat cua tu va mau
{
    int a = abs(tu);
    int b = abs(mau); // Tim uoc so chung lon nha
    while(a != b) //
        if(a > b) a = a - b; //
        else b = b - a;
    tu = tu / a;
    mau = mau / a;
}

int main()
{
    PhanSo p1(1, 3), p2(1, 6); // Tao 2 phan so p1 (1/3) va p2 (1/6)
    cout << "Phan so p1:";
    p1.output();
    cout << "Phan so p2:";
    p2.output();
    PhanSo tong, hieu, tich, thuong;
    tong = p1.cong(p2);
    hieu = p1.tru(p2);
    tich = p1.nhan(p2);
    thuong = p1.chia(p2);
    cout << "Tong hai phan so:";
    tong.output();
    cout << "Hieu hai phan so:";
    hieu.output();
    cout << "Tich hai phan so:";
    tich.output();
    cout << "Thuong hai phan so:";
    thuong.output();
    system("pause>nul");
    return 0;
}

```



Bài tập kỹ thuật lập trình C

LỜI MỞ ĐẦU



Khi bắt đầu làm quen với ngôn ngữ lập trình – Cụ thể là ngôn ngữ C – Sinh Viên thường gặp khó khăn trong việc chuyển vấn đề lý thuyết sang cài đặt cụ thể trên máy. Sách “**Giáo Trình Bài Tập Kỹ Thuật Lập Trình**” nhằm cung cấp cho các Học Sinh - Sinh Viên Trường CĐ Công Nghệ Thông Tin Tp. Hồ Chí Minh hệ thống các bài tập, những **kỹ năng thực hành cơ bản và nâng cao về ngôn ngữ lập trình C**. Cuốn sách này được xem như tài liệu hướng dẫn từng bước cho Học Sinh - Sinh Viên của Trường trong việc học và áp dụng kiến thức lý thuyết trên lớp một cách thành thạo và sâu rộng.

Giáo trình được chia thành **10 chương** theo từng nội dung kiến thức, kèm theo **Các đề thi mẫu** và **1 phụ lục** hướng dẫn viết chương trình, chuẩn đoán lỗi và sửa lỗi. Mỗi chương gồm 2 phần:

- ❖ **Phần lý thuyết:** được tóm tắt ngắn gọn với đầy đủ ví dụ minh họa kèm theo.
- ❖ **Phần bài tập:** với nhiều bài tập được chia làm hai mức độ cơ bản và luyện tập nâng cao, bài tập có đánh dấu * là bài tập khó dành cho sinh viên luyện tập thêm.
- ❖ **Phần kết luận:** Tóm tắt nội dung và các thao tác mà sinh viên cần nắm hay những lưu ý của chương đó.

Trong quá trình biên soạn, chúng tôi đã cố gắng trích lọc những kiến thức rất cơ bản, những lỗi hay gặp đối với người mới lập trình. Bên cạnh đó chúng tôi cũng bổ sung thêm một số bài tập nâng cao để rèn luyện thêm kỹ năng lập trình.

Tuy nhiên, chủ đích chính của giáo trình này là phục vụ cho một môn học nên chắc chắn không thể tránh khỏi những thiếu sót, vì thế, rất mong nhận được những góp ý quý báu của các thầy cô, các đồng nghiệp và các bạn Học Sinh – Sinh Viên để giáo trình này ngày càng hoàn thiện hơn.

Chân thành cảm ơn.

LỊCH TRÌNH THỰC HÀNH



Tổng thời gian: 90 tiết.

STT	NỘI DUNG	SỐ TIẾT
1	Chương 1: Lưu đồ thuật toán	03
2	Chương 2: Cấu trúc điều khiển	06
3	Chương 3: Hàm con	12
4	Chương 4: Mảng một chiều	24
5	Chương 5: Chuỗi ký tự	06
6	Chương 6: Mảng hai chiều	12
7	Chương 7: Kiểu dữ liệu có cấu trúc	12
8	Chương 8: Tập tin	06
9	Chương 9: Đệ qui	06
10	Chương 10: Hướng dẫn lập trình bằng phương pháp Project	03

CHƯƠNG 1 LƯU ĐỒ THUẬT TOÁN (FLOWCHART)

Các ký hiệu biểu diễn lưu đồ thuật toán, cách biểu diễn các cấu trúc điều khiển rẽ nhánh, cấu trúc lặp và các kỹ thuật liên quan đến lưu đồ thuật toán.

I. TÓM TẮT LÝ THUYẾT

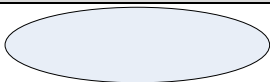


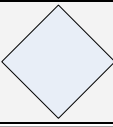

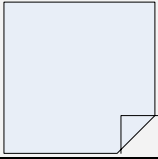

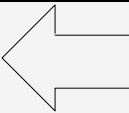

I.1. Khái niệm

Lưu đồ thuật toán là công cụ dùng để **biểu diễn thuật toán**, việc mô tả **nhập** (input), dữ liệu **xuất** (output) và luồng xử lý thông qua các **ký hiệu hình học**.

I.2. Phương pháp duyệt

- Duyệt từ trên xuống.
- Duyệt từ trái sang phải.

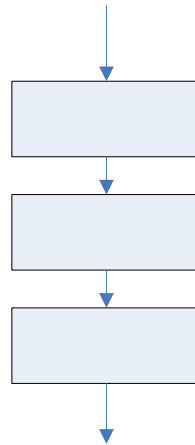
I.3. Các ký hiệu

STT	KÝ HIỆU	DIỄN GIẢI
1		Bắt đầu chương trình
2		Kết thúc chương trình
3		Luồng xử lý
4		Điều khiển lựa chọn
5		Nhập
6		Xuất
7		Xử lý, tính toán hoặc gán
8		Trả về giá trị (return)
9		Điểm nối liên kết tiếp theo (Sử dụng khi lưu đồ vượt quá trang)

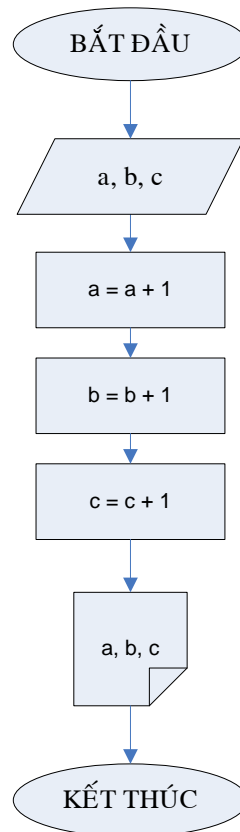
I.4. Các cấu trúc điều khiển cơ bản

a. Cấu trúc tuần tự

Tuần tự thực thi tiến trình. Mỗi lệnh được thực thi theo một chuỗi **từ trên xuống**, xong lệnh này rồi **chuyển xuống** lệnh kế tiếp.



Ví dụ: Nhập vào 3 số nguyên a, b, c và xuất ra màn hình với giá trị của mỗi số tăng lên 1.

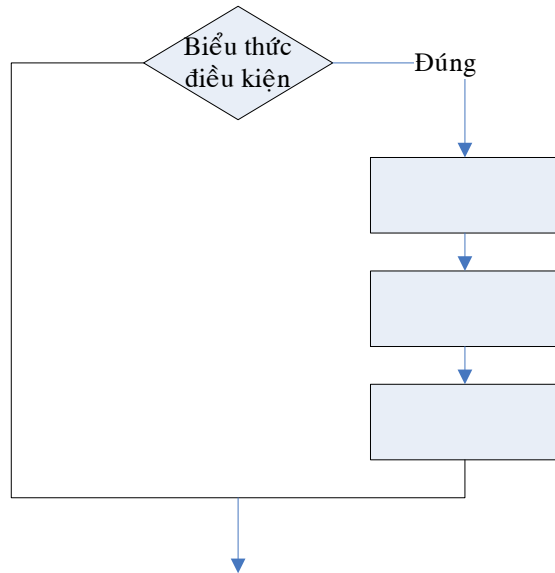


b. Cấu trúc lựa chọn

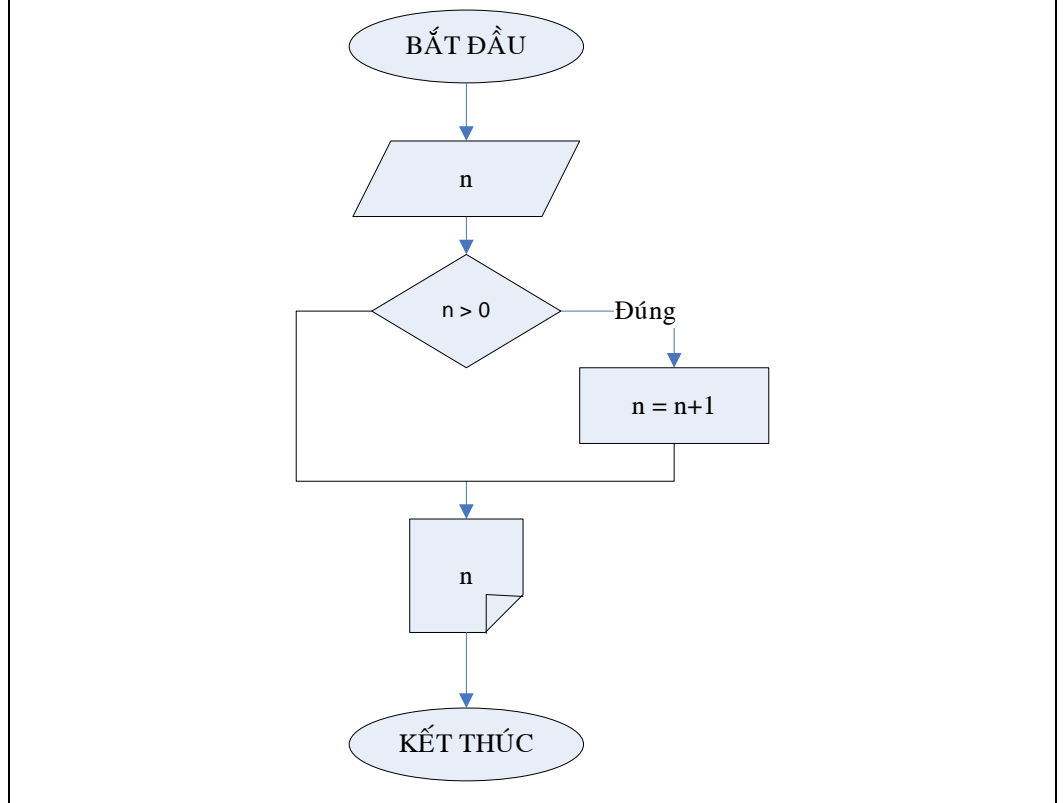
Điểm quyết định cho phép **chọn một trong hai trường hợp**.

- **if**

Chỉ xét trường hợp đúng.

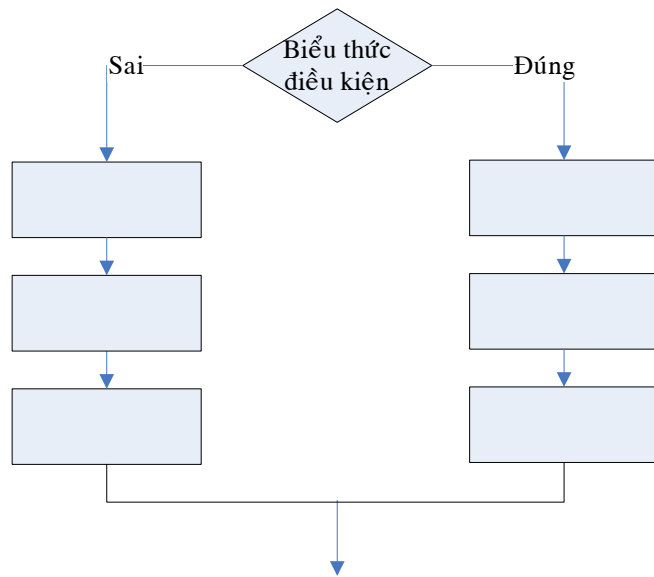


Vi dụ: Nhập vào số nguyên n. Kiểm tra nếu $n > 0$ tăng n lên 1 đơn vị. Xuất kết quả.

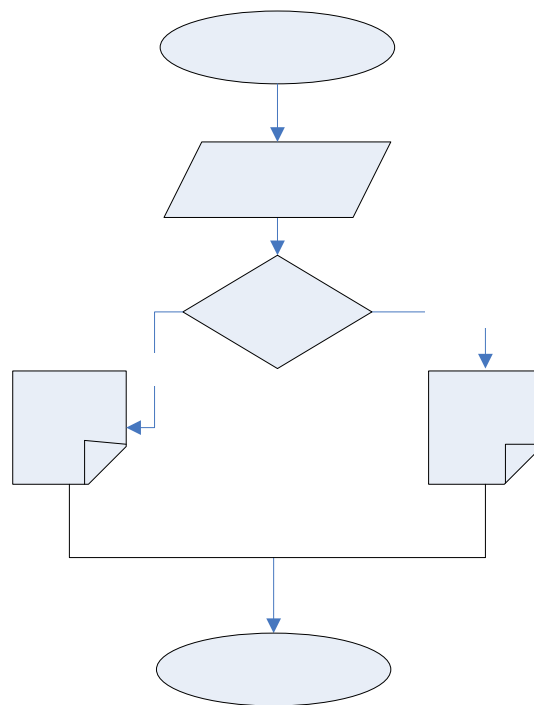


- **if...else**

Xét trường hợp đúng và trường hợp sai.



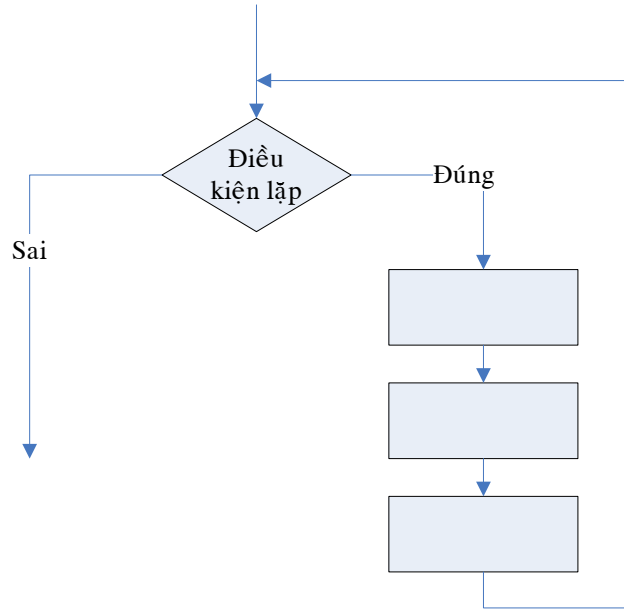
Ví dụ: Nhập vào số nguyên n. Kiểm tra nếu n chẵn xuất ra màn hình “n chẵn”, ngược lại xuất “n lẻ”.



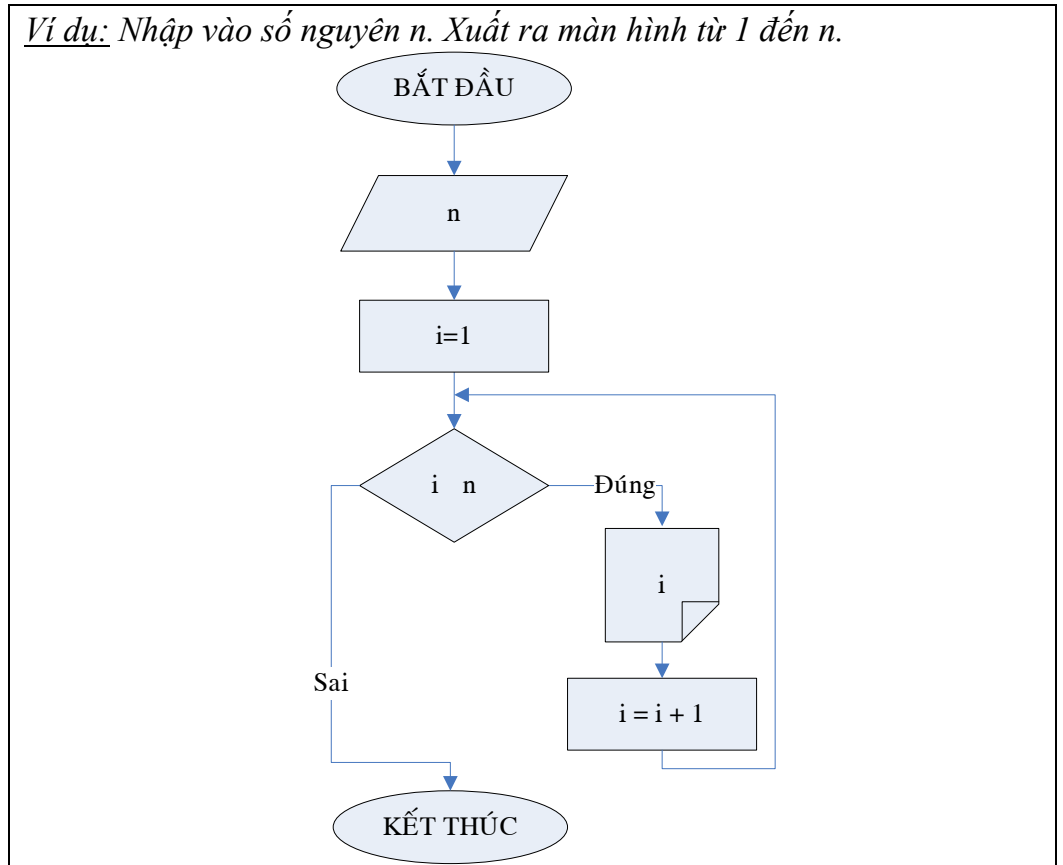
c. Cấu trúc lặp

Thực hiện liên tục 1 lệnh hay tập lệnh với số lần lặp dựa vào điều kiện. Lặp sẽ kết thúc khi điều kiện được thỏa.

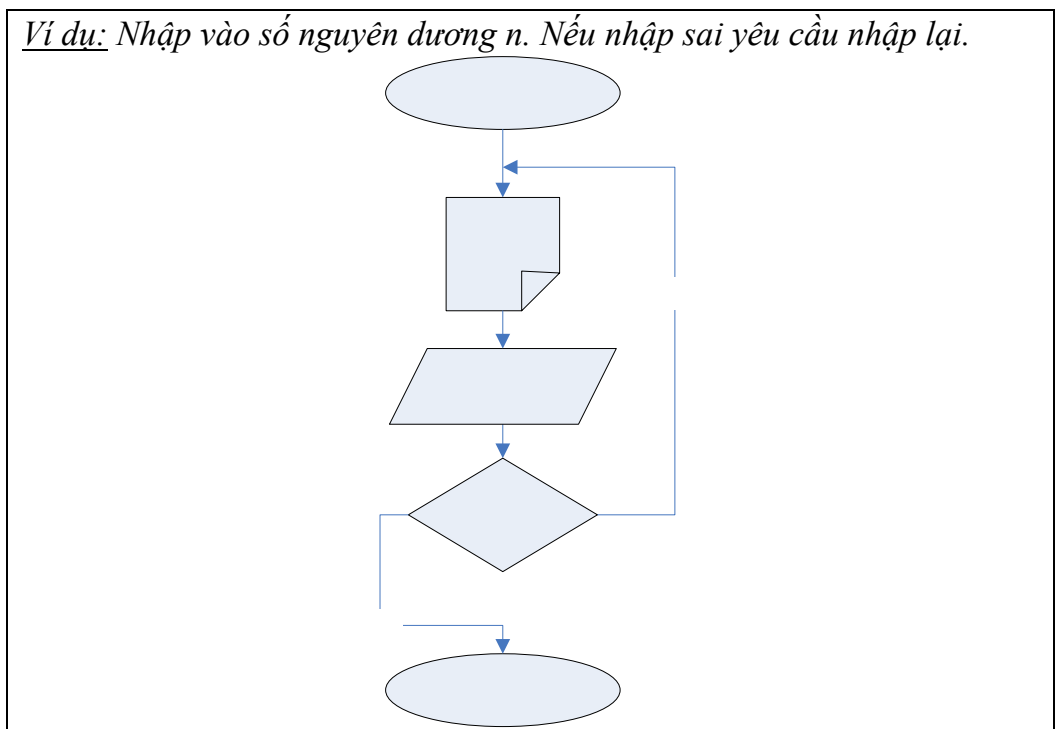
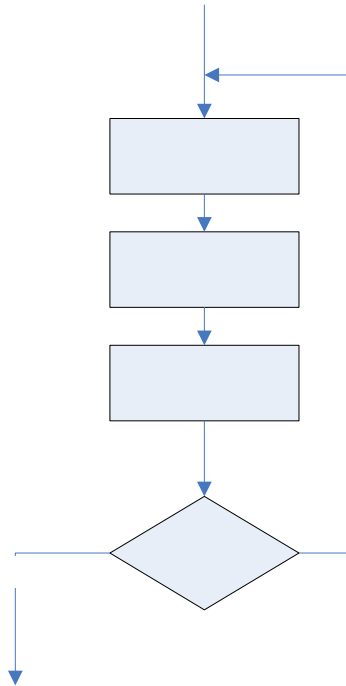
- **for / while (Kiểm tra điều kiện trước khi lặp)**



Ví dụ: Nhập vào số nguyên n. Xuất ra màn hình từ 1 đến n.

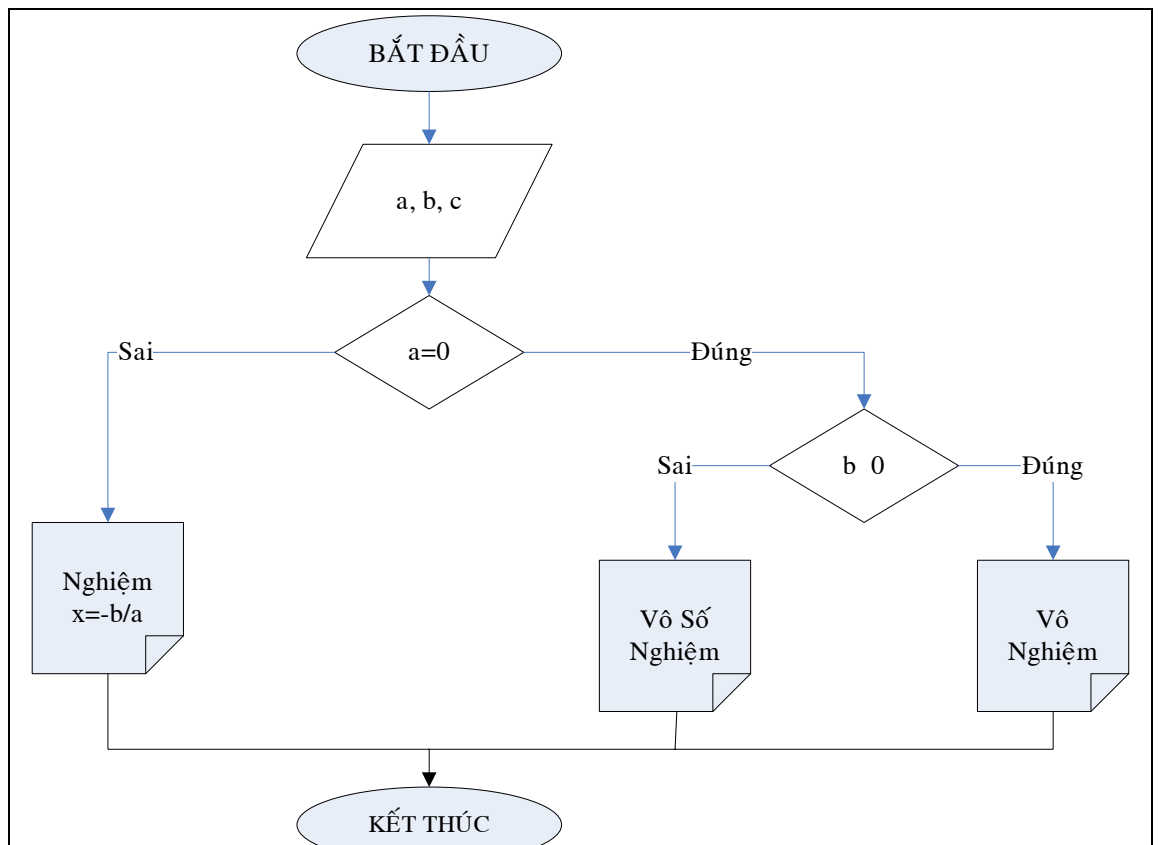


- do ... while (Thực hiện lặp trước khi kiểm tra điều kiện)

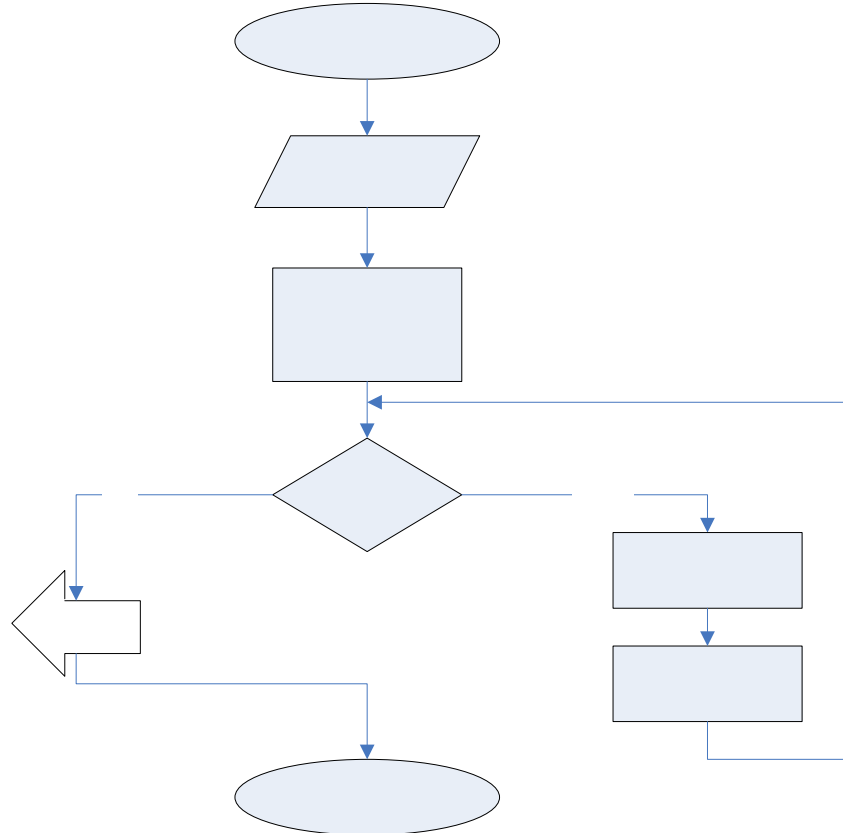


d. Các ví dụ

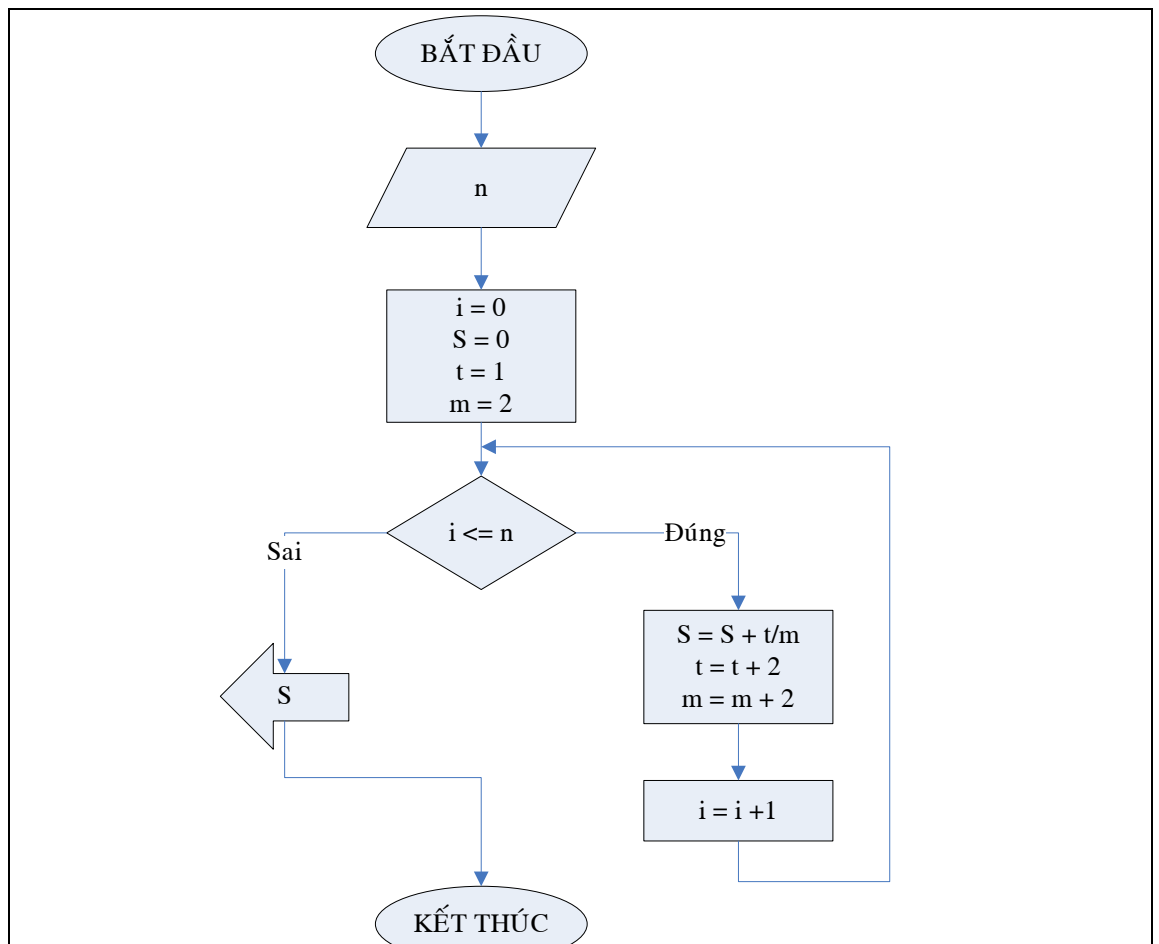
Ví dụ 1: Giải và biện luận phương trình: $ax+b=0$.



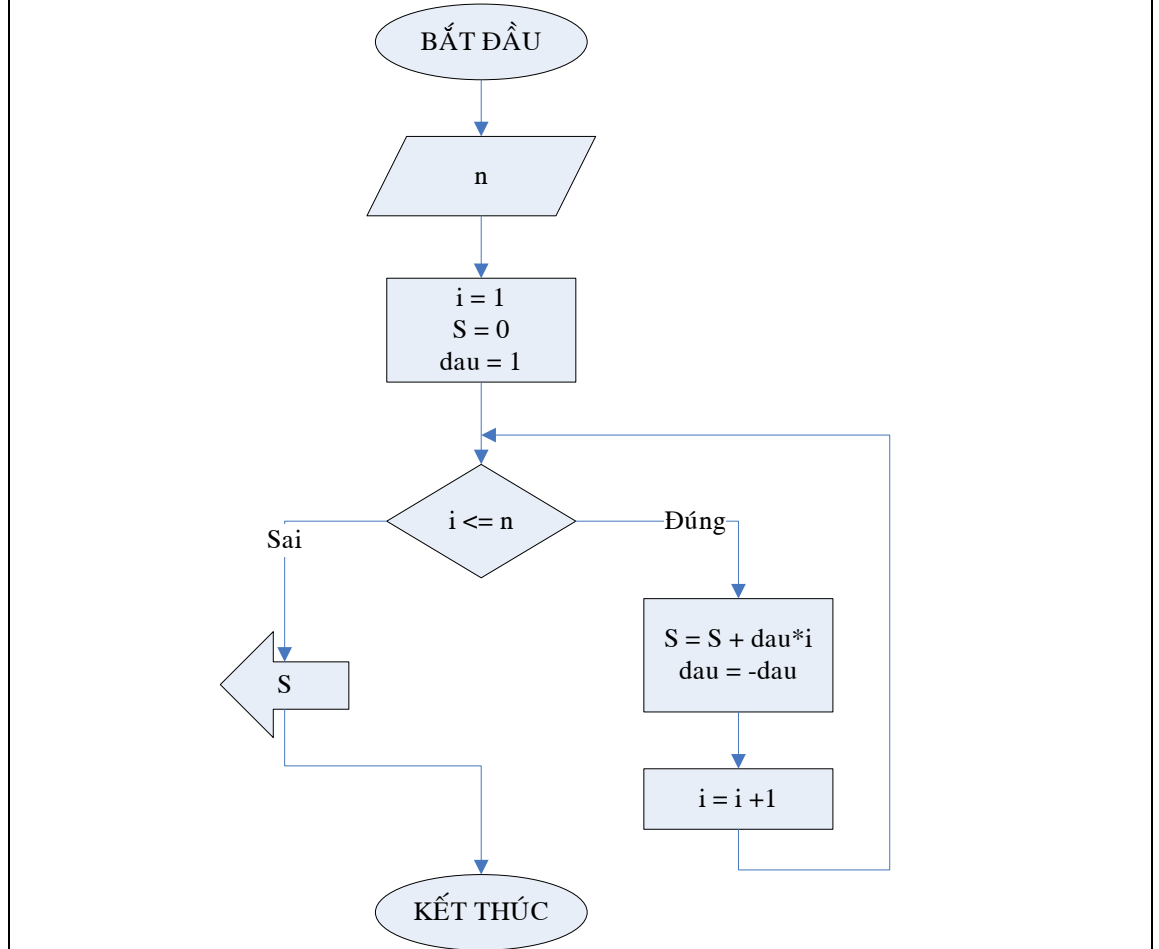
Ví dụ 2: Tính tổng: $S = 1 + 2 + 3 + \dots + n$, với $n > 0$



Ví dụ 3: Tính tổng: $S(n) = \frac{1}{2} + \frac{3}{4} + \frac{5}{6} + \dots + \frac{2n+1}{2n+2}$, với $n > 0$



Ví dụ 4: Tính tổng: $S(n) = 1 - 2 + 3 - 4 + \dots + (-1)^{n+1}n$, với $n > 0$



II. BÀI TẬP

Vẽ lưu đồ thuật toán sau

II.1. Bài tập cơ bản

1. Nhập vào hai số x, y . Xuất ra màn hình tổng, hiệu, tích, thương của hai số trên.
2. Nhập vào số nguyên n , kiểm tra xem n chẵn hay lẻ và xuất ra màn hình.
3. Nhập vào ba cạnh a, b, c của tam giác. Xuất ra màn hình tam giác đó thuộc loại tam giác gì? (Thường, cân, vuông, đều hay vuông cân).
4. Nhập vào số nguyên n . Xuất ra n màn hình (Nếu n chẵn thì gấp đôi giá trị).
5. Nhập vào số nguyên n . Nếu $n > 5$ thì tăng n lên 2 đơn vị và trả về giá trị n , ngược lại trả về giá trị 0.
6. Tính $n!$, với $n \geq 0$
7. Tính $P(n) = 1.3.5 \dots (2n+1)$, với $n \geq 0$
8. Tính $S(n) = 1 + 3 + 5 + \dots + (2 \times n + 1)$, với $n \geq 0$
9. Tính $S(n) = 1 - 2 + 3 - 4 + \dots + (-1)^{n+1} n$, với $n > 0$
10. Tính $S(n) = 1 + 1.2 + 1.2.3 + \dots + 1.2.3 \dots n$, với $n > 0$
11. Tính $S(n) = 1^2 + 2^2 + 3^2 + \dots + n^2$, với $n > 0$
12. Tính $S(n) = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$, với $n > 0$
13. (*) Tính $S(n) = 1 + \frac{1}{1+2} + \frac{1}{1+2+3} + \dots + \frac{1}{1+2+3+\dots+n}$, với $n > 0$
14. Tính $P(x, y) = x^y$.
15. Tính $S(n) = 1 + (1+2) + (1+2+3) + \dots + (1+2+3+\dots+n)$, với $n > 0$
16. Cho số nguyên n . Tính trị tuyệt đối của n .
17. Cho số nguyên dương n gồm k chữ số. Tìm chữ số có giá trị lớn nhất.
18. Đếm số lượng ước số chẵn của số nguyên dương n .
19. In ra chữ số đầu tiên của số nguyên dương n gồm k chữ số.
20. Cho 2 số nguyên dương a, b . Tìm USCLN của a và b .
21. Cho 2 số nguyên dương a, b . Tìm BSCNN của a và b .
22. Cho số nguyên dương x . Kiểm tra xem x có phải là số nguyên tố không?
23. Cho số nguyên dương x . Kiểm tra x có phải là số chính phương không?
24. Cho số nguyên dương x . Kiểm tra xem x có phải là số hoàn thiện không?

II.2. Bài tập luyện tập và nâng cao

25. Tính $S(n) = 1 + 2^2 + 3^3 + \dots + n^n$, với $n \geq 0$

26. Tính $S(n) = \frac{1}{2} + \frac{2}{3} + \frac{3}{4} + \dots + \frac{n}{n+1}$, với $n > 0$

27. Tính $S(n) = 1 + \frac{1}{2!} + \frac{1}{3!} + \dots + \frac{1}{n!}$, với $n > 0$

28. Tính $S(n) = 1 + \frac{1+2}{2!} + \frac{1+2+3}{3!} + \dots + \frac{1+2+3+\dots+n}{n!}$, với $n > 0$

29. Giải và biện luận phương trình: $ax^2 + bx + c = 0$

30. Giải và biện luận phương trình: $ax^4 + bx^2 + c = 0$

31. (*) Tính $S(n) = \sqrt{n + \sqrt{(n-1) + \sqrt{(n-2) + \dots + \sqrt{1}}}}$, với $n > 0$

32. (***) Tính $S(n) = \sqrt{1 + \sqrt{2 + \sqrt{3 + \dots + \sqrt{n}}}}$, với $n > 0$

III. KẾT LUẬN

Lưu đồ thuật toán rất hữu ích trong việc mô tả cách giải quyết của một bài toán. Việc mô tả này rất trực quan thông qua các ký hiệu hình học, đây là giai đoạn đầu tiên trước khi bắt tay vào lập trình trên một ngôn ngữ lập trình cụ thể.

Khi xây dựng lưu đồ thuật toán, chúng ta cần chú ý một vài điểm sau:

- ❖ Một lưu đồ phải có điểm **bắt đầu** và điểm **kết thúc** (điều kiện kết thúc).
- ❖ Phải có **dữ liệu vào**, **dữ liệu ra** sau khi xử lý tính toán.
- ❖ Tại mỗi vị trí quyết định lựa chọn rẽ nhánh phải ghi rõ điều kiện **đúng hoặc sai** thì đi theo nhánh nào.

CHƯƠNG 2 **CẤU TRÚC ĐIỀU KHIỂN**

Tìm hiểu và cài đặt các cấu trúc rẽ nhánh, lựa chọn, lặp và các ký hiệu phép toán trong ngôn ngữ C. Mô tả cách hoạt động và hướng dẫn chạy từng bước chương trình.

I. TÓM TẮT LÝ THUYẾT

I.1. Các ký hiệu

STT	KÝ HIỆU	DIỄN GIẢI	VÍ DỤ
1	{ }	Bắt đầu và kết thúc hàm hay khối lệnh.	void main() { }
2	;	Kết thúc khai báo biến, một lệnh, một lời gọi hàm, hay khai báo nguyên mẫu hàm.	int x; void NhapMang(int a[], int &n);
3	//	Chú thích (ghi chú) cho một dòng. Chỉ có tác dụng đối với người đọc chương trình.	//Ham nay dung de nhap mang void NhapMang(int a[], int &n);
4	/* */	Tương tự như ký hiệu //, nhưng cho trường hợp nhiều dòng.	/* Dau tien nhap vao n. Sau do nhap cho tung phan tu */ void NhapMang(int a[], int &n);

I.2. Các kiểu dữ liệu cơ bản trong C

STT	KIỂU	GHI CHÚ	KÍCH THƯỚC	ĐỊNH DẠNG
KIỂU LIÊN TỤC (SỐ THỰC)				
1	float		4 bytes	%f
2	double		8 bytes	%lf
3	long double		10 bytes	%lf
KIỂU RỜI RẠC (SỐ NGUYÊN)				
1	char	Ký tự	1 byte	%c
		Số nguyên	1 byte	%d
2	unsigned char	Số nguyên dương	1 byte	%d
3	int	Số nguyên	2 bytes	%d
4	unsigned int	Số nguyên dương	2 bytes	%u
5	long	Số nguyên	4 bytes	%ld
6	unsigned long	Số nguyên dương	4 bytes	%lu
7	char *	Chuỗi		%s

I.3. Bảng ký hiệu các phép toán

STT	PHÉP TOÁN	Ý NGHĨA	GHI CHÚ
PHÉP TOÁN SỐ HỌC			
1	+	Cộng	
2	-	Trừ	
3	*	Nhân	
4	/	Chia lấy phần nguyên	
5	%	Chia lấy phần dư	
PHÉP TOÁN QUAN HỆ			
1	>	Lớn hơn	
2	<	Nhỏ hơn	
3	>=	Lớn hơn hoặc bằng	
4	<=	Nhỏ hơn hoặc bằng	
5	==	Bằng nhau	
6	!=	Khác nhau	
PHÉP TOÁN LOGIC			
1	!	NOT	
2	&&	AND	
3		OR	
TOÁN TỬ TĂNG GIẢM			
1	++	Tăng 1	Nếu toán tử tăng giảm đặt trước thì tăng giảm trước rồi tính biểu thức hoặc ngược lại.
2	--	Giảm 1	
PHÉP TOÁN THAO TÁC TRÊN BIT			
1	&	AND	
2		OR	
3	^	XOR	
4	<<	Dịch trái	
5	>>	Dịch phải	
6	~	Lấy phần bù theo bit	

I.4. Các hàm cơ bản

STT	TÊN HÀM	THƯ VIỆN	DIỄN GIẢI	VÍ DỤ
1	printf	#include<stdio.h>	Xuất ra màn hình.	<pre>#include<stdio.h> #include<conio.h> #include<dos.h> void main() { int c = 1, n; clrscr(); printf("Nhap n:"); scanf("%d", &n); do{ textcolor(c); gotoxy(20, 10); cprintf("%d", n); c++; if (c>15) c = 1; delay(200); } while(!kbhit()); }</pre>
2	scanf	#include<stdio.h>	Lấy dữ liệu từ bàn phím.	
3	gotoxy	#include<conio.h>	Di chuyển dấu nháy đến tọa độ (x, y) trên màn hình văn bản.	
4	textcolor	#include<conio.h>	Đặt màu cho chữ (có giá trị từ 0 đến 15).	
5	cprintf	#include<stdio.h>	Xuất ra màn hình với màu chữ đã định liền trước đó.	
6	delay	#include<dos.h>	Dừng thực hiện lệnh tiếp sau một khoảng thời gian.	
7	kbhit	#include<conio.h>	Kiểm tra xem có nhấn phím.	

I.5. Cấu trúc rẽ nhánh

a. Cấu trúc if

```
if (biểu thức điều kiện)
{
    <khối lệnh> ;
}
```

Nếu biểu thức điều kiện cho kết quả khác không thì thực hiện khối lệnh.

Ví dụ:

```
#include <conio.h>
#include <stdio.h>
void main ()
{
    float number ;

    printf ( "Nhap mot so trong khoang tu 1 den 10 => " );
    scanf ( "%f", &number );
    if (number >5)
        printf ( "So ban nhap lon hon 5. \n" );
    printf ( "%f la so ban nhap. ", number );
}
```

b. Cấu trúc if ... else

```

if (biểu thức điều kiện)
{
    <khối lệnh 1>;
}
else
{
    <khối lệnh 2>;
}

```

Nếu biểu thức điều kiện cho kết quả khác không thì thực hiện khối lệnh 1, ngược lại thì cho thực hiện khối lệnh thứ 2. Biểu thức điều kiện phải đặt trong cặp dấu ngoặc tròn.

Ví dụ: Giải và biện luận phương trình: $ax+b=0$

```

#include <conio.h>
#include <stdio.h>
void main ()
{
    float a, b;
    printf ( "\n Nhap vao a:");
    scanf ( "%f", &a);
    printf ( " Nhap vao b:");
    scanf ( "%f", &b) ;
    if (a== 0)
        if (b== 0)
            printf ( " \n PTVSN");
        else
            printf ( " \n PTVN");
    else
        printf ( " \n Nghiem x=%f", -b/a);
    getch ();
}

```

I.6. Cấu trúc lựa chọn switch

```

switch (biểu thức)
{
    case n1:
        các câu lệnh ;
        break ;
    case n2:
        các câu lệnh ;
        break ;
    .....
    case nk:
        <các câu lệnh> ;
        break ;
}

```

```
[default: các câu lệnh]
```

```
}
```

- n_i là các **hằng số nguyên hoặc ký tự**.
- Phụ thuộc vào giá trị của biểu thức viết sau **switch**, nếu:
 - Giá trị này = n_i thì thực hiện câu lệnh sau case n_i .
 - Khi giá trị biểu thức không thỏa tất cả các n_i thì thực hiện câu lệnh sau **default** nếu có, hoặc thoát khỏi câu lệnh **switch**.
 - Khi chương trình đã thực hiện xong câu lệnh của **case** n_i nào đó thì nó sẽ thực hiện luôn các lệnh thuộc **case** bên dưới nó mà không xét lại điều kiện (do các n_i được xem như các nhãn) → Vì vậy, để chương trình thoát khỏi lệnh **switch** sau khi thực hiện xong một trường hợp, ta dùng lệnh **break**.

Ví dụ: Tạo menu cấp 1 cho phép chọn menu bằng số nhập từ bàn phím.

```
#include<stdio.h>
#include<conio.h>
int ChonTD ()
{
    int chon ;

    printf ("Thuc Don") ;
    printf ("\n1. Lau thai!") ;
    printf ("\n2. Nuoc ngot!") ;
    printf ("\n3. Ca loc hap bau!") ;
    printf ("\n4. Chuot dong!") ;
    printf ("\n Xin moi ban chon mon an!") ;
    scanf ("%d",&chon) ;
    return chon ;
}

void TDchon(int chon)
{
    switch (chon)
    {
        case 1:
            printf ("\nBan chon lau thai!") ;
            break ;
        case 2:
            printf ("\nBan chon nuoc ngot!") ;
            break ;
        case 3:
            printf ("\nBan chon ca loc hap bau!") ;
            break ;
        case 4:
            printf ("\nBan chon chuot dong!") ;
```



```

        break ;
    default:
        printf ("\nBan chon khong dung!");
    }
}

void main()
{
    clrscr() ;
    int c ;
    c=ChonTD() ;
    TDchon(c) ;
    getch() ;
}

```

I.7. Cấu trúc lặp

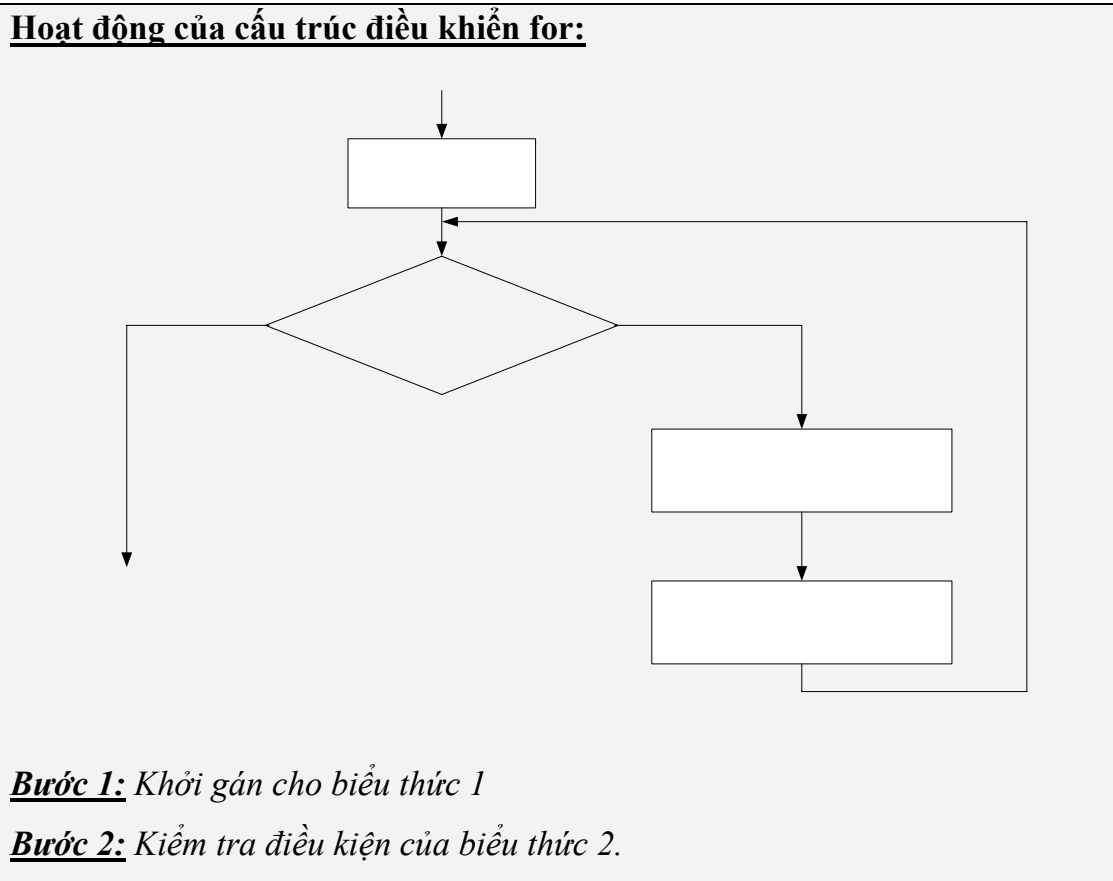
a. for

```

for (<biểu thức khởi gán>; <biểu thức điều kiện>; <biểu thức tăng/giảm>)
{
    <khối lệnh>;
}

```

Bất kỳ biểu thức nào trong 3 biểu thức nói trên đều có thể vắng nhưng phải giữ dấu chấm phẩy (;).



- Nếu **biểu thức 2** $\neq 0$ thì cho thực hiện các lệnh của vòng lặp, thực hiện biểu thức 3. Quay trở lại bước 2.
- Ngược lại thoát khỏi lặp.

Ví dụ: In ra màn hình bảng mã ASCII từ ký tự số 33 đến 255.

```
#include<conio.h>
#include<stdio.h>
void main()
{
    for (int i=33;i<=255;i++)
        printf("Ma ASCII cua %c: %d\t", i, i) ;
    getch () ;
}
```

b. while

```
< Khởi gán>
while ( <biểu thức điều kiện> )
{
    lệnh/ khối lệnh;
    < tăng/giảm chỉ số lặp>;
}
```

🔗 **Lưu ý:** Cách hoạt động của **while** giống **for**

Ví dụ: Tính giá trị trung bình các chữ số của số nguyên n gồm k chữ số.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    long n, tong=0;
    int sochuso=0;
    float tb;

    printf ("Nhap vao gia tri n gom k chu so" );
    scanf ("%ld",&n) ;
    while(n>0)
    {
        tong=tong+n%10 ;
        sochuso++ ;
        n=n/10 ;
    }

    tb=1.0*tong/sochuso ;
    printf ("Gia tri trung binh la: %f", tb) ;
}
```

```

    getch () ;
}

```

c. *do ... while*

```

do
{
    < khối lệnh> ;
} while (biểu thức điều kiện) ;

```

Thực hiện khối lệnh cho đến khi biểu thức có giá trị bằng 0.

Ví dụ: Nhập ký tự từ bàn phím hiển thị lên màn hình mã ASCII của ký tự đó, thực hiện đến khi nhấn phím ESC (Mã ASCII của phím ESC là 27).

```

#include<stdio.h>
#include<conio.h>

void main()
{
    int ma ;
    do {
        ma=getch ();
        if (ma !=27)
            printf ("Ma ASCII %c:%d\t", ma, ma);
    }while (ma!=27) ;

    getch () ;
}

```

⊗ Lặp **while** kiểm tra điều kiện trước khi thực hiện lặp, còn vòng lặp **do...while** thực hiện lệnh lặp rồi mới kiểm tra điều kiện. Do đó vòng lặp **do...while** thực hiện lệnh ít nhất một lần.

I.8. break và continue

a. *break*

Dùng để kết thúc vòng lặp trực tiếp chứa nó khi thỏa điều kiện nào đó.

Ví dụ: Cho phép người dùng nhập liên tục giá trị n cho đến khi nhập âm thì dừng.

```

#include<stdio.h>
#include<conio.h>

void main()
{

```

```

while (1)
{
    printf("\nNhập n: ");
    scanf("%d", &n);
    if(n<0)
        break;
}

getch ();
}

```

b. *continue*

Dùng để bỏ qua một lần lặp khi thỏa điều kiện nào đó.

Ví dụ: In ra màn hình giá trị từ 10 đến 20 trừ đi số 13 và số 17.

```

#include<stdio.h>
#include<conio.h>

void main()
{
    for(int i=10 ; i<=20; i++)
    {
        if(i==13||i==17)
            continue;
        printf("%d\t", i);
    }
    getch ();
}

```

II. BÀI TẬP

II.1. Phương pháp chạy tay từng bước để tìm kết quả chương trình

- ❖ Xác định chương trình có sử dụng những biến nào.
- ❖ **Giá trị ban đầu** của mỗi biến.
- ❖ Những **biến nào sẽ bị thay đổi** trong quá trình chạy chương trình thì lập thành bảng có dạng sau:

Bước (Hoặc lần thực hiện)	Biến 1	Biến 2	...	Biến n	Kết quả in ra màn hình
0	Giá trị 0	Giá trị 0	...	Giá trị 0	
1	Giá trị 1	Giá trị 1	...	Giá trị 1	
2	Giá trị 2	Giá trị 2	...	Giá trị 2	
...	
...	

☞ *Lưu ý từng lệnh và biểu thức điều kiện trong đoạn chương trình*

Ví dụ: Cho biết kết quả của đoạn chương trình sau:

```
void main()
{
    int i, a = 4;
    clrscr();
    for(i = 0 ; i<a; i++)
        printf("%d\n", i);
}
```

Chương trình gồm 2 biến i và a, chỉ có biến i có giá trị thay đổi trong quá trình chạy chương trình nên ta lập bảng sau:

a có giá trị là 4

Bước thực hiện	Giá trị của biến i	Kết quả in ra màn hình
0	0	0
1	1	0 1
2	2	0 1 2
3	3	0 1 2 3
4	4	

Tại bước 4, giá trị của i = 4 vi phạm điều kiện lặp (i<a) nên vòng lặp kết thúc. Do đó kết quả in ra màn hình:

0
1
2
3

II.2. Bài tập cơ bản

a. Cấu trúc if/ if..else và switch

1. Cho biết kết quả của đoạn chương trình sau:

```
int a=9, b=6;
a++;
a=a+b--;
a=a+(-b);
```

```
if(a%2==0)
    printf("Gia tri cua a la chan");
printf("Tong cua a va b la: %d", a+b) ;
```

2. Cho biết kết quả của đoạn chương trình sau:

```
int a=7, b=8;
a++;
a=a+(b--);
--b;
a--;
a=(-a)+(-b);
if(a%2!=0)
    printf("\n a la so le");
else
    printf("\n a la so chan");
printf("\na = %d",a);
```

3. Cho biết kết quả của đoạn chương trình sau:

```
int x=5, y;
y=x++ + 5;
printf("x=%d, y=%d\n", x, y);
y*=6;
x=y%7;
printf("x=%d,y=%d,y/x=%d", x, y, y/x);
```

4. Nhập vào hai số nguyên a, b. In ra màn hình giá trị lớn nhất.
5. Cho ba số a, b, c đọc vào từ bàn phím. Hãy tìm giá trị lớn nhất của ba số trên và in ra kết quả.
6. Cho ba số a, b, c đọc vào từ bàn phím. Hãy in ra màn hình theo thứ tự tăng dần các số. (Chỉ được dùng thêm hai biến phụ).
7. Viết chương trình nhập vào một số nguyên n gồm ba chữ số. Xuất ra màn hình chữ số lớn nhất ở vị trí nào?
Ví dụ: $n=291$. Chữ số lớn nhất nằm ở hàng chục (9).
8. Viết chương trình nhập vào số nguyên n gồm ba chữ số. Xuất ra màn hình theo thứ tự tăng dần của các chữ số.

Ví dụ: $n=291$. Xuất ra 129.

9. Nhập vào ngày, tháng, năm. Kiểm tra xem ngày, tháng, năm đó có hợp lệ hay không? In kết quả ra màn hình.
10. Nhập vào giờ, phút, giây. Kiểm tra xem giờ, phút, giây đó có hợp lệ hay không? In kết quả ra màn hình.
11. Viết chương trình nhập vào ngày, tháng, năm hợp lệ. Cho biết năm này có phải là năm nhuận hay không? In kết quả ra màn hình.
12. Viết chương trình tính diện tích và chu vi các hình: tam giác, hình vuông, hình chữ nhật và hình tròn với những thông tin cần được nhập từ bàn phím.
13. Viết chương trình tính tiền cước TAXI. Biết rằng:
 - KM đầu tiên là 5000^d.
 - 200m tiếp theo là 1000^d.
 - Nếu lớn hơn 30km thì mỗi km thêm sẽ là 3000^d.Hãy nhập số km sau đó in ra số tiền phải trả.
14. Nhập vào 3 số nguyên dương a, b, c. Kiểm tra xem 3 số đó có lập thành tam giác không? Nếu có hãy cho biết tam giác đó thuộc loại nào? (Cân, vuông, đều, ...).
15. Viết chương trình nhập vào số nguyên dương n. Kiểm tra xem n có phải là số chính phương hay không? (số chính phương là số khi lấy căn bậc 2 có kết quả là nguyên).

b. Cấu trúc lặp

16. Cho biết kết quả của đoạn chương trình sau:

```
int a=18;  
for(int i=1; i<=a; i++)  
if(a%i==0)  
printf("\t %d", i);
```

17. Cho biết kết quả của đoạn chương trình sau:

```
for(int i=0; i<5; i++)  
{  
for(int j=0; j<=i; j++)  
printf("%d\t", j);  
printf("\n");
```



```
}
```

18. Cho biết kết quả của đoạn chương trình sau:

```
int i=10, s=0;
while(i>0)
{
    if(i%2==0)
        s+=i;
    else
        if(i>5)
            s+=2*i;
    i--;
}
printf("s = %d",s);
```

19. Cho biết kết quả của đoạn chương trình sau:

```
int a=18, i=1;
do{
    if(a%i==0)
        printf("\t %d",i);
    i++;
} while(i<=a);
```

20. Cho biết kết quả của đoạn chương trình sau:

```
int a=11, b=16, i=a;
while( i<b )
{
    if(i%2==0)
    {
        printf("\t %d", i);
        break;
    }
    i++;
}
```

21. Cho biết kết quả của đoạn chương trình sau:

```
int a=10, s=0, i=0;
```

```

while( i<a )
{
    i++;
    if(i%2==0) continue;
    else s=s+i;
}
printf("s=%d",s);

```

22. Cho biết kết quả của đoạn chương trình sau:

```

int i=1,s=0;
while(1)
{
    s=s+i++;
    if(i%2)
        i=i+2;
    else
        i=i+1;
    if(i>20)
        break;
}
printf("%d",s);

```

23. Viết chương trình in ra màn hình hình chữ nhật đặc kích thước $m \times n$ (m, n nhập từ bàn phím).

Ví dụ: Nhập $m=5, n=4$

```

* * * * *
* * * * *
* * * * *
* * * * *

```

24. Viết chương trình in ra màn hình hình chữ nhật rỗng kích thước $m \times n$ (m, n nhập từ bàn phím).

Ví dụ: Nhập $m=5, n=4$

```

* * * * *
*           *
*           *
* * * * *

```

25. Viết chương trình in ra màn hình tam giác vuông cân đặc có độ cao h (h nhập từ bàn phím).

Ví dụ: Nhập $h=4$

```

*
* *
* * *
* * * *
    
```

26. Viết chương trình in ra màn hình tam giác cân rộng có độ cao h (h nhập từ bàn phím).

Ví dụ: Nhập $h=4$

```

*
* *
*   *
* * * *
    
```

27. Viết chương trình in ra màn hình tam giác cân đặc có độ cao h (h nhập từ bàn phím).

Ví dụ: Nhập $h=4$

```

      *
     * * *
    * * * * *
   * * * * * * *
  * * * * * * * *
    
```

28. Viết chương trình in ra màn hình tam giác cân rộng có độ cao h (h nhập từ bàn phím).

Ví dụ: Nhập $h=4$

```

      *
     *   *
    *     *
   *       *
  *         *
 *           *
*             *
    
```

29. Viết chương trình nhập số nguyên dương n . Liệt kê n số nguyên tố đầu tiên.
 30. Viết chương trình nhập vào hai số nguyên dương a và b . Tìm ước số chung lớn nhất và bội số chung nhỏ nhất của a và b .
 31. Viết chương trình nhập vào một số nguyên n gồm tối đa 10 chữ số (4 bytes). In ra màn hình giá trị nhị phân của số trên. (Hướng dẫn: chia lấy dư cho 2 và xuất theo thứ tự ngược lại dùng hàm gotoxy, wherex, wherey).
 32. Viết chương trình đếm số ước số của số nguyên dương N .

Ví dụ: $N=12$

số ước số của 12 là 6

33. Một số hoàn thiện là một số có tổng các ước số của nó (không kể nó) bằng chính nó. Hãy liệt kê các số hoàn thiện nhỏ hơn 5000.

Ví dụ: số 6 là số hoàn thiện vì tổng các ước số là $1+2+3=6$.

34. Nhập vào ngày, tháng, năm. Cho biết đó là ngày thứ mấy trong năm.
35. In ra dãy số Fibonacci
- $$f_1 = f_0 = 1 ;$$
- $$f_n = f_{n-1} + f_{n-2} ; \quad (n > 1)$$

II.3. Bài tập luyện tập và nâng cao

36. Cài đặt tất cả các lưu đồ đã vẽ ở chương 1.
37. Nhập vào ngày, tháng, năm. Kiểm tra xem ngày, tháng, năm đó có hợp lệ hay không, nếu hợp lệ cho biết ngày sau đó là bao nhiêu.
Ví dụ: *Nhập 31/12/2003*
Ngày sau đó 01/01/2004
38. Nhập vào ngày, tháng, năm. Kiểm tra xem ngày, tháng, năm đó có hợp lệ hay không, nếu hợp lệ cho biết ngày trước đó là bao nhiêu.
Ví dụ: *Nhập 01/01/2003*
Ngày trước đó 31/12/2002
39. (*) Nhập vào ngày, tháng, năm của năm 2003. Hãy kiểm tra xem dữ liệu có hợp lệ hay không? Nếu hợp lệ hãy cho biết đó là ngày thứ mấy trong tuần. (hai, ba, tư, ..., CN). (Hướng dẫn: lấy ngày 01 tháng 01 năm 2003 là ngày thứ tư làm mốc).
40. Nhập vào giờ, phút, giây. Kiểm tra xem giờ, phút, giây đó có hợp lệ hay không, nếu hợp lệ cho biết giờ sau đó 1 giây là bao nhiêu.
Ví dụ: *Nhập 01:59:59*
Giờ sau đó 1 giây 02:00:00
41. Nhập vào giờ, phút, giây. Kiểm tra xem giờ, phút, giây đó có hợp lệ hay không, nếu hợp lệ cho biết giờ trước đó 1 giây là bao nhiêu.
Ví dụ: *Nhập 02:00:00*
Giờ trước đó 1 giây 01:59:59
42. Viết chương trình in ra bảng cửu chương từ 2 đến 9.
43. (*) Vẽ hình cánh quạt sau:

```
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
```

Sử dụng các hàm `cprintf()`, `textcolor()`, `delay()`, `kbhit()`, ... thay đổi màu để tạo cảm giác cho cánh quạt xoay cho đến khi nhấn một phím bất kỳ.

III. KẾT LUẬN

- ❖ **Cấu trúc lặp và rẽ nhánh** (lựa chọn) là hai cấu trúc chính hình thành nên chương trình. Dựa vào những cấu trúc điều khiển này ta có thể xây dựng thành những chương trình phức tạp hơn. Vì vậy phải nắm rõ cách hoạt động của những cấu trúc điều khiển này để cài đặt đúng yêu cầu bài toán.
- ❖ Khi sử dụng phải **lưu ý điều kiện thực hiện hay kết thúc** của một thao tác nào đó.
- ❖ **Bên trong một phát biểu điều khiển phải là một lệnh hay một khối lệnh** (*khối lệnh được đặt bên trong cặp dấu ngoặc {}*).
- ❖ Những **biến không phụ thuộc vào vòng lặp nên đặt bên ngoài vòng lặp**.
- ❖ Khi sử dụng **cấu trúc điều khiển lồng nhau phải lưu ý vị trí mở ngoặc hay đóng ngoặc cho hợp lý**.

Trình bày cấu trúc của một chương trình, các bước xây dựng cài đặt chương trình theo phương pháp thủ tục hàm và một số kỹ thuật liên quan.

I. TÓM TẮT LÝ THUYẾT

I.1. Khái niệm

Hàm là một đoạn chương trình độc lập **thực hiện trọn vẹn một công việc nhất định** sau đó trả về giá trị cho chương trình gọi nó, hay nói cách khác hàm là sự chia nhỏ của chương trình.

I.2. Ví dụ

```
//Khai báo thư viện hàm
#include<conio.h>
#include<stdio.h>
#include<string.h>
#include<dos.h>
#include<process.h>

//Khai báo biến toàn cục và nguyên mẫu hàm
void ThayThe(char * S, char *St);
void Doc1Sector(int vt);
void Ghi1Sector(int vt);

//Hàm chính
void main()
{
    unsigned char buf[512];
    char S[20], St[20];
    printf("Nhap chuoi can tim: ");
    gets(S);
    printf("Nhap chuoi thay the:");
    gets(St);
    printf("\nXin cho...");
    TimVaThayThe(S,St,buf);
    printf("\n Thanh cong.");
    getch();
}

//Cài đặt các hàm con
void ThayThe(char * S, char *St)
{
```

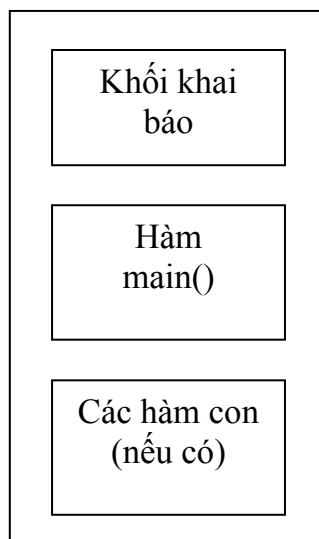
```
    int l=strlen(St);
    for(int i=0;i<l;i++)
        S[i]=St[i];
}

void Doc1Sector(int vt, char buf[512])
{
    if(absread(0,1,vt,buf))
    {
        printf("\n loi doc dia, nhan enter thoat");
        getch();
        exit(1);
    }
}

void Ghi1Sector(int vt, char buf[512])
{
    if(abswrite(0,1,vt,buf))
    {
        printf("\n loi ghi dia, nhan enter thoat");
        getch();
        exit(1);
    }
}

void TimVaThayThe(char * S, char *St, unsigned char buf[])
{
    for(int i=33;i<=500;i++)
    {
        Doc1Sector(i, buf);
        char * p=strstr(buf, S);
        if(p)
        {
            ThayThe(p, St);
            Ghi1Sector(i, buf);
        }
    }
}
```

I.3. Cấu trúc một chương trình C



a. *Khởi khai báo*

Bao gồm các khai báo về sử dụng thư viện, khai báo hằng số, khai báo hàm con (các nguyên mẫu hàm), khai báo các biến toàn cục và khai báo các kiểu dữ liệu tự định nghĩa.

b. *Hàm chính (main())*

Chứa các biến, các lệnh và các lời gọi hàm cần thiết trong chương trình.

c. *Các hàm con*

Được sử dụng nhằm mục đích:

- Khi có một công việc giống nhau cần thực hiện ở nhiều vị trí.
- Khi cần chia một chương trình lớn phức tạp thành các đơn thể nhỏ (hàm con) để chương trình được trong sáng, dễ hiểu trong việc xử lý, quản lý việc tính toán và giải quyết vấn đề.

d. *Nguyên mẫu hàm*

<Kiểu dữ liệu của hàm> Tên hàm ([danh sách các tham số]);

Nguyên mẫu hàm thực chất là dòng đầu của hàm thêm dấu chấm phẩy (;) vào cuối, tuy nhiên tham số trong nguyên mẫu hàm có thể bỏ phần tên.

I.4. Cách xây dựng một hàm con

a. Kiểu dữ liệu của hàm

Xác định dựa vào kết quả của bài toán (Output). Gồm 2 loại :

- **void:** Hàm không trả về giá trị. Những hàm loại này thường rơi vào những nhóm chức năng: **Nhập / xuất dữ liệu , thống kê, sắp xếp, liệt kê.**

```
void Tên_hàm (danh sách các tham số)
{
    Khai báo các biến cục bộ
    Các câu lệnh / khối lệnh hay lời gọi đến hàm khác.
}
```

- **Kiểu dữ liệu cơ bản (rời rạc/ liên tục) hay kiểu dữ liệu có cấu trúc:** Kiểu dữ liệu tùy theo mục đích của hàm cần trả về giá trị gì thông qua việc phân tích bài toán. Những hàm loại này thường được sử dụng trong các trường hợp: **Đếm, kiểm tra, tìm kiếm, tính trung bình, tổng, tích, ...**

```
<Kiểu dữ liệu> Tên_hàm ([danh sách các tham số])
{
    <Kiểu dữ liệu> kq;
    Khai báo các biến cục bộ
    Các câu lệnh / khối lệnh hay lời gọi đến hàm khác.

    return kq;
}
```

☞ Đối với những hàm trả về nhiều loại giá trị cho từng trường hợp cụ thể (chẳng hạn như kiểm tra: đúng hay sai, so sánh: bằng , lớn hơn hay nhỏ hơn, ...) thì cần ghi chú rõ giá trị trả về là gì cho từng trường hợp đó.

b. Tham số

Xác định dựa vào dữ liệu đầu vào của bài toán (Input). Gồm 2 loại :

- **Tham số không là con trỏ (tham trị):** Không thay đổi hoặc không cần lấy giá trị mới của tham số sau lời gọi hàm. Tham số dạng này chỉ mang ý nghĩa là **dữ liệu đầu vào.**
- **Tham số con trỏ (tham biến):** Có sự thay đổi giá trị của tham số trong quá trình thực hiện và cần lấy lại giá trị đó sau khi ra khỏi hàm. Ứng dụng của tham số loại này có thể là **dữ liệu đầu ra** (kết quả) hoặc cũng có thể vừa là **dữ liệu đầu vào** vừa là **dữ liệu đầu ra.**

c. *Tên hàm*

Đặt tên theo **quy ước đặt tên trong C** sao cho tên gọi **đúng với chức năng hay mục đích thực hiện của hàm và gọi nhớ.**

d. *Ví dụ*

Ví dụ 1: Viết chương trình nhập số nguyên dương n và in ra màn hình các ước số của n

Phân tích bài toán:

- **Input:** n (Để xác định tham số)
 - Kiểu dữ liệu: số nguyên dương (*unsigned int*).
 - Giá trị n không bị thay đổi trong quá trình tìm ước số \rightarrow Tham số của hàm *không là con trỏ*.
- **Output:** In ra các ước số của n (Để xác định kiểu dữ liệu hàm)
 - Không trả về giá trị.
 - Kiểu dữ liệu của hàm là *void*.
- **Xác định tên hàm:** Hàm này dùng in ra các ước số của n nên có thể đặt là *LietKeUocSo*

Ta có nguyên mẫu hàm:

```
void LietKeUocSo ( unsigned int n );
```

```
#include<conio.h>
#include<stdio.h>

//Khai bao nguyen mau ham
void LietKeUocSo ( unsigned int n );

void main()
{
        unsigned int n;

        printf("Nhap n = ");
        scanf("%u",&n);
        printf("Cac uoc so cua n : ");
        LietKeUocSo(n);
        getch();
}

void LietKeUocSo ( unsigned int n)
{
```

```

    for(int i=1; i<=n; i++)
        printf("%u\t", i);
}

```

⚠ **Lưu ý cách gọi hàm:** Đối với hàm có kiểu dữ liệu hàm là **void** thì khi gọi không cần phải gán giá trị vào biến, ngược lại phải gọi như trong ví dụ 2 (Phải khai báo tương ứng kiểu với kiểu dữ liệu hàm sẽ gọi và gán giá trị trả về vào biến đó).

Ví dụ 2: Viết chương trình nhập số nguyên dương n và tính tổng

$$S = 1 + 2 + 3 + \dots + n, \text{ với } n > 0$$

Phân tích bài toán:

- **Input:** n (Để xác định tham số)
 - Kiểu dữ liệu: số nguyên dương (*unsigned int*).
 - Giá trị n không bị thay đổi trong quá trình tính tổng → Tham số của hàm *không là con trỏ*.
- **Output:** Tổng S (Để xác định kiểu dữ liệu hàm)
 - Trả về giá trị của S .
 - S là tổng các số nguyên dương nên S cũng là số nguyên dương → Kiểu trả về của hàm là *unsigned int* (hoặc *unsigned long* cho trường hợp giá trị của tổng lớn hơn 2 bytes).
- **Xác định tên hàm:** Hàm này dùng tính tổng S nên có thể đặt là *TongS*.

Ta có nguyên mẫu hàm:

```

    unsigned long TongS ( unsigned int n );

```

```

#include<conio.h>
#include<stdio.h>

//Khai bao nguyen mau ham
unsigned long TongS ( unsigned int n );

void main()
{
    unsigned int n;
    unsigned long kq;

    printf("Nhap n = ");

```

```

scanf("%u",&n);
kq = TongS ( n );
printf("Tong can tinh la: %lu ", kq);
getch();
}
unsigned long TongS (unsigned int n)
{
    unsigned long S=0;
    int i=1;
    while(i<=n)
    {
        S+=i;
        i++;
    }
    return S;
}

```

II. BÀI TẬP

II.1. Bài tập cơ bản

1. Cài đặt lại tất cả các bài tập ở chương 2 theo phương pháp hàm.
2. Viết chương trình tính diện tích và chu vi của hình chữ nhật với chiều dài và chiều rộng được nhập từ bàn phím.
3. Viết chương trình tính diện tích và chu vi hình tròn với bán kính được nhập từ bàn phím.
4. Nhập số nguyên dương n ($n > 0$). Liệt kê tất cả các số nguyên tố nhỏ hơn n .
5. Nhập số nguyên dương n ($n > 0$). Liệt kê n số chính phương đầu tiên.
6. Nhập số nguyên dương n ($n > 0$). Đếm xem có bao nhiêu số hoàn thiện nhỏ hơn n .
7. Nhập số nguyên dương n ($0 \leq n < 1000$) và in ra cách đọc của n .
Ví dụ: Nhập $n = 105$. In ra màn hình: *Mot tram le nam*.
8. Viết chương trình tính tiền thuê máy dịch vụ Internet và in ra màn hình kết quả. Với dữ liệu nhập vào là giờ bắt đầu thuê (GBD), giờ kết thúc thuê (GKT), số máy thuê (SoMay).
 - Điều kiện cho dữ liệu nhập: $6 \leq \text{GBD} < \text{GKT} \leq 21$. Giờ là số nguyên.
 - Đơn giá: 2500đ cho mỗi giờ máy trước 17:30 và 3000đ cho mỗi giờ máy sau 17:30.
9. Viết chương trình tính tiền lương ngày cho công nhân, cho biết trước giờ vào ca, giờ ra ca của mỗi người.

Giả sử rằng:

- Tiền trả cho mỗi giờ trước 12 giờ là 6000đ và sau 12 giờ là 7500đ.
- Giờ vào ca sớm nhất là 6 giờ sáng và giờ ra ca trễ nhất là 18 giờ (*Giả sử giờ nhập vào nguyên*).

10. Nhập vào 2 số nguyên p, q và tính biểu thức sau:

$$(-q/2 + (p^3/27 + q^2/4)^{1/2})^{1/3} + (-q/2 - (p^3/27 + q^2/4)^{1/2})^{1/3}$$

11. Nhập vào 3 số thực a, b, c và kiểm tra xem chúng có thành lập thành 3 cạnh của một tam giác hay không? Nếu có hãy tính diện tích, chiều dài mỗi đường cao của tam giác và in kết quả ra màn hình.

- Công thức tính diện tích $s = \sqrt{p \cdot (p-a) \cdot (p-b) \cdot (p-c)}$
 - Công thức tính các đường cao: $h_a = 2s/a, h_b = 2s/b, h_c = 2s/c$.
- (Với p là nửa chu vi của tam giác).

12. Nhập vào 6 số thực a, b, c, d, e, f . Giải hệ phương trình sau :

$$\begin{cases} ax+by=c \\ dx+ey=f \end{cases}$$

13. Viết chương trình nhập 2 số nguyên dương a, b . Tìm USCLN và BSCNN của hai số nguyên đó.

14. Viết chương trình tính tổng nghịch đảo của n giai thừa.

15. Cho 2 số nguyên a, b . Viết hàm hoán vị giá trị 2 số trên.

16. (*) Viết chương trình nhập số nguyên dương n gồm 5 chữ số, kiểm tra xem các chữ số n có phải là số đối xứng hay không.

Ví dụ: *Đối xứng:* 13531

Không đối xứng: 13921

17. Viết chương trình nhập số nguyên dương n gồm k chữ số ($0 < k \leq 5$), đếm xem n có bao nhiêu chữ số chẵn và bao nhiêu chữ số lẻ.

18. Viết chương trình nhập số nguyên dương n gồm k chữ số ($0 < k \leq 5$), đếm xem n có bao nhiêu chữ số là số nguyên tố.

19. Viết chương trình nhập số nguyên dương n gồm k chữ số ($0 < k \leq 5$), tính tổng các ước số dương của n .

Ví dụ: Nhập $n=6$

Tổng các ước số từ 1 đến n : $1+2+3+6=12$.

20. Viết chương trình nhập số nguyên dương n gồm k chữ số ($0 < k \leq 5$), tìm ước số lẻ lớn nhất của n .

Ví dụ: Ước số lẻ lớn nhất của 27 là 9.

21. Viết chương trình nhập số nguyên dương n gồm k chữ số ($0 < k \leq 5$), kiểm tra xem các chữ số của n có toàn lẻ hay toàn chẵn không.
22. (*) Viết chương trình nhập số nguyên dương n gồm k chữ số ($0 < k \leq 5$), sắp xếp các chữ số của n theo thứ tự tăng dần.

Ví dụ: Nhập $n=1536$

Kết quả sau khi sắp xếp: 1356.

II.2. Bài tập luyện tập và nâng cao

23. Viết chương trình nhập số nguyên dương n gồm k chữ số ($0 < k \leq 5$), sau đó nhập một số nguyên x , tìm vị trí xuất hiện của chữ số có giá trị x trong n .

Ví dụ: Nhập $n=1526$, $x=2$

Kết quả: Chu số 2 ở vị trí thứ 3.

24. Viết chương trình nhập số nguyên dương n gồm k chữ số ($0 < k \leq 5$), kiểm tra xem các chữ số của n có được sắp thứ tự không.

Ví dụ: Nhập $n=1569$ hoặc $n=8521$

Kết quả: Có thứ tự.

25. Viết chương trình nhập 2 số a, b sao cho: số lớn nhất trong 2 số phải là một số dương và chia hết cho 7. Nếu nhập sai phải yêu cầu nhập lại cho đến khi đúng.
26. Viết chương trình nhập số nguyên dương n gồm k chữ số ($0 < k \leq 5$), tính giá trị trung bình các chữ số chẵn trong n .
27. (*) Viết chương trình in ra màn hình ngày/tháng/năm của ngày hiện tại, cho phép sử dụng các phím mũi tên lên, xuống để tăng hoặc giảm một ngày.
28. (*) Viết chương trình in ra màn hình giờ:phút:giây hiện tại, cho phép sử dụng các phím mũi tên lên, xuống để tăng hoặc giảm một giây.

III. KẾT LUẬN

- ❖ Trước khi xây dựng một hàm ta phải **xác định mục đích của hàm** là dùng để làm gì, trên cơ sở đó, ta mới xác định được các thành phần của hàm và xây dựng nguyên mẫu hàm.
- ❖ Mỗi hàm phải **thực hiện một chức năng độc lập** và tách biệt với các hàm khác (*không được lồng nhau*).

- ❖ Đối với hàm có giá trị trả về phải lưu ý kiểu dữ liệu phải tương ứng kiểu dữ liệu cả giá trị trả về và kiểu dữ liệu của biến được gán khi gọi hàm. Trường hợp hàm trả về từ **hai loại giá trị trở lên thì phải có dòng chú thích cho trường hợp tương ứng** để khi gọi hàm biết được kết quả (*chẳng hạn như tìm kiếm, kiểm tra, so sánh, ... giá trị trả về có 2 trường hợp: Có hoặc không có phần tử cần tìm, thỏa điều kiện kiểm tra hay không? Do vậy ta phải quy ước giá trị cho từng trường hợp*).
- ❖ Nên đặt tên hàm sao cho **gọi nhớ** được chức năng, đặt tên **theo quy tắc nhất định** để **tránh việc gọi sai tên hàm** do lẫn lộn giữa ký tự hoa và thường, có dấu gạch nối giữa các từ trong hàm hay không?
- ❖ Khi gọi hàm phải truyền **đủ tham số, đúng kiểu dữ liệu** và **đúng thứ tự** của tham số.

CHƯƠNG 4 MẢNG MỘT CHIỀU

Cách khai báo dữ liệu kiểu mảng, các thao tác nhập xuất, các kỹ thuật thao tác trên mảng. Ứng dụng các kỹ thuật này trong việc cài đặt các hàm tìm kiếm, kiểm tra, xây dựng mảng, tách và ghép mảng.

I. TÓM TẮT LÝ THUYẾT

I.1. Khái niệm

Mảng thực chất là một biến được cấp phát bộ nhớ liên tục và bao gồm nhiều biến thành phần.

Các thành phần của mảng là tập hợp các biến có cùng kiểu dữ liệu và cùng tên. Do đó để truy xuất các biến thành phần, ta dùng cơ chế chỉ mục.

I.2. Khai báo mảng

Để khai báo một mảng, ta có 2 cách khai báo sau :

❖ Cách 1: Con trỏ hằng

< Kiểu dữ liệu > < Tên mảng > [< Số phần tử tối đa của mảng >] ;

Ví dụ:

int a[100]; // Khai báo mảng số nguyên a gồm 100 phần tử

float b[50]; // Khai báo mảng số thực b gồm 50 phần tử

❖ Cách 2: Con trỏ

Ý nghĩa: Khi ta khai báo một mảng với kiểu dữ liệu bất kì (int, float, char,...) thì tên của mảng thực chất là một **hằng địa chỉ của phần tử đầu tiên**.

< Kiểu dữ liệu > * < Tên mảng >;

Ví dụ :

*int *p; // khai báo con trỏ p*

int b[100];

p = b; // p trỏ vào phần tử 0 của mảng b

Với cách viết như trên thì ta có thể hiểu các cách viết sau là tương đương

$p[i] \Leftrightarrow *(p + i) \Leftrightarrow b[i] \Leftrightarrow *(b+i)$

☞ **Lưu ý:** Khi sử dụng biến con trỏ để truy xuất mảng, theo cách như trên thì thực chất con trỏ p chỉ chiếm 2 byte bộ nhớ để chứa địa chỉ mà thôi. Để tạo mảng chứa dữ liệu thành phần thì ta phải cấp phát vùng nhớ cho con trỏ p .
Dùng hàm : **malloc, calloc** trong thư viện **<stdlib.h>** để cấp phát vùng nhớ.

Ví dụ:**+ Cách 1: dùng malloc**

```
int *px; //Khai báo con trỏ px
px = (int *) malloc (100); //Cấp phát 100 ô nhớ kiểu int cho con trỏ px
```

+ Cách 2: dùng calloc

```
int *p; //khai báo con trỏ p
p=(int *) calloc (100,sizeof (int)); //cấp phát 10 ô nhớ mỗi ô chiếm 2bytes
```

Sau khi sử dụng xong thì nên giải phóng vùng nhớ bằng hàm free

Ví dụ : free (p) ; // giải phóng vùng nhớ cho con trỏ p.

I.3. Truy xuất phần tử của mảng

Với khái niệm và cách khai báo như trên ta có hình dạng của mảng một chiều như sau:

Ví dụ : int A[5] // Khai báo mảng A gồm tối đa 5 phần tử nguyên.

Chỉ số 0 1 2 3 4

A[0]	A[1]	A[2]	A[3]	A[4]
------	------	------	------	------

Ví dụ minh họa:

Khai báo và gán giá trị cho mảng

```
#include <conio.h>
#include <stdio.h>

void main ()
{
    clrscr ();
    int a[4] = {5,9,3,8};
    for (int i = 0; i < 4 ; i++)
        printf (" a [ %d ] = %d \t", i , a[i] );
    getch ();
}
```

Đối với con trỏ: Lấy địa chỉ của phần tử trong mảng ta dùng dấu “&”

Ví dụ:

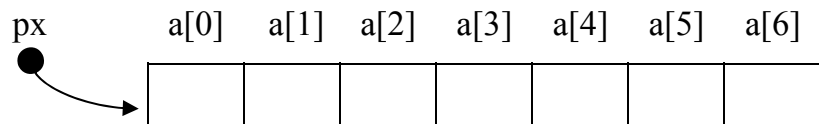
```
int a[7];
```

```
int *p = a[3]; //Lấy địa chỉ phần tử thứ 3
```

Ví dụ :

```
int a[7];
int *px;
px = a; //px trở tới phần tử thứ 0
px = px + 4; //px trở tới phần tử thứ 4
```

Từ ví dụ trên ta có thể mô hình hoá mảng như sau:



Ví dụ minh họa: Viết chương trình nhập vào mảng một chiều 10 phần tử kiểu số nguyên

```
#include <conio.h>
#include <stdio.h>

void main ()
{
    int a[10], i;
    int *p;

    for (i = 0 ; i < 10 ; i ++)
    {
        printf (" a [ %d ] = ", i);
        scanf (" %d", &a[i] );
    }
    p = a;
    printf ("\n Nội dung mảng vừa nhập: ");
    for (i = 0; i < 10 ; i ++)
        printf (" %d \t ", *(p + i));
    getch ();
}
```

II. BÀI TẬP

II.1. Một số kĩ thuật cơ bản

a. Kĩ thuật đặt cờ hiệu

Kĩ thuật này thường được áp dụng cho những bài toán “kiểm tra” hay “đánh dấu”.

Viết hàm kiểm tra xem mảng các số nguyên có thứ tự tăng dần không?
(Trả về 1: Nếu mảng tăng dần, ngược lại trả về 0).

```

int KiemTraTang (int a[ ], int n)
{
    int flag = 1;
    for (int i = 0; i < n-1; i++)
        if ( a[i] > a[i+1] ) //Vi phạm điều kiện tăng dần
        {
            flag = 0;
            break;
        }
    return flag;
}

```

Viết hàm kiểm tra xem trong mảng các số nguyên có tồn tại số nguyên lẻ lớn hơn 100 hay không?

(Trả về 1: Nếu có tồn tại số lẻ và lớn hơn 100, ngược lại trả về 0).

```

int KiemTraLe (int a[ ], int n)
{
    int flag = 0;
    for (int i = 0; i < n; i++)
        if ( a[i] % 2 != 0 && a[i]>100 ) //Gặp phần tử thoả
        {
            flag = 1;
            break;
        }
    return flag;
}

```

b. Kỹ thuật đặt lính canh

Kỹ thuật này thường được áp dụng cho những bài tập về “tìm kiếm”, “liệt kê” theo một điều kiện nhất định nào đó.

Viết hàm tìm và trả về giá trị lớn nhất trong mảng một chiều các số nguyên.

```

int TimMax (int a[], int n)
{
    int max, i = 1;

    max = a[0];
    while ( i < n )
    {
        if ( a[i] > max )
            max = a[i] ;
        i++;
    }
    return max;
}

```

II.2. Bài tập cơ bản

a. Nhập xuất mảng một chiều

Phương pháp cơ bản

Viết chương trình nhập xuất mảng một chiều các số nguyên.

```
#include <conio.h>
#include <stdio.h>
#define MAX 100

void NhapMang (int a[], int &n)
{
    printf ("Nhap so phan tu: ");
    scanf ("%d", &n);
    for (int i = 0; i < n; i++)
    {
        printf (" a [%d] = ", i);
        scanf ("%d", &a[i]);
    }
}

void XuatMang (int a[], int n)
{
    printf ("\nNoi dung mang: ");
    for (int i = 0; i < n; i++)
        printf (" %d\t", a[i]);
}

void main ()
{
    clrscr ();
    int a[MAX], n;
    NhapMang (a,n);
    XuatMang (a,n);
    getch ();
}
```

Bài tập

1. Viết chương trình nhập xuất mảng một chiều các số thực.
2. Viết chương trình khởi tạo giá trị các phần tử là 0 cho mảng một chiều các số nguyên gồm n phần tử.
3. Viết chương trình phát sinh ngẫu nhiên mảng một chiều các số nguyên âm.
4. Viết chương trình phát sinh ngẫu nhiên mảng một chiều các số nguyên sao cho mảng có thứ tự tăng dần (Không sắp xếp).

- Viết chương trình nhập mảng các số thực và xuất các phần tử âm trong mảng.
- Viết chương trình nhập mảng các số nguyên và xuất các phần tử lẻ có trong mảng.
- Viết chương trình nhập vào mảng một chiều các số nguyên và xuất ra các phần tử chẵn nhỏ hơn 20.
- Viết chương trình nhập vào mảng một chiều các số nguyên và xuất ra màn hình các phần tử là số nguyên tố.
- Viết chương trình nhập vào số nguyên n và liệt kê các số nguyên tố nhỏ hơn n, nếu mảng không tồn tại số nguyên tố nào nhỏ hơn n thì phải xuất ra một câu thông báo.
- Viết chương trình nhập vào mảng một chiều các số nguyên và xuất ra màn hình các phần tử là số chính phương nằm tại những vị trí lẻ trong mảng.

b. Tìm kiếm trên mảng một chiều

Phương pháp cơ bản

Viết hàm tìm phần tử có giá trị x xuất hiện đầu tiên trong mảng một chiều.

(Nếu tìm thấy trả về vị trí xuất hiện x, ngược lại trả về -1)

```
int TimX (int a[], int n, int x)
{
    for (int i = 0; i < n ; i++)
        if (x==a[i])
            return i;
    return -1;
}
```

Bài tập

- Viết hàm tìm vị trí phần tử có giá trị x xuất hiện cuối cùng trong mảng.
- Viết hàm tìm vị trí của phần tử nhỏ nhất trong mảng các số nguyên.
- Viết hàm tìm vị trí của phần tử lớn nhất trong mảng các số nguyên.
- Viết hàm in vị trí các phần tử nguyên tố trong mảng các số nguyên.
- Viết hàm in vị trí các phần tử nguyên tố lớn hơn 23.
- Viết hàm tìm vị trí phần tử âm đầu tiên trong mảng. Nếu không có phần tử âm trả về -1.
- Viết hàm tìm vị trí phần tử âm lớn nhất trong mảng.

18. Viết hàm tìm vị trí phần tử dương đầu tiên trong mảng. Nếu không có phần tử âm trả về -1.
19. Viết hàm tìm vị trí phần tử dương bé nhất trong mảng.
20. Viết hàm in các phần tử là bội của 3 và 5.
21. Viết hàm tìm số chẵn cuối cùng có trong mảng, nếu không tồn tại số chẵn hàm trả về -1 .
22. Viết hàm tìm số lẻ lớn nhất có trong mảng, nếu không tồn tại số lẻ hàm trả về -1.
23. Viết hàm tìm và đổi chỗ phần tử lớn nhất với phần tử nhỏ nhất trong mảng.
24. Nhập vào X. Viết hàm in ra màn hình những phần tử có giá trị từ 1 đến X có trong mảng.
25. Viết chương trình nhập vào một dãy số a gồm n số thực ($n \leq 100$), nhập vào dãy số b gồm m số thực ($m \leq 100$).
 - In ra những phần tử chỉ xuất hiện trong dãy a mà không xuất hiện trong dãy b.
 - In ra những phần tử xuất hiện ở cả hai dãy.

c. Đếm – Tần suất

Phương pháp cơ bản

Viết hàm đếm các phần tử chia hết cho 5 trong mảng các số nguyên.

```
int Dem (int a[], int n )
{
    int dem = 0;
    for (int i = 0; i < n ; i++)
        if ( a[i] % 5 == 0 )
            dem++;
    return dem;
}
```

Bài tập

26. Viết hàm đếm các phần tử âm, dương trong mảng.
27. Viết hàm đếm các phần tử chẵn, lẻ trong mảng.
28. Viết hàm đếm số lần xuất hiện của phần tử x trong mảng.
29. Viết hàm đếm các phần tử nhỏ hơn x trong mảng.
30. Viết hàm đếm các phần tử là số nguyên tố trong mảng.

31. Viết hàm đếm các phần tử là số hoàn thiện trong mảng.
32. Viết hàm đếm các phần tử là bội của 3 và 5 trong mảng các số nguyên.

d. Tính tổng – Trung bình có điều kiện

Phương pháp cơ bản

Viết hàm tính tổng các phần tử trong mảng.

```
long TinhTong (int a[], int n)
{
    long tong = 0;
    for (int i = 0; i < n; i++)
        tong = tong + a[i];
    return tong;
}
```

Viết hàm tính giá trị trung bình các phần tử có giá trị âm trong mảng.

Đối với hàm tính trung bình có điều kiện phải lưu ý khi chia giá trị (Có thể mảng không có phần tử nào thoả điều kiện, nếu ta chia tức là chia cho 0).

```
float TrungBinhAm (int a[], int n)
{
    long tong = 0;
    int spt=0;
    for (int i = 0; i < n; i++)
        if( a[i]<0 )
        {
            tong = tong + a[i];
            spt++;
        }
    if(spt==0)
        return 0;
    return 1.0*tong/spt;
}
```

Bài tập

33. Viết hàm tính tổng các phần tử chẵn trong mảng.
34. Viết hàm tính tổng các phần tử lẻ trong mảng các số nguyên.
35. Viết hàm tính tổng các phần tử nguyên tố trong mảng.
36. Viết hàm tính tổng các phần tử nằm ở vị trí chẵn trong mảng các số nguyên.
37. Viết hàm tính tổng các phần tử nằm ở vị trí nguyên tố trong mảng.
38. Viết hàm tính tổng các phần tử chia hết cho 5 có trong mảng.
39. Viết hàm tính tổng các phần tử cực đại trong mảng các số nguyên (*phần tử cực đại là phần tử lớn hơn các phần tử xung quanh nó*).

Ví dụ : 1 5 2 6 3 5 1 8 6

40. Viết hàm tính tổng các phần tử cực tiểu trong mảng các số nguyên (*phần tử cực tiểu là phần tử nhỏ hơn các phần tử xung quanh nó*).

Ví dụ : 6 4 2 9 5 3 7 1 5 8

41. Viết hàm tính tổng các phần tử là bội của 3 và 5 trong mảng các số nguyên.
42. Viết hàm tính tổng các phần tử là số hoàn thiện trong mảng các số nguyên.
43. Viết hàm tính giá trị trung bình của các số hoàn thiện trong mảng các số nguyên.

e. Sắp xếp

Kĩ thuật cơ bản

Viết hàm sắp xếp mảng theo thứ tự tăng dần.

```
void HoanVi (int &a, int &b)
{
    int tam = a;
    a = b;
    b = tam;
}

void SapTang (int a[], int n)
{
    for (int i = 0; i < n-1; i++)
        for (int j = i+1; j < n; j++)
            if (a[i] > a [j])
                HoanVi (a[i], a[j]);
}
```

Bài tập

44. Viết hàm sắp xếp mảng theo thứ tự giảm dần.
45. Viết hàm sắp xếp mảng theo thứ tự tăng dần của các phần tử là số nguyên tố.
46. Viết hàm sắp xếp các phần tử lẻ tăng dần.
47. Viết hàm sắp xếp các phần tử chẵn giảm dần.
48. Viết hàm sắp xếp các phần tử chẵn nằm bên trái theo thứ tự tăng dần còn các phần tử lẻ bên phải theo thứ tự giảm dần.
49. Viết hàm sắp xếp các phần tử âm giảm dần từ trái sang phải, phần tử dương tăng dần từ phải sang trái.

f. Xoá

Kĩ thuật cơ bản

Duyệt mảng từ trái sang phải. Xuất phát từ vị trí cần xoá tiến hành dời lần lượt các phần tử về phía trước cho đến khi kết thúc mảng, sau đó giảm kích thước mảng.

Vấn đề đặt ra là tìm vị trí cần xoá theo điều kiện bài toán rồi thực hiện xoá.

Viết hàm xoá phần tử đầu tiên của mảng.

```
void XoaDau (int a[], int &n)
{
    for (int i = 0; i < n-1 ; i++)
        a[i] = a[i+1];
    n--;
}
```

Viết hàm xoá phần tử tại vị trí (vitri) cho trước trong mảng.

```
void XoaTaiViTri (int a[], int &n, int vitri)
{
    for (int i = vitri; i < n-1 ; i++)
        a[i] = a[i+1];
    n--;
}
```

Bài tập

50. Viết hàm xoá phần tử tại vị trí lẻ trong mảng.
51. Viết hàm xoá phần tử có giá trị lớn nhất trong mảng.
52. Nhập vào giá trị X. Viết hàm xoá tất cả các phần tử có giá trị nhỏ hơn X.
53. Nhập vào giá trị X. Viết hàm xoá phần tử có giá trị gần X nhất.

g. Chèn

Kĩ thuật cơ bản

Duyệt mảng từ phải sang trái. Xuất phát từ cuối mảng tiến hành đẩy lần lượt các phần tử về phía sau cho đến vị trí cần chèn, chèn phần tử cần chèn vào vị trí chèn và tăng kích thước mảng.

Trước khi chèn ta phải xác định vị trí cần chèn theo điều kiện bài toán.

Thêm phần tử có giá trị X vào cuối mảng.

```
void ThemCuoi (int a[], int &n, int X)
{
    a[n]=X;
    n++;
}
```

}

Chèn phần tử có giá trị X vào mảng tại vị trí cho trước

```

void ChenX (int a[], int &n, int X, int vitri)
{
    for (int i = n; i > vitri; i--)
        a[i] = a[i-1];
    a[vitri] = X;
    n++;
}

```

Bài tập

54. Viết hàm chèn phần tử có giá trị X vào vị trí đầu tiên của mảng.
55. Viết hàm chèn phần tử có giá trị X vào phía sau phần tử có giá trị lớn nhất trong mảng.
56. Viết hàm chèn phần tử có giá trị X vào trước phần tử có giá trị là số nguyên tố đầu tiên trong mảng.
57. Viết hàm chèn phần tử có giá trị X vào phía sau tất cả các phần tử có giá trị chẵn trong mảng.

h. Tách / ghép mảng**Kĩ thuật tách cơ bản**

Cho mảng a kích thước n (n chẵn). Tách mảng a thành 2 mảng b và c sao cho: b có $\frac{1}{2}$ phần tử đầu của mảng a, $\frac{1}{2}$ phần tử còn lại đưa vào mảng c.

```

void TachMang(int a[], int n, int b[], int &m, int c[], int &l)
{
    int k=n/2;

    m=l=0;
    for(int i=0; i<k; i++)
    {
        b[m++]=a[i];
        c[l++]=a[k+i]
    }
}

```

Kĩ thuật ghép cơ bản

Cho 2 mảng số nguyên a và b kích thước lần lượt là n và m. Viết chương trình nối mảng b vào cuối mảng a.

```
void NoiMang(int a[], int &n, int b[], int m)
{
    for(int i=0; i<m; i++)
        a[n+i]=b[i];
    n=n+m;
}
```

Cho 2 mảng số nguyên a và b kích thước lần lượt là n và m. Viết chương trình nối xen kẽ (*đan xen*) lần lượt các phần tử mảng a và b vào mảng c.

Cách thực hiện: Đưa lần lượt từng phần tử của mảng a và mảng b vào mảng c, tăng chỉ số tương ứng. Nếu một trong hai mảng hết trước thì chép tất cả các phần tử còn lại của mảng chưa hết vào mảng c.

Đặt *i* là chỉ số của mảng a; *j*: chỉ số của mảng b và *k* là chỉ số của mảng c.

```
void NoiMang(int a[], int &n, int b[], int m, int c[], int &k)
{
    int i=0, j=0;
    k=0;
    while(i<n && j<m)
    {
        c[k++]=a[i++];
        c[k++]=b[j++];
    }
    while(i<n)
        c[k++]=a[i++];
    while(j<m)
        c[k++]=b[j++];
}
```

Bài tập

58. Viết chương trình tách 1 mảng các số nguyên thành 2 mảng a và b, sao cho mảng a chứa toàn số lẻ và mảng b chứa toàn số chẵn.

Ví dụ: Mảng ban đầu: 1 3 8 2 7 5 9 0 10

Mảng a: 1 3 7 5 9

Mảng b: 8 2 10

59. Cho 2 mảng số nguyên a và b kích thước lần lượt là n và m. Viết chương trình nối 2 mảng trên thành mảng c theo nguyên tắc chẵn ở đầu mảng và lẻ ở cuối mảng.

Ví dụ: Mảng a: 3 2 7 5 9

Mảng b: 1 8 10 4 12 6

Mảng c: 6 12 4 10 2 8 3 1 7 5 9

II.3. Bài tập luyện tập và nâng cao

60. Viết chương trình nhập vào mảng A gồm n phần tử, trong quá trình nhập kiểm tra các phần tử nhập vào không được trùng, nếu trùng thông báo và yêu cầu nhập lại.
61. Viết hàm tính tổng của từng dãy con giảm có trong mảng.
62. (*) Cho mảng các số nguyên a gồm n phần tử ($n \leq 30000$) và số dương k ($k \leq n$). Hãy chỉ ra số hạng lớn thứ k của mảng.

Ví dụ: Mảng a: 6 3 1 10 11 18

$$k = 2$$

Kết quả: 10

63. (*) Cho 2 dãy A, B các số nguyên (kích thước dãy A nhỏ hơn dãy B). Hãy kiểm tra xem A có phải là con của B hay không?
64. Viết hàm liệt kê các bộ 4 số a, b, c, d trong mảng các số nguyên (có ít nhất 4 phần tử và đôi một khác nhau) sao cho $a + b = c + d$.
65. (*) Viết chương trình tính trung bình cộng của các tổng các dãy tăng dần có trong mảng các số nguyên.

Ví dụ: 1 2 3 4 2 3 4 5 6 4 5 6 => TB = 15.

66. Viết chương trình tính tổng tất cả các phần tử xung quanh trên mảng các số nguyên. (Phần tử xung quanh là hai phần tử bên cạnh cộng lại bằng chính nó (Ví dụ: 1 3 2 → 1,2 là hai phần tử xung quanh của 3).

Ví dụ: 1 3 2 5 3 9 6 → tổng 17

67. (**) Viết chương trình nhập vào hai số lớn a, b nguyên (a, b có từ 20 chữ số trở lên). Tính tổng, hiệu, tích, thương của hai số trên.
68. Viết hàm tính tổng các phần tử là số Armstrong (số Armstrong là số có đặc điểm như sau: số có k ký số, tổng của các lũy thừa bậc k của các ký số bằng chính số đó.

Ví dụ: 153 là số có các ký số $1^3 + 5^3 + 3^3 = 153$ là một số Armstrong).

69. Viết hàm tìm và xóa tất cả các phần tử trùng với x trong mảng một chiều các số nguyên, nếu không tồn tại phần tử x trong mảng thì trả về -1.

70. Viết hàm xoá tất cả những phần tử trùng nhau trong dãy chỉ giữ lại một phần tử trong đó.

Ví dụ: 1 6 2 3 2 4 2 6 5 → 1 6 2 3 4 5

71. (**) Viết hàm xoá những phần tử sao cho mảng kết quả có thứ tự tăng dần và số lần xoá là ít nhất.

72. Cho dãy a gồm n số nguyên có thứ tự tăng dần. Nhập vào một phần tử nguyên X, viết hàm chèn X vào dãy sao cho dãy vẫn có thứ tự tăng dần (*không sắp xếp*).

73. Viết chương trình tìm số lẻ nhỏ nhất lớn hơn mọi số chẵn có trong mảng.

74. Viết hàm tìm giá trị chẵn nhỏ nhất nhỏ hơn mọi giá trị lẻ trong mảng các số nguyên.

75. Viết hàm tìm phần tử xuất hiện nhiều nhất trong mảng các số nguyên.

76. Viết chương trình đếm và liệt kê các mảng con tăng dần trong mảng một chiều các số nguyên.

Ví dụ: 6 5 3 2 3 4 2 7 các dãy con tăng dần là 2 3 4 và 2 7

77. Viết chương trình tìm mảng con tăng dần có tổng lớn nhất trong mảng một chiều.

78. (*) Viết chương trình nhập vào một dãy số a gồm n số nguyên ($n \leq 100$). Tìm và in ra dãy con tăng dài nhất

Ví dụ: Nhập dãy a : 1 2 3 6 4 7 8 3 4 5 6 7 8 9 4 5

Dãy con tăng dài nhất : 3 4 5 6 7 8 9

79. (**) Viết chương trình tách 1 mảng các số nguyên thành 2 mảng a và b, sao cho kết quả thu được là:

- Mảng a chứa toàn số lẻ tăng dần.
- Mảng b chứa toàn số chẵn giảm dần.

(*Không dùng sắp xếp*)

Hướng dẫn: Tìm vị trí chèn thích hợp khi trích phần tử từ mảng ban đầu.

Ví dụ: Mảng ban đầu: 9 3 8 2 7 5 1 0 10

Mảng a: 1 3 5 7 9

Mảng b: 10 8 2

80. (**) **Viết** chương trình in ra tam giác Pascal (dùng mảng một chiều).

81. Viết chương trình nhập vào dãy số a gồm n số thực ($n \leq 100$), nhập vào dãy số b gồm m số thực ($m \leq 100$).

- Hãy sắp xếp hai dãy theo thứ tự tăng dần.

- (*) Trộn 2 dãy trên thành dãy c sao cho dãy c vẫn có thứ tự tăng.
 - Xuất dãy a, b, c ra màn hình.
82. (*) Cho mảng C có n phần tử ($n < 200$), các phần tử là các chữ số trong hệ đếm cơ số 16 (Hexa) (điều kiện mỗi phần tử $\leq n$). Hãy tách mảng C ra các mảng con theo điều kiện sau: các mảng con được giới hạn bởi hai lần xuất hiện thứ hai của con số trong dãy.
- Ví dụ: 123A4518B23 → có các dãy con là 123A451, 23A4518B2, 23A4518B23*
83. (**) Cho hai số nguyên dương A, B. Hãy xác định hai số C, D tạo thành từ hai số A, B sao cho C là số lớn nhất, D là số nhỏ nhất. Khi gạch đi một số chữ số trong C (D), thì các số còn lại giữ nguyên tạo thành A, các chữ số bỏ đi giữ nguyên tạo thành B.
- Ví dụ: A = 52568, B = 462384 -> C = 54625682384, D = 45256236884.*
84. Viết chương trình nhập vào dãy số a gồm n số nguyên ($n \leq 100$).
- Hãy đảo ngược dãy đó.
- Ví dụ: Nhập a: 3 4 5 2 0 4 1
Dãy sau khi đảo: 1 4 0 2 5 4 3*
- (*) Hãy kiểm tra xem dãy đã cho có thứ tự chưa (dãy được gọi là thứ tự khi là dãy tăng hoặc dãy giảm).
85. Cho mảng A có n phần tử hãy cho biết mảng này có đối xứng hay không.
86. (**) Hãy viết chương trình phát sinh ngẫu nhiên mảng các số nguyên gồm 10.000 phần tử, mỗi phần tử có giá trị từ 0 đến 32.000 và xây dựng hàm thống kê số lần xuất hiện các phần tử trong mảng, sau đó cho biết phần tử nào xuất hiện nhiều lần nhất.
- Ví dụ: Mảng: 5 6 11 4 4 5 4
5 xuất hiện 2 lần
6 xuất hiện 1 lần
11 xuất hiện 1 lần
4 xuất hiện 3 lần*
- 4 xuất hiện nhiều lần nhất**
87. Cho mảng A có n phần tử. Nhập vào số nguyên k ($k \geq 0$), dịch phải xoay vòng mảng A k lần.
- Ví dụ: Mảng A: 5 7 2 3 1 9
Nhập k = 2
Dịch phải xoay vòng mảng A: 1 9 5 7 2 3*

III. KẾT LUẬN

- ❖ Dữ liệu kiểu mảng dùng cho việc biểu diễn những thông tin có **cùng kiểu** dữ liệu liên tiếp nhau.
- ❖ Khi cài đặt bài tập mảng một chiều nên **xây dựng thành những hàm chuẩn** để dùng lại cho các bài tập khác.
- ❖ Các thao tác trên mảng đều theo quy tắc nhất định, chúng ta có thể ứng dụng mảng trong việc biểu diễn số lớn, dùng bảng tra, khử đệ qui, ...

CHƯƠNG 5 CHUỖI KÝ TỰ

Chuỗi ký tự là trường hợp đặc biệt của mảng một chiều. Chương này mô tả một số hàm thư viện thao tác trên chuỗi và các kỹ thuật cài đặt xử lý trên chuỗi.

I. TÓM TẮT LÝ THUYẾT

I.1. Khái niệm

Chuỗi ký tự là một dãy các phần tử, mỗi phần tử có kiểu ký tự.

Lưu ý: Chuỗi ký tự được kết thúc bằng ký tự '\0'. Do đó khi khai báo độ dài của chuỗi luôn luôn khai báo dư 1 phần tử để chứa ký tự '\0'.

Ví dụ: `char S[5]="CNTT"` //khai báo chuỗi có 5 phần tử kiểu char và gán dãy ký tự `CNTT` và chuỗi.

C	N	T	T	\0
Phần tử S[0]	Phần tử S[1]	Phần tử S[2]	Phần tử S[3]	Phần tử S[4]

Chuỗi rỗng là chuỗi chưa có ký tự nào trong mảng ký hiệu ""

I.2. Khai báo chuỗi

Để khai báo một chuỗi, ta có 2 cách khai báo sau :

❖ **Cách 1: Con trỏ hằng**

`char < Tên chuỗi > [< Số ký tự tối đa của chuỗi >] ;`

Ví dụ: `char chuoi[25];`

Ý nghĩa khai báo **1 mảng kiểu ký tự tên là chuoi** có 25 phần tử (như vậy tối đa ta có thể nhập 24 ký tự vì **phần tử thứ 25 đã chứa ký tự kết thúc chuỗi '\0'**)

❖ **Cách 2: Con trỏ**

`char * < Tên chuỗi >;`

Ví dụ : `char *chuoi;`

I.3. Các thao tác trên chuỗi

a. Nhập chuỗi

Cú pháp : `char *gets(char *s);`

Nhận các ký tự nhập từ phím cho đến khi nhấn phím Enter và đưa vào s.

Ví dụ:


```
void main()
{
    char chuoi[80];
    printf("Nhap vao chuoi:");
    gets(chuoi);
    printf("Chuoi vua nhap la: %s\n", chuoi);
}
```

b. Xuất chuỗi

Cú pháp : int puts(const char *s);

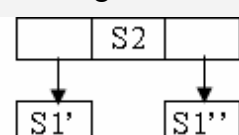
Xuất chuỗi s ra màn hình.

Ví dụ:

```
void main()
{
    char chuoi[] = "Vi du xuat chuoi\n";
    puts(string);
}
```

c. Các hàm thư viện (string.h)

STT	TÊN HÀM	CHỨC NĂNG	VÍ DỤ
1	int strlen(char s[]);	Trả về độ dài của chuỗi s.	char *s = "Borland International"; printf("Do dai s: %d\n", strlen(s)); Kết quả: Do dai s: 21
2	strcpy(char dest[], char src[]);	Sao chép nội dung chuỗi src vào chuỗi dest.	char dest[10]; char *src = "abcdefghi"; strcpy(dest, src); printf("%s\n", dest); Kết quả: abcdefghi
3	strncpy(char dest[], char src[], int n);	Chép n ký tự từ chuỗi src sang chuỗi dest. Nếu chiều dài src < n thì hàm sẽ điền khoảng trắng cho đủ n ký tự vào dest.	char dest[4]; char *src = "abcdefghi"; strncpy(dest, src, 3); printf("%s\n", dest); Kết quả: abc
4	strcat(char s1[],char s2[]);	Nối chuỗi s2 vào chuỗi s1.	char *s1 = "Khoa "; char *s2 = "CNTT"; strcat(s1, s2); printf("%s\n", s1); Kết quả: Khoa CNTT
5	strncat(char s1[],char s2[],int n)	Nối n ký tự đầu tiên của chuỗi s2 vào chuỗi s1.	char *s1 = "Khoa "; char *s2 = "CNTT"; strncat(s1, s2, 2); printf("%s\n", s1); Kết quả: Khoa CN
6	Int strcmp(char s1[],char s2[])	So sánh 2 chuỗi s1 và s2 theo nguyên tắc <u>thứ tự từ điển</u> . Phân biệt chữ hoa và thường.	char *s1 = "abcd"; char *s2 = "abCD"; if(strcmp(s1, s2)==0)

		<p>Trả về:</p> <ul style="list-style-type: none"> • 0 : nếu s1 bằng s2. • >0: nếu s1 lớn hơn s2. • <0: nếu s1 nhỏ hơn s2. 	<pre>printf("Giống nhau"); else printf("Khac nhau"); Kết quả: Khac nhau</pre>
7	int strcmp(char s1[], char s2[], int n)	Tương tự như strcmp(), nhưng chỉ so sánh n ký tự đầu tiên của hai chuỗi.	<pre>char *s1 = "abcd"; char *s2 = "abef"; if(strcmp(s1, s2, 2)==0) printf("Giống nhau"); else printf("Khac nhau"); Kết quả: Giống nhau</pre>
6	int stricmp(char s1[], char s2[])	Tương tự như strcmp(), nhưng không phân biệt hoa thường.	<pre>char *s1 = "abcd"; char *s2 = "abCD"; if(stricmp(s1, s2)==0) printf("Giống nhau"); else printf("Khac nhau"); Kết quả: Giống nhau</pre>
7	int strnicmp(char s1[], char s2[], int n);	Tương tự như strcmp(), nhưng chỉ so sánh n ký tự đầu tiên của hai chuỗi.	<pre>char *s1 = "aBcd"; char *s2 = "Abef"; if(strnicmp(s1, s2, 2)==0) printf("Giống nhau"); else printf("Khac nhau"); Kết quả: Giống nhau</pre>
8	char *strchr(char s[], char c);	<p>Tìm lần xuất hiện đầu tiên của ký tự c trong chuỗi s. Trả về:</p> <ul style="list-style-type: none"> • NULL: nếu không có. • Địa chỉ c: nếu tìm thấy. 	<pre>char s[15]; char *ptr, c = 'm'; strcpy(s, "Vi du tim ky tu"); ptr = strchr(s, c); if (ptr) printf("Ky tu %c tai: %d", c, ptr-s); else printf("Khong tim thay"); t quả: Ky tu m tai: 8</pre>
9	char *strstr(char s1[], char s2[]);	<p>Tìm sự xuất hiện đầu tiên của chuỗi s2 trong chuỗi s1. Trả về:</p> <ul style="list-style-type: none"> • NULL: nếu không có. • Ngược lại: Địa chỉ bắt đầu chuỗi s2 trong s1. 	<pre>char *s1 = "Borland International"; char *s2 = "nation", *ptr; ptr = strstr(s1, s2); printf("Chuoi con: %s\n", ptr); Kết quả: Chuoi con: national</pre>
10	char *strtok(char s1[], char s2[]);	<ul style="list-style-type: none"> • Nếu s2 có xuất hiện trong s1: Tách chuỗi s1 thành hai chuỗi: Chuỗi đầu là những ký tự cho đến khi gặp chuỗi s2 đầu tiên, chuỗi sau là những ký tự còn lại của s1 sau khi đã bỏ đi chuỗi s2 xuất hiện trong s1. <p>S1: </p>	<pre>char input[16] = "abc,d"; char *p; // Lay chuoi dau p = strtok(input, ","); if (p) printf("S11: %s\n", p); /*Lay chuoi con lai, tham so dau la NULL*/ p = strtok(NULL, ","); if (p) printf("S12: %s\n", p); Kết quả: <u>S11: abc</u> <u>S12: d</u></pre>

		<ul style="list-style-type: none"> Nếu s2 không xuất hiện trong s1 thì kết quả chuỗi tách vẫn là s1. 	
<p>🔗 Lưu ý: Cách truy xuất các ký tự tương tự như mảng một chiều.</p>			

d. Ví dụ

Nhập vào một chuỗi ký tự, xuất ra màn hình chuỗi bị đảo ngược thứ tự các ký tự.

Ví dụ: Nhập vào: *Tran minh thai*. Xuất ra màn hình: *iaht hnim narT*

```
#include<stdio.h>
#include<string.h>
#include<conio.h>

void DaoChuoi(char *s1, char *s2)
{
    int l=strlen(s1);
    for(int i=0; i<l; i++)
        s2[i]=s1[l-i-1];

    s2[i]='\0';
}

void main()
{
    char *s1, *s2;
    clrscr();
    printf("\nNhap vao chuoi ky tu: ");
    gets(s1);
    DaoChuoi(s1, s2);
    printf("\nKet qua sau khi dao nguoc chuoi: %s", s2);
}
```

II. BÀI TẬP

II.1. Bài tập cơ bản

- Cho biết kết quả của đoạn chương trình sau:

```
char input[20]="Truong cao dang CNTT", *p, *temp;
strcpy(temp, input);
do
{
    p = strtok(temp, " ");
    printf("%s\n",p);
```

```

    p = strtok(NULL, "");
    strcpy(temp, p);
}while(p!=NULL);
printf("Chuoi temp: %s \n Chuoi input: %s", temp, input);

```

2. Cho biết kết quả của đoạn chương trình sau:

```

char s1[20]="Truong cao dang CNTT", s1[10]="Tp. HCM", *input, *s3;
strcpy(input, s1);    strcpy(s3,"aeiou"); strcat(input, s2);
int n=strlen(input), k=0;
printf("Chuoi: %s",input);
for(int i=0; i<n; i++)
{
    if(strchr(s3, input[i]))
        k++;
}
printf("\nKet qua: %d", k);

```

3. Viết chương trình nhập vào một chuỗi ký tự, đếm số ký tự có trong chuỗi.
4. Viết chương trình đếm có bao nhiêu khoảng trắng trong chuỗi.
5. Viết chương trình nhập vào một chuỗi, hãy loại bỏ những khoảng trắng thừa trong chuỗi.
6. Viết chương trình nhập vào hai chuỗi s1 và s2, nối chuỗi s2 vào s1. Xuất chuỗi s1 ra màn hình.
7. Đổi tất cả các ký tự có trong chuỗi thành chữ thường (không dùng hàm `strlwr`).
8. Đổi tất cả các ký tự trong chuỗi sang chữ in hoa (không dùng hàm `struppr`).
9. Viết chương trình đổi những ký tự đầu tiên của mỗi từ thành chữ in hoa.
10. Viết chương trình đổi chữ xen kẽ 1 chữ hoa và 1 chữ thường.
Ví dụ: nhập ABCDEfgh đổi thành AbCdEfGh
11. Viết chương trình đảo ngược các ký tự trong chuỗi .
Ví dụ: nhập ABCDE, xuất ra màn hình là:EDCBA
12. Viết chương trình tìm kiếm 1 ký tự xem có trong chuỗi hay không, nếu có xuất ra vị trí của từ đó.
13. Viết 1 chương trình đếm một ký tự xuất hiện bao nhiêu lần trong chuỗi.

14. Viết chương trình tìm kiếm tên trong chuỗi họ tên. Nếu có thì xuất ra là tên này đã nhập đúng, ngược lại thông báo là đã nhập sai.
15. Viết chương đảo vị trí của từ đầu và từ cuối.
Ví dụ: nhập “bo an co” xuất ra “co an bo”
16. Viết hàm cắt chuỗi họ tên thành chuỗi họ lót và chuỗi tên.
Ví dụ: chuỗi họ tên là: “Nguyễn Văn A” cắt ra 2 chuỗi là chuỗi họ lót: “Nguyễn Văn”, chuỗi tên là: “A”
17. Nhập một chuỗi bất kỳ, sau đó hỏi người dùng cần tách bắt đầu từ đâu trong chuỗi trở về sau.
Ví dụ: Nhập chuỗi S1: “trường Cao Đẳng Công Nghệ Thông tin”. Người nhập muốn tách bắt đầu từ chữ “Công” thì sẽ xuất ra chuỗi “Công Nghệ Thông Tin” ra màn hình.
18. Viết hàm kiểm tra xem chuỗi có đối xứng hay không?.
19. Viết hàm tra xem trong chuỗi có ký tự số hay không nếu có tách ra thành một mảng số riêng.
20. Nhập một chuỗi bất kì, yêu cầu nhập 1 ký tự muốn xóa. Thực hiện xóa tất cả những ký tự đó trong chuỗi.
21. Viết chương trình tìm kiếm xem ký tự nào xuất hiện nhiều nhất trong chuỗi.
22. Viết 1 chương trình xoá một từ nào đó trong chuỗi.
Ví dụ: Chuỗi ban đầu: “CAO DANG CNTT”
Nhập: “CNTT”, và kết quả xuất ra: “CAO DANG”

II.2. Bài tập luyện tập và nâng cao

23. Đổi các từ ở đầu câu sang chữ hoa và những từ không phải đầu câu sang chữ thường.
Ví dụ: nGuYen vAN a đổi thành: Nguyễn Văn A
24. (*) Viết chương trình đảo ngược thứ tự các từ có trong chuỗi
Ví dụ: Nhập Truong CD CNTT TpHCM
Xuất ra màn hình là: TpHCM CNTT CD Truong
25. Nhập 1 chuỗi bất kì, liệt kê xem mỗi ký tự xuất hiện mấy lần.
26. Viết hàm kiểm tra xem trong 2 chuỗi có bao nhiêu ký tự giống nhau.
27. Viết chương trình mình chạy từ trái qua phải màn hình.

28. Viết 1 chương trình chèn 1 từ ở bất cứ vị trí nào mà người dùng yêu cầu.
29. (*) Viết chương trình nhập vào một chuỗi đếm xem chuỗi có bao nhiêu từ. Các từ cách nhau bằng khoảng trắng, dấu chấm câu: dấu chấm (.), dấu phẩy (,), dấu chấm phẩy (;), dấu hỏi (?) và dấu chấm than (!).
30. (**) Viết chương trình hiển thị một chuỗi ký tự. Chương trình cho phép di chuyển dấu nháy sang trái, sang phải, lên dòng hay xuống dòng bằng phím mũi tên, chèn hay xoá ký tự tại vị trí dấu nháy.

III. KẾT LUẬN

- ❖ Cũng giống như kiểu mảng một chiều, thao tác truy xuất các phần tử trên chuỗi hoàn toàn tương tự. Bên cạnh đó, kiểu dữ liệu này còn được cài đặt sẵn một số hàm thư viện rất hữu ích nên trong quá trình thao tác trên chuỗi nên khi cài đặt ta cố gắng **tận dụng tối đa những hàm liên quan**.
- ❖ **Không nên sử dụng hàm scanf()** để nhập chuỗi trong trường hợp chuỗi dữ liệu nhập vào có chứa khoảng trắng.
- ❖ Nếu **nhập chuỗi phía sau hàm scanf()** nên chèn hàm **fflush(stdin)** hoặc hàm **flushall()** giữa **scanf** và **gets()** để xóa vùng đệm, tránh trường hợp chương trình **bỏ qua hàm gets()** do trong vùng đệm còn lưu ký tự xuống dòng của phím **ENTER**.
- ❖ Khi thao tác trên chuỗi lưu ý phải **đảm bảo chuỗi được kết thúc bằng ký tự kết thúc '\0'**.

CHƯƠNG 6 MẢNG HAI CHIỀU

Đây là kiểu dữ liệu dùng để biểu diễn dữ liệu kiểu bảng, kiểu dữ liệu này rất thích hợp cho các bài toán liên quan đến đồ thị, biểu diễn ảnh, ...

I. TÓM TẮT LÝ THUYẾT

I.1. Khái niệm

Mảng hai chiều thực chất là mảng một chiều trong đó mỗi phần tử của mảng là một mảng một chiều, và được truy xuất bởi hai chỉ số dòng và cột.

Từ khái niệm trên ta có thể đưa ra một khái niệm về mảng nhiều chiều như sau:
mảng có từ hai chiều trở lên gọi là mảng nhiều chiều.

I.2. Khai báo mảng

Từ khái niệm trên ta có cú pháp khai báo mảng hai chiều như sau:

- Cách 1: Con trỏ hằng

< Kiểu dữ liệu > < Tên mảng > [< Số dòng tối đa >][< Số cột tối đa >];

Ví dụ:

```
int A[10][10];    // Khai báo mảng 2 chiều kiểu int gồm 10 dòng, 10 cột
float b[10][10]; // Khai báo mảng 2 chiều kiểu float gồm 10 dòng, 10 cột
```

- Cách 2 : Con trỏ

< Kiểu dữ liệu > **<Tên mảng>;

Ví dụ :

```
int **A ; // Khai báo mảng động 2 chiều kiểu int
float **B ; // Khai báo mảng động 2 chiều kiểu float
```

Tương tự như mảng một chiều, để sử dụng ta phải cấp phát vùng nhớ cho nó bằng malloc hoặc calloc và huỷ sau khi dùng bằng free

Ví dụ : Khai báo mảng các số nguyên A có kích thước 5x6

```
int **A;
A = (int **) malloc (5) ;
for (int i = 0 ; i < 5 ; i ++)
    A[i]=(int *) malloc (6) ;
```

I.3. Truy xuất phần tử của mảng

Để truy xuất các thành phần của mảng hai chiều ta phải dựa vào chỉ số dòng và chỉ số cột.

Ví dụ:

$int\ A[3][4] = \{ \{2,3,9,4\}, \{5,6,7,6\}, \{2,9,4,7\} \};$

Với các khai báo như trên ta có :

$A[0][0] = 2; A[0][1] = 3;$

$A[1][1] = 6; A[1][3] = 6;$

Với ví dụ trên ta có hình dạng của một ma trận như sau

	0	1	2	3
0	2	3	9	4
1	5	6	7	6
2	2	9	4	7

⚠ Lưu ý: Khi nhập liệu cho mảng hai chiều, nếu là mảng các số nguyên thì ta nhập liệu theo cách thông thường. Nhưng nếu là mảng các số thực thì ta phải thông qua biến trung gian.

Ví dụ :

```
float a[10][10];           // Mảng số thực a
float tmp;                // Biến trung gian tmp
scanf ("%f", &tmp);      // Nhập liệu cho biến trung gian
a[2][2] = tmp;            // Gán dữ liệu vào phần tử a[2][2]
```

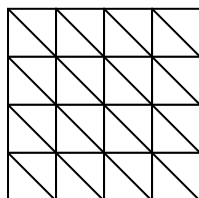
I.4. Ma trận vuông và các khái niệm liên quan

a. Khái niệm

Là ma trận có số dòng và số cột bằng nhau.

b. Tính chất của ma trận vuông

- **Đường chéo loại 1**



- Đường chéo loại 1 bao gồm đường chéo chính và những đường chéo song song với đường chéo chính. Trong đó đường chéo chính là đường chéo có :

chỉ số dòng = chỉ số cột

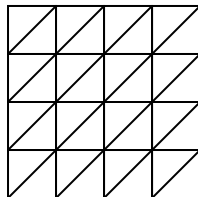
- Truy xuất các phần tử trên đường chéo loại 1 : để truy xuất các phần tử trên các đường chéo loại 1 ta có thể dựa vào chỉ số dòng và chỉ số cột như sau :

$$\text{cột} - \text{dòng} = \text{hằng số}$$

Ví dụ : Cho ma trận vuông $A(n \times n)$. Gọi (i_0, j_0) là tọa độ điểm xuất phát, ta có thể duyệt đường chéo xuất phát từ (i_0, j_0) như sau :

```
for ( i = i0, j = j0; i < n ; i ++, j ++ )
    printf ("%4d", A[i][j]);
```

- **Đường chéo loại 2:**



- Đường chéo loại 2 bao gồm đường chéo phụ và những đường song song với nó. Trong đó đường chéo phụ là đường chéo có:

$$\text{chỉ số cột} + \text{chỉ số dòng} = \text{số dòng (hoặc số cột)}$$

- Truy xuất các phần tử trên đường chéo loại 2 : để truy xuất các phần tử trên các đường chéo loại 1 ta có thể dựa vào chỉ số dòng và chỉ số cột như sau :

$$\text{cột} + \text{dòng} = \text{hằng số}$$

Ví dụ: Cho ma trận vuông $A(n \times n)$. Gọi (i_0, j_0) là tọa độ điểm xuất phát, ta có thể duyệt đường chéo xuất phát từ (i_0, j_0) như sau :

```
for ( i = i0, j = j0; i < n && j >= 0 ; i ++, j -- )
    printf ("%4d", A[i][j]);
```

II. BÀI TẬP

Để đơn giản trong việc khai báo ma trận, ta định nghĩa **kiểu ma trận các phần tử với kiểu dữ liệu bất kỳ** như sau:

```
#define MAX 100
typedef <kiểu dữ liệu> MATRAN[MAX][MAX];
```

Ví dụ: Khai báo ma trận các số nguyên a.

```
#define MAX 100
```

```
typedef int MATRAN[MAX][MAX];
MATRAN a;
```

II.1. Một số kỹ thuật cơ bản

- Phương pháp nhập xuất ma trận

```
void Nhap (MATRAN a, int &d, int &c)
{
    printf ("\nNhap so dong: ");
    scanf ("%d", &d);
    printf ("\nNhap so cot: ");
    scanf ("%d", &c);
    for (int i = 0; i < d; i++)
        for (int j = 0; j < c; j++)
            {
                printf (" a[%d][%d] = ", i, j);
                scanf ("%d", &a[i][j]);
            }
}

void Xuat (MATRAN a, int d, int c)
{
    printf ("\nNoi dung ma tran:\n");
    for (int i = 0; i < d; i++)
        {
            for (int j = 0; j < c; j++)
                printf ("\t %d ", a[i][j]);
            printf ("\n");
        }
}
```

- Kỹ thuật đặt cờ hiệu

Viết hàm kiểm tra xem trong ma trận các số nguyên có tồn tại các số nguyên lẻ lớn hơn 100 không?

```
int KiemTraLe (MATRAN a, int d, int c)
{
    int flag = 0; //tra ve 1 neu co nguoc lai tra ve 0

    for (int i = 0; i < d; i++)
        for (int j = 0; j < c; j++)
            if ( a[i][j] % 2 != 0 && a[i][j] > 100 )
                {
                    flag = 1;
                    break;
                }
    return flag;
}
```

- **Kĩ thuật đặt lính canh**

Viết hàm tìm phần tử nhỏ nhất trong ma trận.

```
int Min (MATRAN a, int d, int c)
{
    int min = a[0][0];
    for (int i = 0; i < d; i++)
        for (int j = 0; j < c; j++)
            if (a[i][j] < min)
                min = a[i][j];
    return min;
}
```

- **Phương pháp tính tổng**

Viết hàm tính tổng các phần tử trong ma trận.

```
long Tong (MATRAN a, int d, int c)
{
    long tong = 0;

    for (int i = 0; i < d; i++)
        for (int j = 0; j < c; j++)
            tong += a[i][j];
    return tong;
}
```

- **Phương pháp sắp xếp**

Viết hàm sắp xếp ma trận tăng dần từ trên xuống dưới và từ trái sang phải không dùng mảng phụ.

```
void SapTang(MATRAN a, int d, int c)
{
    for (int i = 0; i <= d*c-2; i++)
        for (int j = 0; j <= d*c-1; j++)
            if (a[i/c][i%c] < a[j/c][j%c])
            {
                int tmp = a[i/c][i%c];
                a[i/c][i%c] = a[j/c][j%c];
                a[j/c][j%c] = tmp;
            }
}
```

- **Phương pháp đếm**

Viết hàm đếm các phần tử chẵn trong ma trận.

```
int DemChan (MATRAN a, int d, int c)
{
    int dem = 0;
```

```

for ( int i = 0 ; i < d ; i ++ )
    for ( int j = 0 ; j < c ; j ++ )
        if ( a[i][j] % 2 == 0 )
            dem ++;
return dem;
}

```

II.2. Bài tập cơ bản

a. Bài tập nhập xuất

- Viết hàm nhập ma trận các số nguyên dương (*nhập sai báo lỗi và không cho nhập*).
- Viết hàm nhập/ xuất ma trận các số thực.
- Viết hàm in ra những phần tử có ký số tận cùng là 5.
- Viết chương trình in ra các phần tử nằm trên 2 đường chéo.
- Viết hàm in ra các phần tử nằm phía trên đường chéo phụ của ma trận vuông các số nguyên.
- Viết hàm in ra các phần tử nằm phía dưới đường chéo phụ của ma trận vuông các số nguyên.
- Viết hàm in ra các phần tử nằm phía trên đường chéo chính của ma trận vuông các số nguyên.
- Viết hàm in ra các phần tử nằm phía dưới đường chéo chính của ma trận vuông các số nguyên.
- Viết chương trình khởi tạo giá trị các phần tử là ngẫu nhiên cho ma trận các số nguyên kích thước $m \times n$.
- Viết hàm tạo ma trận a các số nguyên gồm 9 dòng 14 cột. Trong đó phần tử $a[i][j] = i * j$
- Viết hàm in tam giác Pascal với chiều cao h.

Ví dụ : h = 5

```

1
1   1
1   2   1
1   3   3   1
1   4   6   4   1

```

b. Bài tập tính tổng

- Viết hàm tính tổng các phần tử trên cùng một dòng.

13. Viết hàm tính tổng các phần tử trên cùng một cột.
14. Viết hàm tính tổng các phần tử chẵn có trong ma trận.
15. Viết hàm tính tổng các phần tử nằm trên đường chéo chính của ma trận vuông.
16. Viết hàm tính tổng các phần tử là số nguyên tố có trong ma trận.
17. Viết hàm tính tổng các số hoàn thiện trong ma trận các số nguyên.
18. Viết hàm tính tổng các giá trị lớn nhất trên mỗi dòng.
19. Viết hàm tính giá trị trung bình của các phần tử nhỏ nhất trên mỗi cột.
20. Viết hàm tính tổng các giá trị nhỏ nhất nằm trên từng đường chéo loại 2.
21. Viết hàm tìm đường chéo có tổng lớn nhất trong các đường chéo loại 1.

c. Bài tập tìm kiếm

22. Viết hàm tìm vị trí phần tử lớn nhất trong ma trận các số nguyên.
23. Viết hàm tìm vị trí phần tử nhỏ nhất trong ma trận các số nguyên.
24. Viết hàm tìm vị trí phần tử chẵn cuối cùng trong ma trận các số nguyên.
25. Viết hàm tìm phần tử âm lẻ lớn nhất trong ma trận.
26. Viết hàm tìm phần tử chẵn dương và nhỏ nhất trong ma trận.
27. Viết hàm tìm số hoàn thiện đầu tiên trong ma trận các số nguyên.
28. Viết hàm tìm số hoàn thiện lớn nhất trong ma trận các số nguyên.
29. Viết hàm tìm vị trí phần tử nguyên tố cuối cùng trong ma trận các số nguyên.
30. Viết hàm tìm phần tử lớn nhất nằm trên đường chéo chính của ma trận vuông.
31. Viết hàm in các số nguyên tố nằm trên đường chéo phụ của ma trận vuông.
32. Viết hàm tìm trong 2 ma trận các số nguyên, những phần tử giống nhau.
33. Viết hàm tìm phần tử nhỏ nhất trên mỗi đường chéo loại 2 của ma trận.
34. Viết hàm tìm và liệt kê những phần tử cực đại trong ma trận (một phần tử được coi là cực đại khi nó lớn hơn các phần tử xung quanh nó).
35. Viết hàm tìm dòng có tổng lớn nhất trong ma trận các số thực.
36. Viết hàm tìm cột có tổng nhỏ nhất trong ma trận các số nguyên.

d. Bài tập đếm

37. Viết hàm đếm các giá trị âm, dương trong ma trận các số thực.
38. Viết hàm đếm các giá trị chẵn, lẻ trong ma trận các số nguyên.

39. Viết hàm đếm số lần xuất hiện của phần tử x trong ma trận các số thực.
40. Viết hàm đếm các giá trị nhỏ hơn x trong ma trận các số thực.
41. Viết hàm đếm các phần tử nguyên tố trong ma trận các số nguyên.
42. Viết hàm đến các phần tử nguyên tố trên đường chéo chính của ma trận vuông các số nguyên.
43. Viết hàm đếm các giá trị chẵn trên đường chéo chính của ma trận vuông các số nguyên.
44. Viết hàm đếm các giá trị là bội của 3 và 5 trên đường chéo chính của ma trận các số nguyên.
45. Viết hàm đếm các giá trị nguyên tố trên 2 đường chéo (chính, phụ) của ma trận vuông các số nguyên.
46. Viết hàm đếm các giá trị cực đại trong ma trận các số nguyên.
47. Viết hàm đếm các giá trị cực tiểu trong ma trận các số nguyên.
48. Viết hàm đếm các cực trị trong ma trận các số nguyên (một phần tử được coi là cực trị khi nó là giá trị cực đại hay cực tiểu).
49. Viết hàm đếm các giá trị là số hoàn thiện trong ma trận các số nguyên.

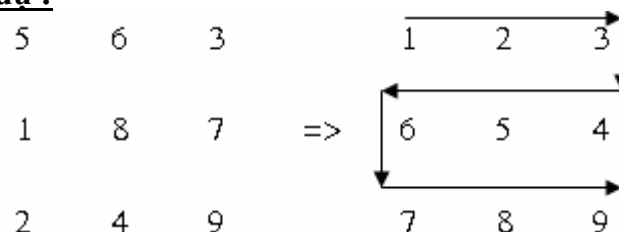
e. Bài tập sắp xếp

50. Viết hàm sắp xếp ma trận theo thứ tự tăng dần từ trên xuống dưới và từ trái qua phải theo phương pháp dùng mảng phụ.

Hướng dẫn: Đổ ma trận sang mảng một chiều, sắp xếp trên mảng một chiều theo thứ tự tăng dần, sau đó chuyển ngược mảng một chiều thành ma trận kết quả.

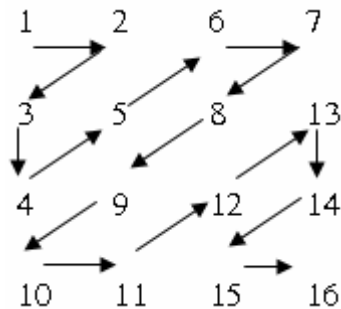
51. Viết hàm sắp xếp ma trận theo thứ tự giảm dần từ trên xuống dưới và từ trái sang phải.
52. Viết hàm sắp xếp các dòng trên ma trận theo thứ tự tăng dần.
53. Viết hàm sắp xếp các cột trên ma trận theo thứ tự giảm dần.
54. Viết hàm sắp xếp ma trận theo đường ziczắc ngang.

Ví dụ :



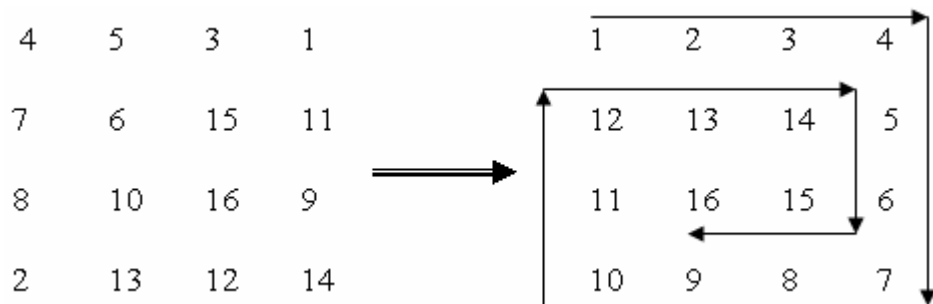
55. Viết hàm sắp xếp ma trận theo đường ziczắc chéo

Ví dụ :



56. Viết hàm sắp xếp ma trận theo đường xoắn ốc từ ngoài vào trong theo chiều kim đồng hồ.

Ví dụ :



57. Cho ma trận vuông, viết hàm sắp xếp tăng dần các phần tử nằm trên các đường chéo song song với đường chéo chính.
58. Viết chương trình nhập một ma trận vuông các số nguyên, và thực hiện những công việc sau :
- Sắp xếp các phần tử nằm trên các đường chéo loại 1 tăng dần
 - Sắp xếp các phần tử nằm trên các đường chéo loại 2 giảm dần.
 - Sắp xếp với điều kiện: các phần tử trên đường chéo chính tăng, các phần tử trên các đường chéo song song với đường chéo chính giảm.

f. Bài tập Thêm – Xoá – Thay thế

59. Viết hàm xoá một dòng i trên ma trận.
60. Viết hàm xoá một cột j trên ma trận.
61. Viết hàm xoá dòng có tổng lớn nhất trên ma trận.
62. Viết hàm hoán vị dòng có tổng lớn nhất với dòng có tổng nhỏ nhất.
63. Viết hàm tìm và thay thế các phần tử chẵn trong ma trận bằng ước số nhỏ nhất của nó.
64. Viết hàm thay thế những phần tử có giá trị x thành phần tử có giá trị y trong ma trận (x , y nhập từ bàn phím).

II.3. Bài tập luyện tập và nâng cao

65. Viết chương trình tính tổng, tích của hai ma trận các số nguyên.
66. Viết hàm kiểm tra xem ma trận vuông các số nguyên có đối xứng qua đường chéo chính hay không.
67. Viết hàm kiểm tra xem trong ma trận vuông cấp n có hàng nào trùng nhau hay không, nếu có thì chỉ rõ những hàng nào. (Trùng giá trị và vị trí).
68. Viết chương trình nhập vào ma trận vuông kích thước $n \times n$ ($2 \leq n \leq 100$). Hãy viết hàm thực hiện những công việc sau :
- In ra các phần tử trên 4 đường biên của ma trận.
 - Tính tổng các phần tử trên biên.
69. (*) Viết chương trình xoay ma trận các số thực 90^0 ngược chiều kim đồng hồ.

Ví dụ:

1	2	3	4		4	8	12	16
5	6	7	8	=>	3	7	11	15
9	10	11	12		2	6	10	14
13	14	15	16		1	5	9	13

70. Viết chương trình dịch phải xoay vòng một cột trong ma trận các số thực.
71. Viết chương trình dịch xuống xoay vòng một dòng trong ma trận các số thực.
72. (*) Cho ma trận A ($m \times n$) các số nguyên hãy phát sinh ma trận B sao cho B là ma trận lật ngược của ma trận A .

Ví dụ :

1	2	3	4		4	3	2	1
5	6	7	8	=>	8	7	6	5
9	10	11	12		12	11	10	9
13	14	15	16		16	15	14	13

73. (**) Cho ma trận A ($m \times n$) hãy phát sinh ma trận B ($m \times n$) sao cho phần tử $B(i, j)$ là trung bình cộng của các phần tử trong hình vuông 3×3 tâm tại (i, j) của A .

Ví dụ :

1	5	2	6		3	2	4	4
4	2	3	6	=>	4	4	4	4
8	7	9	1		5	6	6	7
10	2	12	13		6	8	7	8

74. (**) Cho ma trận các số nguyên dương $A (m \times n)$. Hãy xây dựng ma trận $B (m \times n)$. Sao cho phần tử $B (i, j)$ là số lớn nhất trong ô vuông 3×3 tâm tại (i, j) của A .

Ví dụ :

1	5	2	6		5	5	6	6
4	2	3	6	=>	8	9	9	9
8	7	9	1		10	12	13	13
10	2	12	13		10	12	13	13

75. (**) Cho ma trận $A (m \times n)$. Hãy xây dựng ma trận $B (m \times n)$ với phần tử $B(i,j)$ được xác định theo qui tắc sau: tại vị trí (i, j) trên mảng A kẻ hai tia vuông góc với nhau, tạo thành với trục hoành một góc 45^0 từ trên xuống dưới; $B(i, j)$ là tổng của tất cả các số của vùng mặt phẳng tạo bởi hai tia này và các cạnh của bảng.

Ví dụ :

1	3	2		25	30	23
6	4	3	=>	13	18	15
2	5	7		2	5	7

76. (**) Cho ma trận vuông $A (n \times n)$. Hãy xây dựng mảng $B (n \times n)$ bằng cách: phần tử $B (i, j)$ là số lớn nhất trong tam giác vuông vẽ từ $A (i, j)$ tới đường chéo chính.

Ví dụ :

1	3	2		1	4	7
6	4	3	=>	6	4	7
2	5	7		7	7	7

77. (*) Viết chương trình hiển thị đồng hồ điện tử (gồm giờ phút), với giờ lấy từ hệ thống và đồng hồ được cập nhật theo phút.

Hướng dẫn: Tạo 1 ma trận giá trị gồm 0 hoặc 1, vị trí nào cần hiển thị thì gán giá trị là 1, ngược lại có giá trị là 0. Sau mỗi phút cập nhật lại ma trận và hiển thị lên màn hình.

Ví dụ: 01 giờ 25 phút

1 1 1 1	1 1		1 1 1 1 1 1 1	1 1 1 1 1 1 1
1 1	1 1	1 1	1 1	1 1 1 1
1 1	1 1	1 1		1 1 1 1 1 1 1 1 1 1 1 1
1 1	1 1	1 1	1 1	1 1 1 1 1 1 1 1
1 1 1 1	1 1		1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1

78. Nhập vào mảng hai chiều gồm n dòng và m cột các số nguyên. Hãy tìm phần tử lớn nhất trên mỗi dòng và đồng thời nhỏ nhất trên mỗi cột, hoặc lớn nhất trên mỗi cột và đồng thời nhỏ nhất trên mỗi dòng. Có bao nhiêu phần tử như thế?

Ví dụ:

3	6	2	1
4	7	6	9
5	15	8	7

79. Viết chương trình tạo ngẫu nhiên một ma trận các số nguyên (0 -> 50), tìm những phần tử cực đại (là phần tử lớn hơn các phần tử xung quanh).

Ví dụ :

2	6	8	4
9	7	5	3
6	2	8	1

80. (**) Cho ma trận các số nguyên $A_{m \times n}$ ($n \geq 3, m \geq 3$). Hãy tìm ma trận con (3x3) có tổng lớn nhất.

Ví dụ :

1	2	3	4		6	7	8
5	6	7	8	=>	10	11	12
9	10	11	12		14	15	16
13	14	15	16				

81. Nhập ma trận vuông cấp $n \times n$ ($n < 10$). In ra các phần tử của ma trận này theo hướng của đường chéo chính.

Ví dụ : $n = 4$

					1			
					9	3		
1	3	7	4		3	5	7	
9	5	6	2	=>	2	4	6	4
3	4	7	5		3	7	2	
2	3	1	6		1	5		
					6			

82. (**) Hãy điền các số từ 1 đến n^2 vào ma trận cấp n ($n > 2$), chỉ xét trường hợp n là số lẻ với tính chất P là tổng các số bằng nhau.

Hướng dẫn : Ma phương của một bảng vuông cấp n , trong mỗi ô nhận một giá trị sao cho, mỗi hàng, mỗi cột và mỗi đường chéo đều thỏa mãn một tính chất P nào đó cho trước.

Ví dụ : Với $n = 5$

1	18	25	2	9
10	12	19	21	3
4	6	13	20	22
23	5	7	14	16
17	24	1	8	15

83. (*) Viết hàm in ma trận các số nguyên dương theo qui luật được mô tả như sau : các phần tử phía trên đường chéo phụ là giá trị bình phương của các giá trị $1 \rightarrow n \times 2$, các giá trị từ đường chéo phụ trở xuống là các số nguyên tố. Ma trận được sắp xếp như ví dụ bên dưới.

Ví dụ : $n = 5$

1	9	36	100	31
4	25	81	37	17
16	64	41	19	7
49	43	23	11	3
47	29	13	5	2

84. Cho ma trận vuông a cấp n (n lẻ, $3 \leq n \leq 15$), mỗi phần tử đều có giá trị nguyên dương. Hãy xây dựng hàm kiểm tra xem ma trận a có phải là ma phương hay không?
85. (***) Viết chương trình giải bài toán 8 hậu. Hãy đặt 8 con hậu trên bàn cờ 8×8 sao cho chúng không ăn nhau (2 hậu ăn nhau khi cùng hàng, cùng cột và cùng nằm trên đường chéo).

Hướng dẫn:

Dùng ma trận 8×8 để lưu bàn cờ. Mỗi ô có 3 trạng thái :

- Có hậu 1
- Ô trống 0
- Ô không được đi -1

86. (***) Viết chương trình giải bài toán mã đi tuần. Hãy đi con mã 64 lượt đi trên bàn cờ 8×8 sao cho mỗi ô chỉ đi qua một lần (xuất phát từ một ô bất kỳ)

Hướng dẫn :

Đứng tại một ô trên bàn cờ con mã có thể đi được 1 trong 8 hướng sau .

		1		2			
	8				3		
			•				
	7				4		
		6		5			

Khai báo 8 hướng đi của mã như sau:

```
typedef struct DIEM
```

```
{
```

```
    int x, y;
```

```
};
```

$DIEM\ huongdi[8]=\{\{-2,-1\},\{-2,1\},\{-1,2\},\{1,2\},\{2,1\},\{2,-1\},\{1,-2\},\{-1,-2\}\};$

Trong đó mỗi thành phần của *huongdi* là độ lệch của dòng và cột so với vị trí của con mã.

Ví dụ: *huongdi[0]* (tức đi đến vị trí 1 như hình vẽ) có độ lệch 2 dòng và 1 cột. (Giá trị âm biểu thị độ lệch về bên trái cột hay hướng lên của dòng).

Chọn vị trí đi kế tiếp sao cho vị trí đó phải gần với biên hay góc nhất (tức số đường đi có thể đi là ít nhất).

87. Viết chương trình giải bài toán Taci. Cho ma trận vuông 3x3 gồm các số nguyên từ 0 -> 8 trong đó 0 là ô trống. Bài toán đặt ra là hãy đưa ma trận ở một trạng thái đầu về trạng thái đích, mỗi lần chỉ dịch chuyển được 1 ô.

Ví dụ: Trạng thái đầu

1	3	0
8	2	5
7	4	6

Trạng thái đích

1	2	3
8	0	4
7	6	5

=>

III. KẾT LUẬN

- ❖ Kiểu dữ liệu mảng hai chiều được ứng dụng rộng rãi trong các bài toán về tìm đường đi trong đồ thị, xử lý ảnh, xử lý những dữ liệu dạng bảng, ...
- ❖ Lưu ý khi nhập mảng hai chiều **các số thực** phải thông qua 1 biến trung gian.

CHƯƠNG 7 KIỂU DỮ LIỆU CÓ CẤU TRÚC

Cung cấp cơ chế cho phép khai báo các kiểu dữ liệu mới để giải quyết theo yêu cầu của bài toán dựa vào những kiểu dữ liệu cơ bản được cài đặt sẵn trong ngôn ngữ lập trình.

I. TÓM TẮT LÝ THUYẾT

I.1. Khái niệm

Cấu trúc (struct) thực chất là một kiểu dữ liệu do người dùng định nghĩa bằng cách gom nhóm các kiểu dữ liệu cơ bản có sẵn trong C thành một kiểu dữ liệu phức hợp nhiều thành phần.

I.2. Định nghĩa kiểu dữ liệu

Cú pháp

```
struct < tên cấu trúc >
{
    Các kiểu dữ liệu thành phần ;
};
```

Ngoài ra ta có thể dùng từ khoá **typedef** để định nghĩa một tên mới cho kiểu dữ liệu đã có.

Cú pháp

```
typedef struct < tên cấu trúc > < tên mới >;
```

Ví dụ 1: Kiểu dữ liệu DATE gồm các thành phần:

- Thứ (thu): chuỗi có tối đa 4 ký tự.
- Ngày (ngay): số nguyên 1 byte.
- Tháng (thang): số nguyên 1 byte.
- Năm (nam): số nguyên 2 bytes.

Ta định nghĩa DATE như sau:

```
struct DATE
{
    char thu[5];
    unsigned char ngay;
    unsigned char thang;
    int nam;
};
typedef struct DATE d;
```

Kiểu dữ liệu có cấu trúc có thể lồng vào nhau.

Ví dụ 2: Định nghĩa kiểu dữ liệu của học sinh HOCSINH gồm:

- Mã số học sinh (MSHS): chuỗi có tối đa 5 ký tự.
- Họ tên (hoten): chuỗi có tối đa 30 ký tự.
- Ngày tháng năm sinh (ngaysinh): kiểu DATE.
- Địa chỉ (diachi): chuỗi có tối đa 50 ký tự.
- Giới tính (phai): chuỗi có tối đa 3 ký tự.
- Điểm trung bình (diemtb): số thực.

Ta định nghĩa kiểu HOCSINH như sau:

```
struct DATE
{
    char thu[5];
    unsigned char ngay;
    unsigned char thang;
    int nam;
};

typedef struct HOCSINH
{
    char MSHS[6];
    char hoten[31];
    struct DATE ngaysinh;
    char diachi[51];
    unsigned char phai[4];
    float diemtb;
};
```

⚠ Khi định nghĩa kiểu dữ liệu struct lồng nhau, ta cần lưu ý: Kiểu dữ liệu được sử dụng phải khai báo phía trên.

I.3. Khai báo

Khi ta định nghĩa kiểu dữ liệu tức là ta có một kiểu dữ liệu mới, muốn sử dụng ta phải khai báo biến. Cú pháp khai báo kiểu dữ liệu cũng giống như cách khai báo của các kiểu dữ liệu chuẩn.

```
struct < tên cấu trúc > < tên biến > ;
```

Ví dụ :

```
struct DATE x ; // Khai bao bien x co kieu du lieu DATE
```

Tuy nhiên nếu ta định nghĩa struct có dùng từ khoá **typedef** thì ta có thể khai báo trực tiếp mà không cần từ khoá “**struct**”.

Ví dụ :

DATE x ; // Khai báo biến x có kiểu DATE

***Biến con trỏ kiểu cấu trúc:** Ngoài cách khai báo như trên ta có thể khai báo theo kiểu con trỏ như sau

struct < tên cấu trúc > * < tên biến > ;

Để sử dụng ta cũng phải cấp phát vùng nhớ giống như kiểu dữ liệu chuẩn.

Ví dụ :

*DATE *y; // Khai báo con trỏ y kiểu cấu trúc DATE*

*y = (DATE *) malloc (sizeof (DATE)) ;*

I.4. Truy xuất

Để truy xuất một thành phần dữ liệu nào đó bên trong cấu trúc ta có 2 trường hợp truy xuất như sau :

- Biến x là một biến cấu trúc thông thường, ta dùng toán tử dấu chấm “.”

Cú pháp :

< Tên cấu trúc > . < Biến thành phần > ;

Ví dụ :

DATE x ; // khai báo biến x kiểu DATE

x.ngay = 5 ; // gan ngay bang 5

- Biến x là một biến con trỏ, ta dùng toán tử mũi tên “->” (Gồm dấu trừ ‘-’ và dấu lớn hơn ‘>’).

Cú pháp :

< Tên cấu trúc > -> < Biến thành phần > ;

Ví dụ :

*DATE *x ; // khai báo biến x kiểu con trỏ DATE*

x -> ngay = 5 ; // gan ngay bang 5

⊗ Đối với kiểu dữ liệu có struct lồng nhau phải truy cập đến thành phần cuối cùng có kiểu dữ liệu cơ bản.

Ví dụ: Giả sử, có kiểu HOCSINH như trên

HOCSINH hs; // khai báo biến hs kiểu HOCSINH

Muốn in học sinh A sinh vào tháng mấy ta phải truy cập như sau:

```
printf("Thang sinh của học sinh A là: %d", (hs.ngaysinh).thang);
```

I.5. Ví dụ minh họa

Viết chương trình nhập vào tọa độ hai điểm trong mặt phẳng và tính tổng hai tọa độ này.

```
#include <conio.h>
#include <stdio.h>

typedef struct DIEM //khai bao mot kieu du lieu DIEM gom toa do x va y
{
    int x;
    int y;
};

void Nhap (DIEM &d)
{
    printf ("\nNhap vao tao do diem\n");
    printf ("Tung do : ");
    scanf ("%d", & d.x);
    printf ("Hoanh do : ");
    scanf ("%d", & d.y);
}

void Xuat (DIEM d)
{
    printf ("\nToa do diem : (%d , %d)", d.x, d.y);
}

DIEM Tong (DIEM d1, DIEM d2)
{
    DIEM temp;
    temp.x = d1.x + d2.x;
    temp.y = d1.y + d2.y;
    return Temp;
}

void main ()
{
    DIEM A , B, AB; //khai bao 3 diem A, B, AB;
    clrscr ();
    Nhap ( A );
    Xuat ( A );
    Nhap ( B );
    Xuat ( B );
    printf ("\n Tong của hai diem vừa nhập là : ");
    AB = Tong ( A, B );
    Xuat ( AB );
}
```



```

    getch ();
}

```

I.6. Mảng cấu trúc

- Cách khai báo tương tự như mảng một chiều hay ma trận (Kiểu dữ liệu bây giờ là kiểu dữ liệu có cấu trúc).
- Cách truy cập phần tử trong mảng cũng như truy cập trên mảng một chiều hay ma trận. Nhưng do từng phần tử có kiểu cấu trúc nên phải chỉ định rõ cần lấy thành phần nào, tức là phải truy cập đến thành phần cuối cùng có kiểu là dữ liệu cơ bản (xem lại bảng các kiểu dữ liệu cơ bản) .

I.7. Nguyên tắc viết chương trình có mảng cấu trúc

Do kiểu dữ liệu có cấu trúc thường chứa rất nhiều thành phần nên khi viết chương trình loại này ta cần lưu ý:

- Xây dựng hàm xử lý cho một kiểu cấu trúc.
- Muốn xử lý cho mảng cấu trúc, ta gọi lại hàm xử lý cho một kiểu cấu trúc đã được xây dựng bằng cách dùng vòng lặp.

Ví dụ 1: Cho một lớp học gồm n học sinh ($n \leq 50$). Thông tin của một học sinh được mô tả ở ví dụ 2, mục I.2. Hãy viết chương trình nhập và xuất danh sách học sinh sau đó đếm xem có bao nhiêu học sinh được lên lớp (Điều kiện được lên lớp là điểm trung bình ≥ 5.0).

Cách làm:

- Trước hết ta phải xây dựng hàm nhập và xuất cho 1 học sinh.
- Xây dựng hàm nhập và xuất ngày tháng năm (Kiểu dữ liệu DATE).
- Sau đó mới xây dựng hàm nhập và xuất cho danh sách học sinh.

```

#define MAX 50
struct DATE
{
    char thu[5];
    unsigned char ngay;
    unsigned char thang;
    int nam;
};
typedef struct HOCSINH
{
    char MSHS[6];
    char hoten[31];
    struct DATE ngaysinh;
}

```

```

        char diachi[51];
        unsigned char phai[4];
        float diemtb;
    };

    void NhapNamSinh(DATE &d);
    void XuatNamSinh(DATE d);
    void Nhap1HS (HOCSINH &hs);
    void Xuat1HS (HOCSINH hs);
    void NhapDSHS(HOCSINH lh[], int &n);
    void XuatDSHS(HOCSINH lh[], int n);
    int DemHSLenLop(HOCSINH lh[], int n);

    void main()
    {
        HOCSINH lh[MAX]; //Khai báo mảng lh gồm có tối đa 50 học sinh
        int n, sohsdau;
        NhapDSHS(lh, n);
        XuatDSHS(lh, n);
        sohsdau = DemHSLenLop(lh, n);
        printf("\nSố lượng học sinh được lên lớp là: %d", sohsdau);
        getch();
    }

    void NhapNamSinh(DATE &d)
    {
        printf("\nNhập vào ngày: ");
        scanf("%u", &d.ngay);
        printf("\nNhập vào tháng: ");
        scanf("%u", &d.thang);
        printf("\nNhập vào năm: ");
        scanf("%d", &d.nam);
    }

    void XuatNamSinh(DATE d)
    {
        printf("%02u / %02u / %4d", d.ngay, d.thang, d.nam);
    }

    void Nhap1HS(HOCSINH &hs)
    {
        float d;

        lushall(); //Xóa vùng đệm
        printf("\nNhập mã số học sinh: ");
        gets(hs.MSHS);
        printf("\nNhập họ tên học sinh: ");
        gets(hs.hoten);
        printf("\nNhập ngày tháng năm sinh: ");
    }

```

```

    flushall(); //Xoa vung dem
    NhapNamSinh(hs.ngaysinh);
    printf("\nNhap vao dia chi: ");
    flushall(); //Xoa vung dem
    gets(hs.diachi);
    printf("\nPhai: ");
    gets(hs.phai);
    printf("\nNhap vao diem trung binh: ");
    flushall(); //Xoa vùng đệm
    scanf("%f", &d); //Nhập vào biến tạm d sau đó gán vào hs.diemtb
    hs.diemtb=d;
}

void NhapDSHS(HOCSINH lh[], int &n)
{
    printf("\nNhap vao so luong hoc sinh: ");
    scanf("%d", &n);
    for(int i=0; i<n; i++)
    {
        printf("\nNhap vao thong tin cua hoc sinh thu %d:\n", i+1);
        Nhap1HS(lh[i]); //Goi ham nhap thong tin 1 hoc sinh
    }
}

void Xuat1HS(HOCSINH hs)
{
    printf("\nMa so hoc sinh: %s", hs.MSHS);
    printf("\nHo ten hoc sinh: %s", hs.hoten);
    printf("\nNgay thang nam sinh: ");
    XuatNamSinh(hs.ngaysinh);
    printf("\nDia chi: %s", hs.diachi);
    printf("\nPhai: %s", hs.phai);
    printf("\nDiem trung binh: %.2f", hs.diemtb);
}

void XuatDSHS(HOCSINH lh[], int n)
{
    for(int i=0; i<n; i++)
    {
        printf("\n\nThong tin hoc sinh thu %d:", i+1);
        Xuat1HS(lh[i]); //Goi ham xuat thong tin 1 hoc sinh
    }
}

int DemHSLenLop(HOCSINH lh[], int n)
{
    int d=0;
    for(int i=0; i<n; i++)
        if(lh[i].diemtb>=5.0)

```

```
        d++;  
    return d;  
}
```

Kết quả ví dụ khi chạy chương trình:

Nhap vao thong tin cua hoc sinh thu 1:

Nhap ma so hoc sinh: 02313

Nhap ho ten hoc sinh: Nguyen Van A

Nhap ngay thang nam sinh:

Nhap vao ngay: 12

Nhap vao thang: 03

Nhap vao nam: 1980

Nhap vao dia chi: 60 Phan Dang Luu Q.Phu Nhuan

Phai: Nam

Nhap vao diem trung binh: 6.5

Nhap vao thong tin cua hoc sinh thu 2:

Nhap ma so hoc sinh: 03852

Nhap ho ten hoc sinh: Ly Thi B

Nhap ngay thang nam sinh:

Nhap vao ngay: 05

Nhap vao thang: 12

Nhap vao nam: 1981

Nhap vao dia chi: 24 Ly Tu Trong Q.1

Phai: Nu

Nhap vao diem trung binh: 3.5

Thong tin hoc sinh thu 1:

Ma so hoc sinh: 02313

Ho ten hoc sinh: Nguyen Van A

Ngay thang nam sinh: 12 / 03 / 1980

Dia chi: 60 Phan Dang Luu Q.Phu Nhuan

Phai: Nam

Diem trung binh: 6.50

Thong tin hoc sinh thu 2:

Ma so hoc sinh: 03852

Ho ten hoc sinh: Ly Thi B

Ngay thang nam sinh: 05 / 12 / 1981

Dia chi: 24 Ly Tu Trong Q.1

Phai: Nu

Diem trung binh: 3.50

So luong hoc sinh duoc len lop la: 1

Ví dụ 2: Cho một mảng các phân số (**PHANSO**) gồm n phần tử ($n \leq 50$). Hãy viết chương trình nhập và xuất danh sách các phân số sau đó tìm phân số có giá trị lớn nhất, tổng và tích các phân số và nghịch đảo giá trị các phân số trong mảng.

Cách làm:

- Trước hết ta phải xây dựng hàm nhập và xuất cho 1 phân số.
- Xây dựng hàm tính tổng, hiệu, tích, thương, rút gọn, so sánh và nghịch đảo cho 2 phân số.
- Sau đó mới xây dựng hàm nhập, xuất, tính tổng, tích cho mảng các phân số.

```
#define MAX 100
```

```
typedef struct PHANSO
```

```
{  
    int tu, mau;  
};
```

```
void NhapPS(PHANSO &ps);
```

```
void XuatPS(PHANSO ps);
```

```
void NhapMangPS(PHANSO dsps[], int &n);
```

```
void XuatMangPS(PHANSO dsps[], int n);
```

```
PHANSO TimMax(PHANSO dsps[], int n);
```

```
int KiemTra(PHANSO ps);
```

```
//Tra ve 1: Neu hop le
```

```
int USCLN(int a, int b);
```

```
PHANSO RutGon(PHANSO ps);
```

```
PHANSO NghichDao(PHANSO ps);
```

```
PHANSO Nhan(PHANSO ps1, PHANSO ps2);
```

```
PHANSO Chia(PHANSO ps1, PHANSO ps2);
```

```
PHANSO Tru(PHANSO ps1, PHANSO ps2);
```

```
PHANSO Cong(PHANSO ps1, PHANSO ps2);
```

```
int SoSanh(PHANSO ps1, PHANSO ps2);
```

```
//Tra ve 0: ps1=ps2
```

```
//Tra ve 1: ps1>ps2
```

```
//Tra ve -1: ps1<ps2
```

```
PHANSO TongCacPS(PHANSO dsps[], int n);
```

```
PHANSO TichCacPS(PHANSO dsps[], int n);
```

```
void NghichDaoCacPS(PHANSO dsps[], int n);
```

```
void main()
```

```
{
```

```
    int n;
```

```
    PHANSO a[MAX], max, s, p;
```

```
    clrscr();
```

```
    NhapMangPS(a, n);
```

```
    printf("\nMang cac phan so vua nhap: ");
```

```
    XuatMangPS(a, n);
```

```
    max=TimMax(a, n);
```

```
    printf("\nPhan so co gia tri lon nhat: ");
```

```
    XuatPS(max);
```

```
    s=TongCacPS(a, n);
```

```

    printf("\nTong gia tri cac phan so co trong mang: ");
    XuatPS(s);

    p=TichCacPS(a, n);
    printf("\nTich gia tri cac phan so co trong mang: ");
    XuatPS(p);

    NghichDaoCacPS(a, n);
    printf("\nMang phan so sau khi nghich dao cac phan tu: ");
    XuatMangPS(a, n);

    getch();
}

void NhapPS(PHANSO &ps)
{
    do{
        printf("\nNhap tu so: ");
        scanf("%d", &ps.tu);
        printf("\nNhap mau so: ");
        scanf("%d", &ps.mau);
        if(!KiemTra(ps))
            printf("\nMau so khong duoc bang 0, nhap lai phan so\n");
        else
            break;
    } while(1);
    ps=RutGon(ps);
}

void XuatPS(PHANSO ps)
{
    printf("%d", ps.tu);
    if(ps.tu&&ps.mau!=1)
        printf("/%d", ps.mau);
}

void NhapMangPS(PHANSO dsps[], int &n)
{
    printf("\nNhap so luong phan so: ");
    scanf("%d", &n);
    for(int i=0; i<n; i++)
    {
        printf("\nNhap vao phan so thu %d: ", i+1);
        NhapPS(dsps[i]);
    }
}

void XuatMangPS(PHANSO dsps[], int n)
{

```

```

    for(int i=0; i<n; i++)
    {
        XuatPS(dsps[i]);
        printf("\t");
    }
}

int KiemTra(PHANSO ps)
{
    if(ps.mau==0)
        return 0;
    return 1;
}

int USCLN(int a, int b)
{
    a=abs(a);
    b=abs(b);
    while(a!=b)
    {
        if(a>b)
            a=a-b;
        else
            b=b-a;
    }
    return a;
}

PHANSO RutGon(PHANSO ps)
{
    int us;
    if(ps.tu==0)
        return ps;

    us=USCLN(ps.tu, ps.mau);
    ps.tu=ps.tu/us;
    ps.mau=ps.mau/us;
    return ps;
}

PHANSO NghichDao(PHANSO ps)
{
    PHANSO kq;
    kq.tu=ps.mau;
    kq.mau=ps.tu;
    return kq;
}

PHANSO Nhan(PHANSO ps1, PHANSO ps2)

```

```

{
    PHANSO kq;
    kq.tu=ps1.tu*ps2.tu;
    kq.mau=ps1.mau*ps2.mau;
    kq=RutGon(kq);
    return kq;
}

PHANSO Chia(PHANSO ps1, PHANSO ps2)
{
    PHANSO kq;
    kq=Nhan(ps1, NghichDao(ps2));
    return kq;
}

PHANSO Tru(PHANSO ps1, PHANSO ps2)
{
    PHANSO kq;
    kq.tu=ps1.tu*ps2.mau-ps1.mau*ps2.tu;
    kq.mau=ps1.mau*ps2.mau;
    kq=RutGon(kq);
    return kq;
}

PHANSO Cong(PHANSO ps1, PHANSO ps2)
{
    PHANSO kq;
    kq.tu=ps1.tu*ps2.mau+ps1.mau*ps2.tu;
    kq.mau=ps1.mau*ps2.mau;
    kq=RutGon(kq);
    return kq;
}

int SoSanh(PHANSO ps1, PHANSO ps2)
{
    ps1=RutGon(ps1);
    ps2=RutGon(ps2);
    if(ps1.tu==ps2.tu&&ps1.mau==ps2.mau)
        return 0;
    if(ps1.tu*ps2.mau>ps2.tu*ps1.mau)
        return 1;
    return -1;
}

PHANSO TimMax(PHANSO dsps[], int n)
{
    PHANSO max;
    max=dsps[0];
    for(int i=1; i<n; i++)

```



```

        if(SoSanh(dsps[i], max)==1)
            max=dsps[i];
        return max;
    }

    PHANSO TongCacPS(PHANSO dsps[], int n)
    {
        PHANSO s=dsps[0];
        for(int i=1; i<n; i++)
        {
            s=Cong(s, dsps[i]);
        }
        return s;
    }

    PHANSO TichCacPS(PHANSO dsps[], int n)
    {
        PHANSO p=dsps[0];
        for(int i=1; i<n; i++)
        {
            p=Nhan(p, dsps[i]);
        }
        return p;
    }

    void NghichDaoCacPS(PHANSO dsps[], int n)
    {
        for(int i=0; i<n; i++)
        {
            dsps[i]=NghichDao(dsps[i]);
        }
    }

```

Kết quả ví dụ khi chạy chương trình:

Nhap so luong phan so: 5

Nhap vao phan so thu 1:

Nhap tu so: 1

Nhap mau so: 3

Nhap vao phan so thu 2:

Nhap tu so: 7

Nhap mau so: 4

Nhap vao phan so thu 3:

Nhap tu so: 9

Nhap mau so: 7

Nhap vao phan so thu 4:

Nhap tu so: 5

Nhap mau so: 6

Nhap vao phan so thu 5:

Nhap tu so: 4

Nhap mau so: 7

Mang cac phan so vua nhap: 1/3 7/4 9/7 5/6 4/7

Phan so co gia tri lon nhat: 7/4

Tong gia tri cac phan so co trong mang: 401/84

Tich gia tri cac phan so co trong mang: 5/14

Mang phan so sau khi nghich dao cac phan tu: 3 4/7 7/9 6/5 7/4

II. BÀI TẬP

II.1. Bài tập cơ bản

- Viết chương trình sử dụng con trỏ cấu trúc để hiển thị giờ, phút, giây ra màn hình, và tính khoảng cách giữa 2 mốc thời gian.
- Viết chương trình sử dụng con trỏ cấu trúc thể hiện ngày, tháng, năm ra màn hình, và tính khoảng cách giữa 2 ngày.
- Viết chương trình khai báo kiểu dữ liệu thể hiện một số phức. Sử dụng kiểu này để viết hàm tính tổng, hiệu, tích của hai số phức.
- Viết chương trình khai báo kiểu dữ liệu để biểu diễn một phân số. Hãy viết hàm thực hiện những công việc sau:
 - Tính tổng, hiệu, tích, thương hai phân số.
 - Rút gọn phân số.
 - Qui đồng hai phân số.
 - So sánh hai phân số.
- Viết chương trình khai báo kiểu dữ liệu để biểu diễn một hỗn số. Hãy viết hàm thực hiện những công việc sau :
 - Đổi hỗn số sang phân số
 - Tính tổng, tích hai hỗn số
- Viết chương trình khai báo kiểu dữ liệu để biểu diễn một điểm trong hệ tọa độ Oxy . Hãy viết hàm thực hiện các công việc sau:
 - Tìm những điểm đối xứng của nó qua tung độ, hoành độ, tọa độ tâm.
 - Hãy tính tổng, hiệu, tích của hai điểm trong mặt phẳng tọa độ Oxy.
 - Tính khoảng cách giữa hai điểm.

7. Cho một hình trụ có các thông tin sau: BanKinh (bán kính hình trụ kiểu số thực), ChieuCao (chiều cao hình trụ kiểu số thực). Hãy thực hiện các công việc sau.
- Nhập dữ liệu cho hình trụ trên.
 - Tính diện tích xung quanh, diện tích toàn phần, thể tích hình trụ.

II.2. Bài Tập Luyện Tập

8. Viết chương trình tạo một mảng các số phức. Hãy viết hàm tính tổng, tích các số phức có trong mảng.
9. Viết chương trình tạo một mảng các phân số. Hãy viết hàm thực hiện các công việc sau :
- Tính tổng tất cả các phân số (kết quả dưới dạng phân số tối giản)
 - Tìm phân số lớn nhất, phân số nhỏ nhất.
 - Sắp xếp mảng tăng dần.
10. Viết chương trình khai báo kiểu dữ liệu STACK (cơ chế LIFO). Viết hàm làm những công việc sau :
- Kiểm tra STACK rỗng
 - Kiểm tra STACK đầy
 - Thêm phần tử vào STACK
 - Lấy phần tử ra khỏi STACK
11. Tổ chức dữ liệu để quản lí sinh viên bằng cấu trúc mẫu tin trong một mảng N phần tử, mỗi phần tử có cấu trúc như sau:
- Mã sinh viên.
 - Tên.
 - Năm sinh.
 - Điểm toán, lý, hoá, điểm trung bình.

Viết chương trình thực hiện những công việc sau:

- Nhập danh sách các sinh viên cho một lớp học.
- Xuất danh sách sinh viên ra màn hình.
- Tìm sinh viên có điểm trung bình cao nhất.
- Sắp xếp danh sách lớp theo thứ tự tăng dần của điểm trung bình.
- Sắp xếp danh sách lớp theo thứ tự giảm dần của điểm toán.

- Tìm kiếm và in ra các sinh viên có điểm trung bình lớn hơn 5 và không có môn nào dưới 3.
- Tìm sinh viên có tuổi lớn nhất.
- Nhập vào tên của một sinh viên. Tìm và in ra các thông tin liên quan đến sinh viên đó (nếu có).

12. Tổ chức dữ liệu quản lý danh mục các bộ phim VIDEO, các thông tin liên quan đến bộ phim này như sau:

- Tên phim (tựa phim).
- Thể loại (3 loại : hình sự, tình cảm, hài).
- Tên đạo diễn.
- Tên diễn viên nam chính.
- Tên diễn viên nữ chính.
- Năm sản xuất.
- Hãng sản xuất

Viết chương trình thực hiện những công việc sau :

- Nhập vào bộ phim mới cùng với các thông tin liên quan đến bộ phim này.
- Nhập một thể loại: In ra danh sách các bộ phim thuộc thể loại này.
- Nhập một tên nam diễn viên. In ra các bộ phim có diễn viên này đóng.
- Nhập tên đạo diễn. In ra danh sách các bộ phim do đạo diễn này dàn dựng.

13. Một thư viện cần quản lý thông tin về các đầu sách. Mỗi đầu sách bao gồm các thông tin sau : MaSSach (mã số sách), TenSach (tên sách), TacGia (tác giả), SL (số lượng các cuốn sách của đầu sách). Viết chương trình thực hiện các chức năng sau:

- Nhập vào một danh sách các đầu sách (tối đa là 100 đầu sách)
- Nhập vào tên của quyển sách. In ra thông tin đầy đủ về các sách có tên đó, nếu không có thì tên của quyển sách đó thì báo là :Không Tìm Thấy.
- Tính tổng số sách có trong thư viện.

14. Viết chương trình tạo một mảng danh sách các máy tính của một cửa hàng, thông tin của một máy tính bao gồm :

- Loại máy
- Nơi sản xuất

- *Thời gian bảo hành*

- Viết hàm nhập một dãy các loại máy tính có thông tin như trên.
- Hãy viết hàm thống kê xem có bao nhiêu máy có thời gian bảo hành là 1 năm.
- In ra danh sách các máy tính có xuất xứ từ Mỹ.

15. Để lắp ráp một máy vi tính hoàn chỉnh cần phải có tối thiểu 10 linh kiện loại A và có thể lắp bổ sung thêm vào khoảng tối đa 8 linh kiện loại B. Tại một cửa hàng vi tính cần quản lý bán hàng các loại linh kiện tại cửa hàng. Thông tin về một loại linh kiện gồm có: Tên linh kiện, quy cách, loại, đơn giá loại 1 (chất lượng tốt – số nguyên), đơn giá loại 2 (chất lượng thường – số nguyên). Viết chương trình thực hiện những công việc sau :

- Nhập vào thông tin về các linh kiện có ở cửa hàng.
- Xuất danh sách các linh kiện đã nhập theo thứ tự tăng dần của loại linh kiện và tên linh kiện.
- Cho biết đã có đủ 10 linh kiện loại A cần thiết lắp ráp máy hay chưa?

16. Một cửa hàng cần quản lý các mặt hàng, thông tin một mặt hàng bao gồm:

- *Mã hàng.*
- *Tên mặt hàng.*
- *Số lượng.*
- *Đơn giá.*
- *Số lượng tồn.*
- *Thời gian bảo hành (tính theo đơn vị tháng).*

- Hãy nhập vào một danh sách các mặt hàng.
- Tìm mặt hàng có số lượng tồn nhiều nhất.
- Tìm mặt hàng có số lượng tồn ít nhất.
- Tìm mặt hàng có giá tiền cao nhất.
- In ra những mặt hàng có thời gian bảo hành lớn hơn 12 tháng.
- Sắp xếp các mặt hàng theo thứ tự tăng dần của số lượng tồn.

17. Viết chương trình quản lý hồ sơ nhân viên trong một công ty, chương trình thực hiện những công việc sau :

- *Họ và tên.*
- *Phái.*

- Ngày sinh.
 - Địa chỉ.
 - Lương cơ bản.
 - Bảo hiểm xã hội.
 - Thưởng.
 - Phạt.
 - Lương thực lĩnh = lương cơ bản + thưởng – BH xã hội – phạt.
- Nhập vào hồ sơ của các nhân viên trong công ty.
 - Xuất danh sách các nhân viên theo lương thực lĩnh giảm dần bằng 2 cách sau :
 - Cấp phát vùng nhớ tĩnh.
 - Cấp phát vùng nhớ động.
18. (*) Viết chương trình quản lý lớp học của một trường. Các thông tin của một lớp học như sau :
- Tên lớp.
 - Sĩ số.
 - Danh sách các sinh viên trong lớp.
- Nhập vào danh sách các lớp với thông tin yêu cầu như trên.
 - In danh sách các lớp có trên 5 sinh viên có điểm trung bình loại giỏi.
 - Tìm lớp có nhiều sinh viên nhất.
 - Tìm lớp có ít sinh viên nhất.
 - Tìm sinh viên có điểm trung bình cao nhất.
 - Tìm lớp có số lượng sinh viên đạt điểm trung bình loại giỏi nhiều nhất.
19. Viết chương trình quản lý vé tàu, thông tin một vé tàu như sau :
- Ngày giờ khởi hành, ngày giờ đến.
 - Ga đi, ga đến.
 - Loại tàu, loại chỗ ngồi (ngồi, nằm, cứng, mềm).
 - Số toa, số ghế.
- Viết hàm nhập vào danh sách các vé tàu.
 - In danh sách các vé tàu có ga đến là Huế.
 - In danh sách các vé tàu có ga đến là Hà Nội và đi ngày 8/6/2005.
 - Đếm xem có bao nhiêu khách đi tàu loại chỗ ngồi là nằm cứng.

20. Viết chương trình tính tiền điện hàng tháng của các hộ gia đình, thông tin các khách hàng như sau :
- *Kỳ thu, từ ngày.....đến ngày.*
 - *Tên khách hàng, mã khách hàng.*
 - *Địa chỉ.*
 - *Điện năng tiêu thụ (Kwh).*
 - Nhập vào danh sách các khách hàng.
 - Xuất danh sách hoá đơn theo thứ tự tăng dần của điện năng tiêu thụ.
 - Tính tiền điện của các khách hàng theo quy định sau.
 - *100 kw đầu tiên là 550 đ / kw*
 - *50 kw tiếp theo là 900 đ / kw*
 - *50 kw tiếp theo là 1210 đ / kw*
 - *Thuế 10 % trên tổng số tiền phải trả*
 - Tính tổng số tiền thu được của các khách hàng.

III. KẾT LUẬN

- ❖ Kiểu dữ liệu có cấu trúc cho phép ta định nghĩa những kiểu dữ liệu bất kỳ trên cơ sở là những kiểu dữ liệu cơ bản có sẵn trong ngôn ngữ lập trình.
- ❖ Khi xây dựng xong kiểu dữ liệu mới ta phải **định nghĩa những thao tác** cho kiểu dữ liệu đó.
- ❖ Những kiểu dữ liệu tự định nghĩa này thông thường có rất nhiều thành phần, mỗi thành phần cũng có thể là một kiểu dữ liệu tự định nghĩa, vấn đề là ta chọn kiểu dữ liệu cơ bản nào để xây dựng nên chúng sao cho **phù hợp về mặt kiểu dữ liệu và phù hợp về kích thước lưu trữ** (vừa đủ).
- ❖ Các sử dụng những kiểu dữ liệu tự định nghĩa cũng giống như các kiểu dữ liệu cơ bản. Muốn sử dụng phải khai báo biến, khi truy cập các thành phần phải truy cập theo quy ước.
- ❖ Nếu thành phần cấu trúc có kiểu dữ liệu là số thực thì khi sử dụng hàm **scanf()** phải thông qua biến trung gian rồi gán lại cho thành phần cấu trúc đó.
- ❖ Đối với mảng các kiểu dữ liệu có cấu trúc ta nên **xử lý cho từng thành phần cấu trúc** rồi mới xử lý cho mảng cấu trúc bằng cách dùng vòng lặp.

Trong chương này, chúng ta sẽ tìm hiểu cấu trúc tập tin, cài đặt các thao tác, một số hàm thư viện và ứng dụng trong việc tổ chức dữ liệu trên tập tin.

I. TÓM TẮT LÝ THUYẾT

I.1. Khái niệm

Trong các chương trình trước thì các dữ liệu đưa vào chương trình chỉ được tồn tại trong RAM, khi thoát chương trình thì tất cả dữ liệu đều bị mất. Để khắc phục tình trạng này Borland C cung cấp cho ta các hàm để lưu trữ và truy xuất tập tin, đó là kiểu **FILE**. Và ở đây ta chỉ đề cập đến 2 loại tập tin :

- Tập tin văn bản: là tập tin dùng để ghi các ký tự lên đĩa theo các dòng.
- Tập tin nhị phân: là tập tin dùng để ghi các cấu trúc dạng nhị phân (được mã hoá).

I.2. Thao tác với tập tin

Quá trình thao tác trên tập tin thông qua 4 bước:

Bước 1: Khai báo con trỏ trỏ đến tập tin.

Bước 2: Mở tập tin.

Bước 3: Các xử lý trên tập tin.

Bước 4: Đóng tập tin.

a. Khai báo

FILE * < tên biến >;

Ví dụ : `FILE *f; // Khai bao bien con tro file f`

b. Mở tập tin

fopen (< đường dẫn tên tập tin> , < kiểu truy nhập >);

Ví dụ : `FILE *f; // Khai bao bien con tro f`
`f = fopen ("C:\\VD1.txt" , "rt");`

Các kiểu truy nhập tập tin thông dụng:

- t** là kiểu truy nhập tập tin đối với dạng tập tin văn bản (text).
- b** là kiểu truy nhập tập tin đối với dạng tập tin nhị phân (binary).
- r** mở ra để đọc (ready only).
- w** mở ra để ghi (create / write).
- a** mở ra để thêm vào (append).
- r+** mở ra để đọc và ghi (modify).

c. Các hàm đọc ghi nội dung tập tin

- Tập tin văn bản

STT	TÊN HÀM	Ý NGHĨA SỬ DỤNG	VÍ DỤ
ĐỌC TẬP TIN			
1	fscanf(<FILE *>, <định dạng>, <các tham biến>);	Đưa dữ liệu từ một tập tin theo định dạng.	<i>fscanf(f, "%d", &x);</i>
2	fgets(<vùng nhớ>, <kích thước tối đa>, <FILE *>);	Đọc một chuỗi ký tự từ một tập tin với kích thước tối đa cho phép, hoặc gặp ký tự xuống dòng.	<i>char s[80]; fgets(s, 80, f);</i>
3	getc(<FILE *>);	Đọc một ký tự từ tập tin đang mở.	<i>char c=getc(f);</i>
GHI TẬP TIN			
1	fprintf(<FILE *>, <định dạng>[, <các tham biến>]);	Ghi dữ liệu theo một định dạng nào đó vào tập tin.	<i>fprintf(f, "%d", x);</i>
2	fputs(<chuỗi ký tự>, <FILE *>);	Ghi một chuỗi ký tự vào tập tin đang mở.	<i>fputs("Giao trình BT", f);</i>

- Tập tin nhị phân

STT	TÊN HÀM	Ý NGHĨA SỬ DỤNG	VÍ DỤ
ĐỌC TẬP TIN			
1	fread(&ptr, <size>, <len>, <FILE *>);	<ul style="list-style-type: none"> • ptr: vùng nhớ để lưu dữ liệu đọc. • size: kích thước mỗi ô nhớ (tính bằng byte). • len: độ dài dữ liệu cần đọc. FILE: đọc từ tập tin nhị phân nào.	<i>int a[30], b, n; fread(a, sizeof(int), n, f); Fread(&b, sizeof(int), 1, f);</i>
GHI TẬP TIN			
1	fwrite(&prt, <size>, <len>, <FILE *>);	Tham số tương tự như hàm fread.	<i>fwrite(a, sizeof(int), n, f);</i>

d. Đóng tập tin

Sau khi không còn làm việc với tập tin, để đảm bảo an toàn cho dữ liệu thì nhất thiết ta phải đóng tập tin lại.

```
fclose ( < biến con trỏ tập tin > );
hoặc fcloseall ( );
```

Ví dụ : fclose (f);

e. Các thao tác khác trên tập tin*** Xoá tập tin :**

```
remove ( < đường dẫn tập tin > );
```

*** Đổi tên tập tin :**

```
rename ( < tên tập tin cũ > , < tên tập tin mới > );
```

*** Di chuyển con trỏ tập tin :**

```
fseek ( < FILE * > , < độ dời > , < mốc > );
```

Các mốc :

SEEK_SET dời đến đầu tập tin (giá trị 0).

SEEK_END dời đến cuối tập tin (giá trị 2).

SEEK_CUR dời vị trí hiện hành (giá trị 1).

Ví dụ :

```
fseek ( f , +5 , SEEK_CUR ); // dời vị trí hiện hành về cuối 5 bytes
```

```
fseek ( f , -4 , SEEK_CUR ); // dời vị trí hiện hành về trước 4 bytes
```

*** Cho biết vị trí con trỏ file:**

```
ftell ( < FILE * > );
```

Ví dụ :

```
int size = ftell ( f );
```

```
/*size: khoảng cách từ đầu tập tin đến vị trí hiện hành (tính bằng
byte)*/
```

f. Ví dụ minh hoạ

```
void KiemTra (FILE *f, char duongdan [])
{
    f = open ( duongdan , "rt");
    if ( f == NULL )
    {
        printf ("Khong mo duoc tap tin %s", duongdan);
        perror ("\nLy do");
    }
}
```

```

        getch ();
        return ;
    }
    printf ("Tập tin %s da duoc mo", duongdan);
    fclose (f);
}

```

I.3. Các ví dụ minh hoạ

a. Tập tin văn bản

Ví dụ 1: Viết chương trình tạo tập tin văn bản SO.OUT gồm n số nguyên, các số của dãy được tạo ngẫu nhiên có giá trị tuyệt đối không vượt quá M (n, M đọc từ tập tin SO.INP). Kết quả chương trình là 1 tập tin văn bản có dòng thứ nhất ghi số n ; n dòng tiếp theo ghi các số tạo được, mỗi số trên một dòng.

SO.INP
3
10

SO.OUT
3
5
7
2

```

#include < conio.h >
#include < stdio.h >

#define in "SO.INP"
#define out "SO.OUT"
int n, M ;

void Nhap ()
{
    FILE *fi;
    fi = fopen ( in , "rt" );
    fscanf ( fi, "%d %d ", &n, &M );
    fclose ( fi );
}

void Xuat ()
{
    FILE *fo;
    fo = fopen ( out , "wt " );
    fprintf ( fo , "%d\n", n );
    randomize ();
    for ( ; n > 0 ; n -- )
        fprintf ( fo , "%d\n", random ( ( 2 * M + 1 ) - M ) );
    fclose ( fo );
}

```

```

}
void main ()
{
    clrscr ();
    Nhap ();
    Xuat ();
}

```

Ví dụ 2: Viết chương trình phát sinh ngẫu nhiên ma trận a kích thước 5×6 , lưu ma trận này vào file **test.inp**. Đọc lại file **test.inp** đưa dữ liệu vào ma trận b và xuất ra màn hình xem kết quả lưu đúng không? Cấu trúc của file **test.inp** như sau:

- Dòng đầu lưu 2 số nguyên: m, n thể hiện số dòng và số cột của ma trận.
- m dòng tiếp theo, mỗi dòng gồm n phần tử là giá trị các phần tử trên một dòng của ma trận.

```

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

#define MAX 100
#define dl "test.inp"

void LuuFile(int a[MAX][MAX], int m, int n)
{
    FILE *f;
    f=fopen(dl, "wt");
    if(f==NULL)
    {
        printf("\nKhong tao duoc file.");
        getch();
        exit(0);
    }
    fprintf(f, "%d %d\n", m, n);
    for(int i=0; i<m; i++)
    {
        for(int j=0; j<n; j++)
            fprintf(f, "%d\t", a[i][j]);
        fprintf(f, "\n");
    }
    fclose(f);
}

void DocFile(int a[MAX][MAX], int &m, int &n)
{

```

```

FILE *f;
f=fopen(dl, "rt");
if(f==NULL)
{
    printf("\nKhong doc duoc file.");
    getch();
    exit(0);
}
fscanf(f, "%d%d", &m, &n);
for(int i=0; i<m; i++)
{
    for(int j=0; j<n; j++)
        fscanf(f, "%d", &a[i][j]);
}
fclose(f);
}

void main()
{
    int a[MAX][MAX], m=5, n=6, i, j;
    int b[MAX][MAX], x, y;

    randomize();
    for(i=0; i<m; i++)
        for(j=0; j<n; j++)
            a[i][j]=random(1000);
    LuuFile(a, m, n);
    DocFile(b, x, y);
    for(i=0; i<x; i++)
    {
        for(j=0; j<y; j++)
            printf("%d\t", b[i][j]);
        printf("\n");
    }
}

```

Kết quả ví dụ sau khi chạy chương trình, file test.inp có dạng sau:

5	6				
480	661	395	736	998	987
31	414	211	801	774	416
166	191	454	830	508	72
121	382	35	365	567	726
159	309	1	275	870	378

b. Tập tin nhị phân

Viết hàm đọc/ ghi một danh sách sinh viên của một lớp vào tập tin SV.DAT

```
SINHVIEN ds[100];
```

```

int siso;

void nhap ( )
{
    FILE *fi;
    fi = fopen ( "SV.DAT", "rb" );
    fseek ( fi, 0, SEEK_END );
    siso = ( ftell ( fi ) + 1 ) / sizeof ( SINHVIEN );
    fseek ( fi, 0, SEEK_SET );
    fread ( ds, sizeof ( SINHVIEN ), siso, fi );
    fclose ( fi );
}

void xuat ( )
{
    FILE *fo;
    fo = fopen ( "SV.DAT", "wb" );
    fwrite ( ds, sizeof ( SINHVIEN ), siso, fo );
    fclose ( fo );
}

```

II. BÀI TẬP

II.1. Bài tập cơ bản

- Viết chương trình tạo tập tin văn bản chứa 1 dãy số nguyên bất kỳ.
- Viết chương trình tạo tập tin nhị phân chứa 10000 số nguyên bất kỳ ghi vào file SONGUYEN.INP. Mỗi dòng 10 số, sau đó viết chương trình đọc file SONGUYEN.INP, sắp xếp theo thứ tự tăng dần và lưu kết quả vào file SONGUYEN.OUT.
- Viết chương trình tạo một file chứa 10000 số nguyên ngẫu nhiên đôi một khác nhau trong phạm vi từ 1 đến 32767 và đặt tên là "SONGUYEN.INP".
- Viết chương trình tạo một file chứa các số nguyên có tên SONGUYEN.INP. Sau đó đọc file SONGUYEN.INP và ghi các số chẵn vào file SOCHAN.OUT và những số lẻ vào file SOLE.OUT.
- Viết chương trình ghi vào tập tin SOCHAN.DAT các số nguyên chẵn từ 0 đến 100.
- Viết chương trình đọc tập tin SOCHAN.DAT và xuất ra màn hình, mỗi dòng 30 số.
- Viết chương trình giả lập lệnh **COPY CON** để tạo tập tin văn bản. Khi kết thúc tập tin nhấn phím **F6** để lưu.

8. Viết chương trình giả lập lệnh **TYPE** để in nội dung của tập tin văn bản ra màn hình.
9. Viết chương trình kiểm tra một tập tin nào đó có trong một thư mục được chỉ định hay không?
10. Viết chương trình giả lập lệnh **DEL** để xoá tập tin. Yêu cầu nhập đường dẫn và tên tập tin, kiểm tra sự tồn tại của tập tin, nếu có thì xoá tập tin được chỉ định.
11. Viết chương trình giả lập lệnh **RENAME** để đổi tên một tập tin.
12. Viết chương trình tạo file văn bản có tên là “MATRIX.INP” có cấu trúc như sau:
 - Dòng đầu ghi hai số m, n .
 - Trong m dòng tiếp theo mỗi dòng ghi n số và các số các nhau một khoảng cách.Hãy kiểm tra xem trong file đó có bao nhiêu số nguyên tố.
Kết quả cần ghi vào file “MATRIX.OUT” có nội dung là một số nguyên đó là số lượng các số nguyên tố trong file “MATRIX.INP”.
13. Cho số nguyên n , hãy in tam giác PASCAL gồm n dòng
Dữ liệu vào: tập tin văn bản PAS.INP gồm 1 dòng chứa giá trị n .
Kết quả: đưa ra tập tin văn bản PAS.OUT thể hiện một tam giác PASCAL n dòng.
14. Cho mảng các số nguyên, hãy sắp xếp mảng theo thứ tự tăng dần.
Dữ liệu vào : tập tin văn bản ARRAY.INP gồm 2 dòng
 - Dòng 1 chứa số nguyên n ($n \leq 100$).
 - Dòng 2 chứa n số nguyên.Kết quả : Đưa ra tập tin văn bản ARRAY.OUT gồm hai dòng
 - Dòng 1 chứa n phần tử của mảng các số nguyên.
 - Dòng 2 chứa n số nguyên được xếp tăng dần.
15. Cho mảng các số nguyên, tìm phần tử lớn nhất của mảng.
Dữ liệu vào: tập tin văn bản ARRAY.INP gồm hai dòng:
 - Dòng 1 chứa số nguyên n ($n \leq 100$).
 - Dòng 2 chứa n số nguyên.Kết quả: Đưa ra tập tin văn bản ARRAY.OUT gồm 1 dòng ghi 2 giá trị x, y trong đó x là giá trị lớn nhất, y là vị trí của x trong mảng.

II.2. Bài tập luyện tập và nâng cao

16. Cho mảng các số nguyên, tính tổng các phần tử của mảng.
 Dữ liệu vào : tập tin văn bản ARRAY.INP gồm hai dòng
 - Dòng 1 chứa số nguyên n ($n \leq 10$)
 - Dòng 2 chứa n số nguyên
 Kết quả : Đưa ra tập tin văn bản ARRAY.OUT gồm một dòng ghi tổng các phần tử trong mảng.
17. Cho mảng các số nguyên, hãy liệt kê các phần tử là số nguyên tố
 Dữ liệu vào : tập tin văn bản NT.INP gồm hai dòng
 - Dòng 1 chứa số nguyên n ($n \leq 100$)
 - Dòng 2 chứa n số nguyên
 Kết quả : đưa ra tập tin văn bản NT.OUT gồm hai dòng:
 - Dòng 1 chứa số lượng các phần tử nguyên tố trong mảng.
 - Dòng 2 liệt kê các số nguyên tố đó.
18. (*) Tạo file văn bản có tên là "INPUT.TXT" có cấu trúc như sau:
 - Dòng đầu tiên ghi N (N là số nguyên dương nhập từ bàn phím).
 - Trong các dòng tiếp theo ghi N số nguyên ngẫu nhiên trong phạm vi từ 0 đến 100, mỗi dòng 10 số (các số cách nhau ít nhất một khoảng trắng).

Hãy đọc dữ liệu của file "INPUT.TXT" và lưu vào mảng một chiều A.

Thực hiện các công việc sau :

- Tìm giá trị lớn nhất của mảng A.
- Đếm số lượng số chẵn, số lượng số lẻ của mảng A.
- Hãy sắp xếp các phần tử theo thứ tự tăng dần.

Hãy ghi các kết quả vào file văn bản có tên OUTPUT.TXT theo mẫu sau:

INPUT.TXT	OUTPUT.TXT
18	Cau a: 99
87 39 78 19 89 4 40 98 29	Cau b: 9 9
65	Cau c:
20 43 1 99 38 34 58 4	1 4 4 19 20 29 34 38 39 40
	43 58 65 78 87 89 98 99

19. (*) Viết chương trình nhập và lưu hồ sơ của sinh viên vào một file có tên là "DSSV.TXT". Sau đó đọc file "DSSV.TXT" và cất vào mảng, hãy sắp xếp các hồ sơ sinh viên theo thứ tự giảm dần theo điểm trung bình môn

học rồi in ra màn hình hồ sơ các sinh viên theo thứ tự đó ra màn hình có thông tin như sau :

- Mã số sinh viên.
- Họ và tên sinh viên.
- Điểm trung bình kiểm tra.
- Điểm thi hết môn.
- Điểm trung bình môn học (tính bằng (điểm TBKT+điểm thi)/2).

20. (*) Tạo một file text có tên là “INPUT.TXT” có cấu trúc như sau :

- Dòng đầu tiên ghi hai số M và N (M, N là hai số nguyên dương nhập từ bàn phím).
- Trong M dòng tiếp theo mỗi dòng ghi N số nguyên ngẫu nhiên trong phạm vi từ 0 đến 100 (các số này cách nhau ít nhất một khoảng trắng).

Hãy đọc dữ liệu từ file trên và lưu vào mảng hai chiều. Rồi thực hiện các công việc sau:

- Tìm giá trị lớn nhất của ma trận.
- Đếm số lượng số chẵn, lẻ, nguyên tố có trong ma trận.
- Hãy tính tổng các phần tử trên mỗi dòng của ma trận.

Hãy ghi kết quả này vào filetext có tên là “OUTPUT.TXT”

INPUT.TXT	OUTPUT.TXT
6 6	Cau a: 49
41 17 33 23 12 1	Cau b: 17 19
44 24 23 49 5 24	Cau c: 127 169 147 214 132
33 20 17 25 33 19	146
0 48 45 48 41 32	
10 24 36 19 19 24	
30 4 23 26 27 36	

21. (**) Xét dãy số a_1, a_2, \dots, a_N . Một đoạn con của dãy là dãy các phần tử liên tiếp nhau được xác định bởi chỉ số của số bắt đầu (L) và chỉ số của số cuối cùng (R). Tổng các số trên đoạn được gọi là tổng đoạn.

Yêu cầu : Cho dãy (a_N), hãy tìm đoạn con có tổng đoạn lớn nhất (T)

Dữ liệu được cho trong tập tin văn bản SUMMAX.INP

- Dòng thứ nhất chứa số nguyên N ($0 < N \leq 30000$)

- *N dòng tiếp theo, mỗi dòng chứa một số là các số của dãy đã cho theo đúng thứ tự. Giá trị tuyệt đối của mỗi số không vượt quá 30000*

Kết quả tìm được ghi vào tập tin văn bản SUMMAX.OUT gồm 1 dòng ghi 3 số T, L, R.

Ví dụ :

SUMMAX.INP	SUMMAX.OUT
5	8 2 5
-1	
5	
-3	
2	
4	

22. (*) Cho dãy (a_N), hãy tìm đoạn con tăng dần có tổng lớn nhất

Dữ liệu : được cho trong tập tin AMAX.INP

- *Dòng 1 chứa số nguyên N ($0 < N \leq 30000$).*
- *N dòng tiếp theo, mỗi dòng chứa một số là các số của dãy đã cho theo đúng thứ tự. Giá trị tuyệt đối của mỗi số không vượt quá 30000.*

Kết quả tìm được ghi vào tập tin văn bản AMAX.OUT gồm hai dòng:

- *Dòng 1 ghi tổng của dãy con.*
- *Dòng 2 ghi mảng con tăng dần có tổng lớn nhất.*

23. Viết chương trình nhập lý lịch một nhân viên vào danh sách các nhân viên.

Khi không nhập nữa bấm phím **Esc** và ghi vào tập tin NHANVIEN.DAT sau đó :

- Đọc từ tập tin NHANVIEN.DAT vừa tạo và in danh sách các nhân viên lên màn hình.
- Tìm và in lý lịch một nhân viên bằng các nhập và họ tên hoặc mã số nhân viên.

24. (***) Để lắp ráp một máy vi tính hoàn chỉnh cần phải có tối thiểu 10 linh kiện loại A và có thể lắp bổ sung thêm vào khoảng tối đa 8 linh kiện loại B. Tại một cửa hàng vi tính cần quản lý bán hàng các loại linh kiện tại cửa hàng. Thông tin về một loại linh kiện gồm có: Tên linh kiện, quy cách , loại, đơn giá loại 1 (chất lượng tốt – số nguyên), đơn giá loại 2 (chất lượng thường – số nguyên). Viết chương trình thực hiện những công việc sau :

- Nhập vào thông tin của các loại linh kiện có ở cửa hàng. Xuất danh sách các linh kiện đã nhập theo thứ tự tăng dần của loại linh kiện và tên linh kiện. Cho biết đã có đủ 10 linh kiện loại A cần thiết để lắp ráp máy tính hay chưa?
- Với giả định là cửa hàng đã có đủ 10 linh kiện loại A để lắp ráp máy. Nhập vào một số tiền để lắp ráp một máy tính. Có thể lắp được một máy tính hoàn chỉnh với các linh kiện toàn bộ theo đơn giá loại 1 hay đơn giá loại 2 hay không? Nếu số tiền trong khoảng giữa thì hãy tìm một phương án gồm những linh kiện theo đơn giá 1 và linh kiện theo đơn giá 2 để lắp?
- Tất cả dữ liệu phải lưu ở tập tin.

III. KẾT LUẬN

- ❖ Mục đích của kiểu dữ liệu tập tin cho phép chúng ta lưu lại những thông tin cần thiết tương đối lớn: những dữ liệu đầu vào, những kết quả của chương trình hoặc những dữ liệu dùng để test chương trình, ...
- ❖ Khi thao tác trên tập tin phải thông qua 4 bước: ***Khai báo con trỏ trỏ đến tập tin, Mở tập tin, Xử lý trên tập tin và cuối cùng là Đóng tập tin.***
- ❖ Lưu ý khi mở tập tin để ghi thì phải cẩn thận với thao tác tạo mới hay chỉnh sửa nội dung tập tin, di chuyển con trỏ hợp lý để tránh mất thông tin.
- ❖ Sử dụng hàm thao tác trên tập tin phải dùng **đúng loại hàm** cho tập tin kiểu nhị phân hay kiểu văn bản.

Giới thiệu phương pháp lập trình theo kỹ thuật đệ quy, phân loại, cách hoạt động và cách cài đặt các hàm đệ quy.

I. TÓM TẮT LÝ THUYẾT

I.1. Khái niệm

Một hàm được gọi có tính đệ qui nếu trong thân của hàm đó có lệnh gọi lại chính nó một cách tường minh hay tiềm ẩn.

I.2. Phân loại đệ qui

- Đệ qui tuyến tính.
- Đệ qui nhị phân.
- Đệ qui phi tuyến.
- Đệ qui hỗ tương.

a. **Đệ qui tuyến tính**

Trong thân hàm có duy nhất một lời gọi hàm gọi lại chính nó một cách tường minh.

```
<Kiểu dữ liệu hàm> TenHam (<danh sách tham số>)  
{  
    if (điều kiện dừng)  
    {  
        ...  
        //Trả về giá trị hay kết thúc công việc  
    }  
    //Thực hiện một số công việc (nếu có)  
    ... TenHam (<danh sách tham số>);  
    //Thực hiện một số công việc (nếu có)  
}
```

Ví dụ 1: Tính $S(n) = 1 + 2 + 3 + \dots + n$

Trước khi cài đặt hàm đệ qui ta xác định:

- Điều kiện dừng: $S(0) = 0$.

- Quy tắc (công thức) tính: $S(n) = S(n-1) + n$.

Ta cài đặt hàm đệ qui như sau:

```

long TongS (int n)
{
    if(n==0)
        return 0;
    return ( TongS(n-1) + n );
}

```

Ví dụ 2: Tính $P(n) = n!$

Trước khi cài đặt hàm đệ qui ta xác định:

- Điều kiện dừng: $P(0) = 0! = 1$.
- Quy tắc (công thức) tính: $P(n) = P(n-1) * n$.

Ta cài đặt hàm đệ qui như sau:

```

long GiaiThua (int n)
{
    if(n==0)
        return 1;
    return ( GiaiThua(n-1) * n );
}

```

b. Đệ qui nhị phân

Trong thân của hàm có hai lời gọi hàm gọi lại chính nó một cách tường minh.

```

<Kiểu dữ liệu hàm> TenHam (<danh sách tham số>)
{
    if (điều kiện dừng)
    {
        ...
        //Trả về giá trị hay kết thúc công việc
    }
    //Thực hiện một số công việc (nếu có)
    ... TenHam (<danh sách tham số>); //Giải quyết vấn đề nhỏ hơn
    //Thực hiện một số công việc (nếu có)
    ... TenHam (<danh sách tham số>); //Giải quyết vấn đề còn lại
    //Thực hiện một số công việc (nếu có)
}

```

Ví dụ 1: Tính số hạng thứ n của dãy Fibonacci được định nghĩa như sau:

$$f_1 = f_0 = 1 ;$$

$$f_n = f_{n-1} + f_{n-2} ; \quad (n > 1)$$

Trước khi cài đặt hàm đệ qui ta xác định:

- Điều kiện dừng: $f(0) = f(1) = 1$.

Ta cài đặt hàm đệ qui như sau:

```

long Fibonacci (int n)
{
    if(n==0 || n==1)
        return 1;
    return Fibonacci(n-1) + Fibonacci(n-2);
}

```

Ví dụ 2: Cho dãy số nguyên a gồm n phần tử có thứ tự tăng dần. Tìm phần tử có giá trị x có xuất hiện trong mảng không?

Trước khi cài đặt hàm đệ qui ta xác định:

- Điều kiện dừng: Tìm thấy x hoặc xét hết các phần tử.

- Giải thuật:

Do dãy số đã có thứ tự tăng nên ta có thể áp dụng cách tìm kiếm theo phương pháp nhị phân. Ý tưởng của phương pháp này là tại mỗi bước ta tiến hành so sánh x với phần tử nằm ở vị trí giữa của dãy để thu hẹp phạm vi tìm.

Goi: l: biên trái của dãy (ban đầu l=0).

r: biên phải của dãy (ban đầu r = n-1).

m: vị trí ở giữa ($m = (l+r)/2$).

l	m	R				
↓	↓	↓				
a[0]	a[1]	...	a[(l+r)/2]	...	a[n-2]	a[n-1]

Thu hẹp dựa vào giá trị của phần tử ở giữa, có hai trường hợp:

i. Nếu x lớn hơn phần tử ở giữa thì x chỉ có thể xuất hiện ở bên phải vị trí này. (từ m+1 đến r).

ii. Ngược lại nếu x nhỏ hơn phần tử ở giữa thì x chỉ có thể xuất hiện ở bên trái vị trí này. (từ l đến m-1).

Quá trình này thực hiện cho đến khi gặp phần tử có giá trị x, hoặc đã xét hết các phần tử.

Ta cài đặt hàm đệ qui như sau:

```

int TimNhiPhan(int a[], int l, int r, int x)
{
    int m = (l+r)/2;
    if(l>r)
        return -1; // Không có phần tử x
    if(a[m]>x) return TimNhiPhan(a, l, m-1, x);
    if(a[m]<x) return TimNhiPhan(a, m+1, r, x);
    return m; // Trả về vị trí tìm thấy
}

```

}

Ví dụ 3: Bài toán tháp Hà Nội:

Bước 1: Di chuyển $n - 1$ đĩa nhỏ hơn từ cọc A sang cọc B.

Bước 2: Di chuyển đĩa còn lại từ cọc A sang cọc C.

Bước 3: Di chuyển $n - 1$ đĩa nhỏ hơn từ cọc B sang cọc C.

Ta cài đặt hàm đệ qui như sau:

```
void ThapHaNoi (int n, char A, char B, char C)
{
    if (n == 1)
        printf("Di chuyen dia tren cung tu %d den %d\n", A, C);
    else
    {
        ThapHaNoi(n-1, A, C, B);
        ThapHaNoi(1, A, B, C);
        ThapHaNoi(n-1, B, A, C);
    }
}
```

c. Đệ qui phi tuyến

Trong thân của hàm có lời gọi hàm gọi lại chính nó được đặt bên trong vòng lặp.

```
<Kiểu dữ liệu hàm> TenHam (<danh sách tham số>)
{
    for (int i = 1; i <= n; i++)
    {
        //Thực hiện một số công việc (nếu có)
        if (điều kiện dừng)
        {
            ...
            //Trả về giá trị hay kết thúc công việc
        }
        else
        {
            //Thực hiện một số công việc (nếu có)
            TenHam (<danh sách tham số>);
        }
    }
}
```

Ví dụ: Tính số hạng thứ n của dãy $\{X_n\}$ được định nghĩa như sau:

$$X_0 = 1 ;$$

$$X_n = n^2 X_0 + (n-1)^2 X_1 + \dots + 1^2 X_{n-1} ; \quad (n \geq 1)$$

Trước khi cài đặt hàm đệ qui ta xác định:

- Điều kiện dừng: $X(0) = 1$.

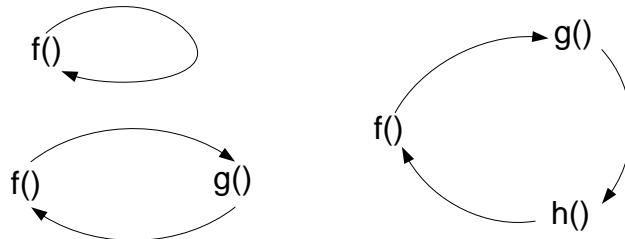
Ta cài đặt hàm đệ qui như sau:

```

long TinhXn (int n)
{
    if(n==0)
        return 1;
    long s = 0;
    for (int i=1; i<=n; i++)
        s = s + i * i * TinhXn(n-i);
    return s;
}
    
```

d. Đệ qui hỗ tương

Trong thân của hàm này có lời gọi hàm đến hàm kia và trong thân của hàm kia có lời gọi hàm tới hàm này.



```

<Kiểu dữ liệu hàm> TenHam2 (<danh sách tham số>);
<Kiểu dữ liệu hàm> TenHam1 (<danh sách tham số>)
{
    //Thực hiện một số công việc (nếu có)
    ...TenHam2 (<danh sách tham số>);
    //Thực hiện một số công việc (nếu có)
}

<Kiểu dữ liệu hàm> TenHam2 (<danh sách tham số>)
{
    //Thực hiện một số công việc (nếu có)
    ...TenHam1 (<danh sách tham số>);
    //Thực hiện một số công việc (nếu có)
}
    
```

Ví dụ: Tính số hạng thứ n của hai dãy $\{X_n\}$, $\{Y_n\}$ được định nghĩa như sau:

$$\begin{aligned}
 X_0 &= Y_0 = 1 ; \\
 X_n &= X_{n-1} + Y_{n-1}; & (n > 0) \\
 Y_n &= n^2 X_{n-1} + Y_{n-1}; & (n > 0)
 \end{aligned}$$

Trước khi cài đặt hàm đệ qui ta xác định:

- Điều kiện dừng: $X(0) = Y(0) = 1$.

Ta cài đặt hàm đệ qui như sau:

```

long TinhXn(int n);
long TinhYn(int n)
{
    if(n==0)
        return 1;
    return TinhXn(n-1) + TinhYn(n-1);
}

long TinhXn (int n)
{
    if(n==0)
        return 1;
    return n*TinhXn(n-1) + TinhYn(n-1);
}

```

I.3. Tìm hiểu cách hoạt động của hàm đệ qui

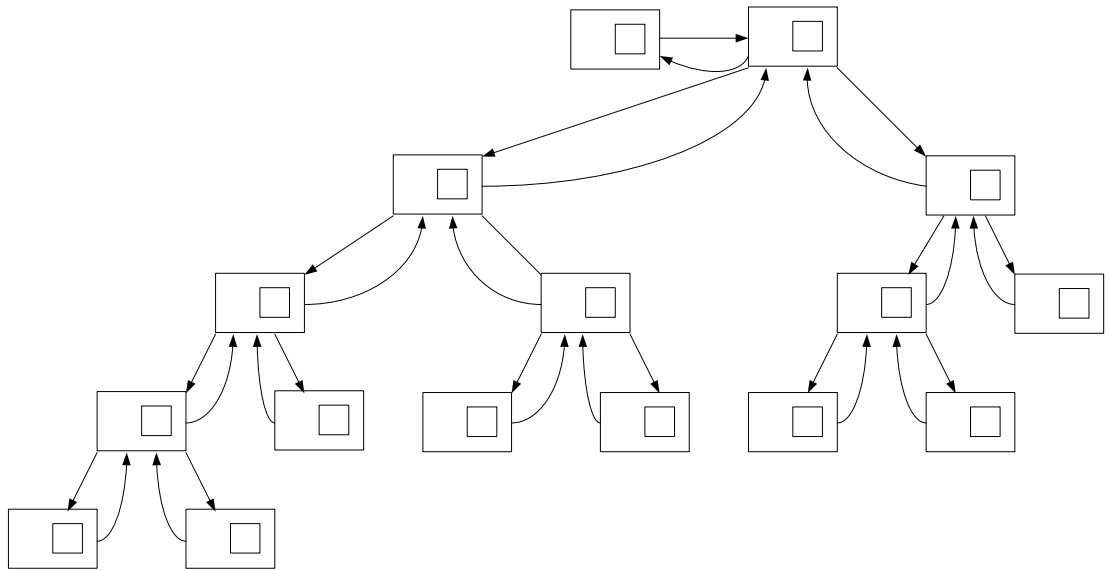
Phục vụ cho công việc kiểm chứng kết quả thực thi của chương trình bằng tay.

Ví dụ 1: Lấy lại ví dụ tính $P(n) = n!$ bằng phương pháp đệ qui như đã mô tả cài đặt ở trên với $n = 5$



Lệnh gọi khởi đầu trong hàm main(), truyền đến hàm GiaiThua(). Ở đó, giá trị của tham số n là 5, do đó nó gọi GiaiThua(4), truyền 4 đến hàm GiaiThua(). Ở đó giá trị của tham số n là 4, do đó nó gọi GiaiThua(3), truyền 3 đến hàm GiaiThua(). Tiến trình này tiếp tục (đệ quy) đến khi gọi GiaiThua(1) được thực hiện từ bên trong lệnh gọi GiaiThua(2). Ở đó, giá trị của tham số n là 1, do đó nó trả về giá trị 1, mà không thực hiện thêm bất kì lệnh gọi nào. Sau đó lần ngược về lệnh gọi GiaiThua(2) trả $2*1=2$ trở về lệnh gọi GiaiThua(3). Sau đó lệnh gọi GiaiThua(3) trả $3*2=6$ trở về lệnh gọi GiaiThua(4). Sau đó lệnh gọi GiaiThua(4) trả $4*6=24$ trở về lệnh gọi GiaiThua(5). Sau cùng, lệnh gọi GiaiThua(5) trả về giá trị 120 cho hàm main().

Ví dụ 2: Lấy lại ví dụ tính số hạng thứ n của dãy Fibonacci như đã mô tả cài đặt ở trên với $n = 5$, quá trình thực hiện tương tự như trong ví dụ trước, ta có sơ đồ sau:



I.4. Ví dụ

Viết chương trình nhập vào mảng một chiều số nguyên a, xuất ra màn hình và tính tổng các phần tử có giá trị chẵn bằng phương pháp đệ qui.

```

#define MAX 100
void Nhap(int a[], int n)
{
    if(n==0)
        return;
    Nhap(a, n-1);
    printf("\nNhập phần tử thứ %d: ", n);
    scanf("%d", &a[n-1]);
}

void Xuat(int a[], int n)
{
    if(n==0)
        return;
    Xuat(a, n-1);
    printf("%d\t", a[n-1]);
}

long TongChan(int a[], int n)
{
    if(n==0)
        return 0;
    int s = TongChan(a, n-1);
    if(a[n-1]%2==0)
        s+=a[n-1];
    return s;
}
    
```

```

void main()
{
    int a[MAX], n;
    long s;

    printf("\nNhap so phan tu cua mang: ");
    scanf("%d", &n);
    Nhap(a, n);
    Xuat(a, n);
    s=TongChan(a, n);
    printf("\nTong cac so chan trong mang la: %ld", s);
    getch();
}

```

II. BÀI TẬP

Viết hàm đệ qui thực hiện các yêu cầu sau:

II.1. Bài tập cơ bản

1. Cài đặt lại những bài tập ở chương mảng một chiều.
2. Tìm chữ số có giá trị lớn nhất của số nguyên dương n.
3. Hãy xây dựng một dãy gồm N số có giá trị từ 1 đến K cho trước, sau cho không có hai dãy con liên tiếp đứng kề nhau.

Ví dụ: $N = 6$

$K = 3$

Kết quả: 121312

4. Tìm ước số chung lớn nhất của hai số nguyên dương a và b.
5. Tìm chữ số đầu tiên của số nguyên dương n.
6. Tìm dãy nhị phân dài nhất sao cho trên dãy này không có hai bộ k bất kỳ trùng nhau. Bộ k là dãy con có k số liên tiếp nhau trên dãy tìm được.

Ví dụ: $k = 3$

Kết quả: 000 101 110 0

7. Tính $P(n) = 1.3.5 \dots (2n+1)$, với $n \geq 0$
8. Tính $S(n) = 1 + 3 + 5 + \dots + (2 \times n + 1)$, với $n \geq 0$
9. Tính $S(n) = 1 - 2 + 3 - 4 + \dots + (-1)^{n+1} n$, với $n > 0$
10. Tính $S(n) = 1 + 1.2 + 1.2.3 + \dots + 1.2.3 \dots n$, với $n > 0$
11. Tính $S(n) = 1^2 + 2^2 + 3^2 + \dots + n^2$, với $n > 0$

12. Tính $S(n) = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$, với $n > 0$
13. Tính $S(n) = 1 + \frac{1}{1+2} + \frac{1}{1+2+3} + \dots + \frac{1}{1+2+3+\dots+n}$, với $n > 0$
14. Tính $P(x, y) = x^y$.
15. Tính $S(n) = 1 + (1+2) + (1+2+3) + \dots + (1+2+3+\dots+n)$, với $n > 0$

II.2. Bài tập luyện tập và nâng cao

16. Cho số nguyên dương n . In ra biểu diễn nhị phân của n .
17. (*) Cài đặt và minh họa bài toán tháp Hà Nội.
18. (**) Cài đặt bài toán mã đi tuần.
19. (**) Cài đặt bài toán tám hậu.
20. (*) Tính $S(n) = \sqrt{n + \sqrt{(n-1) + \sqrt{(n-2) + \dots + \sqrt{1}}}}$, với $n > 0$
21. (*) Tính $S(n) = \sqrt{1 + \sqrt{2 + \sqrt{3 + \dots + \sqrt{n}}}}$, với $n > 0$
22. (*) Tính $S(n) = \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{\ddots \frac{1}{1 + \frac{1}{1+1}}}}}}$ có n dấu phân số.

III. KẾT LUẬN

- ❖ Đệ qui cung cấp cho ta cơ chế giải quyết các bài toán phức tạp một cách đơn giản hơn.
- ❖ Xây dựng hàm đệ qui thông qua việc **xác định điều kiện dừng** và **bước thực hiện tiếp theo**.
- ❖ Chỉ nên cài đặt bằng phương pháp đệ qui khi không còn cách giải quyết bằng cách lặp thông thường.

CHƯƠNG 10 LẬP TRÌNH THEO PHƯƠNG PHÁP PROJECT

I. MỤC TIÊU

Chia một chương trình lớn thành các tập tin nhỏ hơn, mỗi tập tin chứa các khai báo nguyên mẫu hàm, cài đặt các hàm và dữ liệu thực hiện một số chức năng nhất định.

Việc phân chia này giúp quá trình lập trình:

- ❖ Dễ kiểm soát các lệnh và kiểm lỗi.
- ❖ Tránh được giới hạn kích thước tập tin quá lớn của ngôn ngữ lập trình.

II. PHƯƠNG PHÁP

II.1. Tạo một project mới

Bước 1: Tạo thư mục sẽ chứa toàn bộ chương trình sẽ được cài đặt.

Bước 2: Khởi động Borland C++ 3.1.

Bước 3: Thay đổi đường dẫn đến thư mục vừa tạo.

Vào menu **File\Change Dir ...** sau đó chọn đường dẫn thư mục và chọn OK.

Bước 4: Tạo Project.

Vào menu **Project\Open Project** sau đó đặt tên cho project tương ứng, chọn OK.

(Lưu ý: Xem đường dẫn của file Project có nằm đúng thư mục vừa tạo ở bước 1 hay không. Nếu cần có thể chỉnh sửa lại đường dẫn).

Bước 5: Thêm file vào Project.

Chọn menu **Window\Project** sau đó nhấn phím **Insert** hoặc vào menu **Project\Add Item ...** đặt tên file và chọn OK, muốn loại file khỏi project thì chọn **Project>Delete Item** (Hoặc khi đang trong cửa sổ project vừa tạo nhấn phím **insert** để thêm file, muốn xóa chọn file rồi nhấn **delete**).

Lưu ý: Chỉ Insert các file chứa cài đặt lớp và hàm main (.cpp), không insert file header do người dùng định nghĩa (*.h).*

II.2. Mở project có sẵn

Bước 1: Đóng project trước (nếu có).

Vào menu **Project\Close Project**.

Bước 2: Mở project.

Vào menu **Project\Open Project** chọn đường dẫn đến file project cần thực hiện, chọn OK.

Bước 3: Hiệu chỉnh đường dẫn thư viện của BC++ 3.1.

Việc tạo project ở các máy với thông số cài đặt BC++3.1 khác nhau sẽ dẫn đến đường dẫn thư viện hàm của các máy cũng khác nhau, do vậy khi biên dịch sẽ gặp lỗi về thư viện hàm trong BC++3.1.

Vào menu **Options\Directories...** sau đó hiệu chỉnh lại đường dẫn đến thư mục chứa thư viện hàm trong các ô **Include** và **Library** cho đúng với đường dẫn cài BC++3.1 (*Đường dẫn đến thư mục INCLUDE và thư mục BIN của BC++3.1 trên máy đang sử dụng*).

II.3. Một số lưu ý

Nên chia từng file theo từng nhóm hàm. Mỗi một project phải có tối thiểu **3 file** như sau:

- **File header (*.h):** Tạo thư viện tự định nghĩa. Chứa các khai báo nguyên mẫu hàm, kiểu dữ liệu, ...
- **File cài đặt hàm (*.cpp):** Chứa các cài đặt hàm theo nhóm. Nếu có sử dụng thư viện tự định nghĩa thì phải include file chứa thư viện đó vào.
- **File chứa hàm main() (m*.cpp):** Chứa hàm chính (hàm **main()**).

⚠ *Khi cài đặt hay chỉnh sửa một hàm nào đó trước hết phải xem xét hàm đó thuộc nhóm hàm nào và sau đó mở file của nhóm tương ứng để hiệu chỉnh.*

II.4. Ví dụ minh họa

Viết chương trình nhập thông tin của học sinh gồm: họ tên học sinh, điểm văn và toán, xuất thông tin và tính điểm trung bình cho học sinh đó.

Ta chia chức năng chương trình theo các nhóm chức năng để dễ quản lý, gồm các file sau:

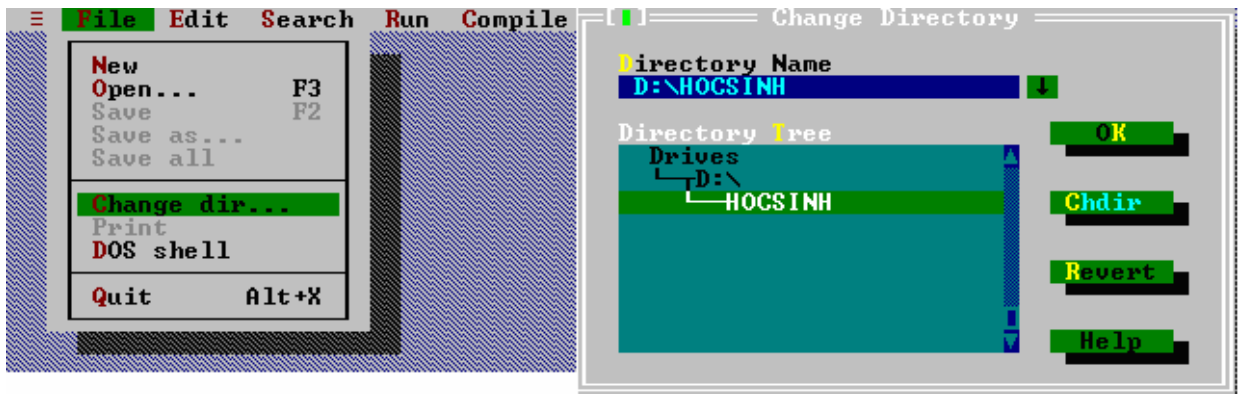
- ❖ *File **hocsinh.h**: Chứa các khai báo biến và nguyên mẫu hàm.*
- ❖ *File **mhocsinh.cpp**: Chứa hàm main().*
- ❖ *File **xuat.cpp**: Chứa các thao tác xuất thông tin học sinh, ...*
- ❖ *File **nhap.cpp**: Chứa các thao tác nhập thông tin học sinh, ...*
- ❖ *File **tinhtoan.cpp**: Chứa các thao tác tính điểm trung bình, ...*

Bước 1: Tạo thư mục **HOCSINH** sẽ chứa toàn bộ các file của chương trình sẽ được cài đặt (Ví dụ tạo ở ổ đĩa D:).

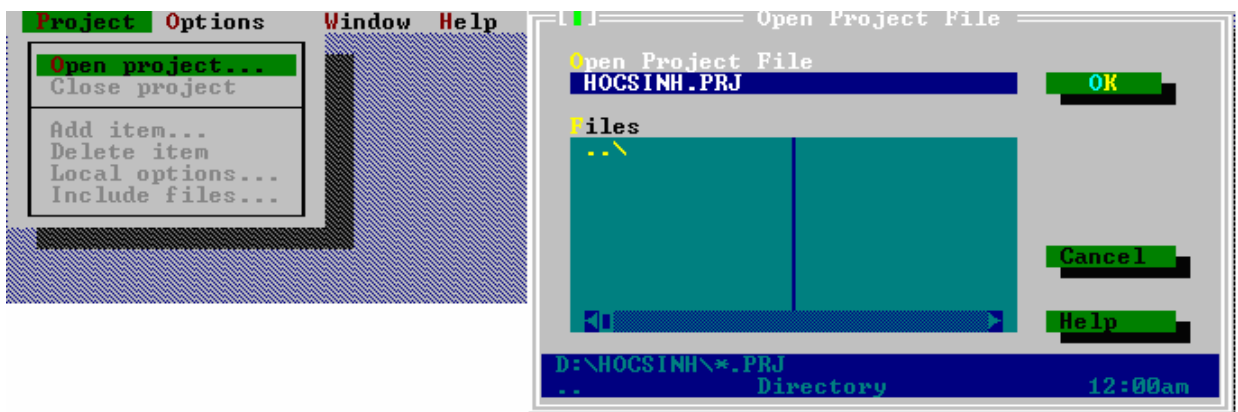
Bước 2: Khởi động Borland C++ 3.1.

Bước 3: Thay đổi đường dẫn đến thư mục **HOCSINH** vừa tạo.

❖ Chọn thư mục **HOCSINH**



Bước 4: Tạo Project: Đặt tên file project là **hocsinh**



❖ Cài đặt file **hocsinh.h** trong thư mục **HOCSINH**.

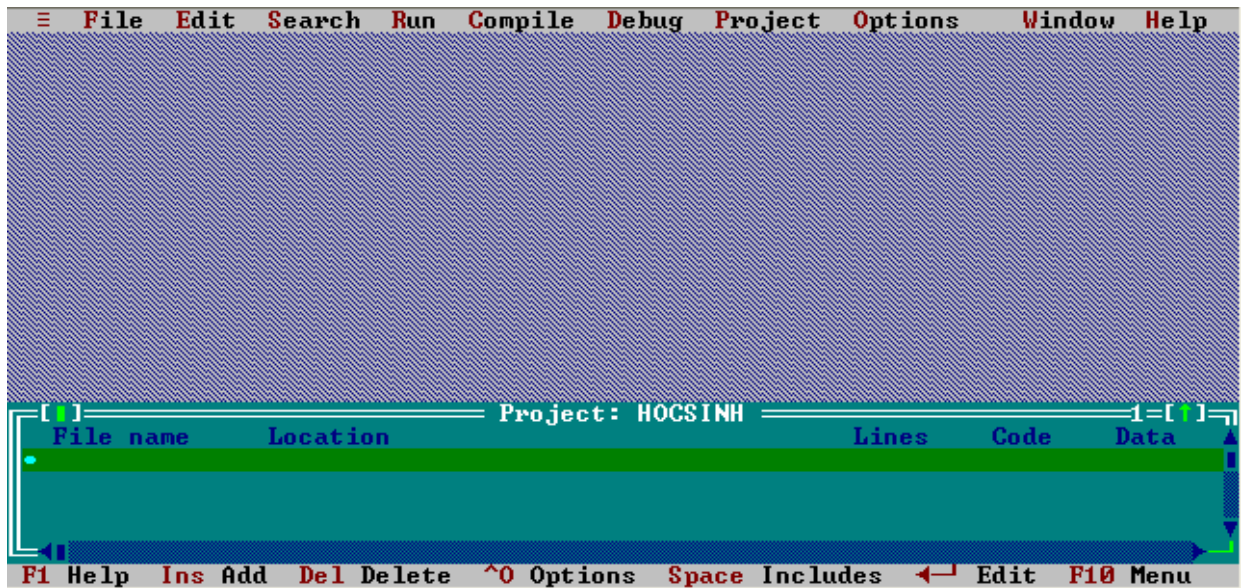
```
#ifndef _HOCSINH_H
#define _HOCSINH_H
```

```
typedef struct HOCSINH
{
    char hoten[30];
    int toan, van;
};
```

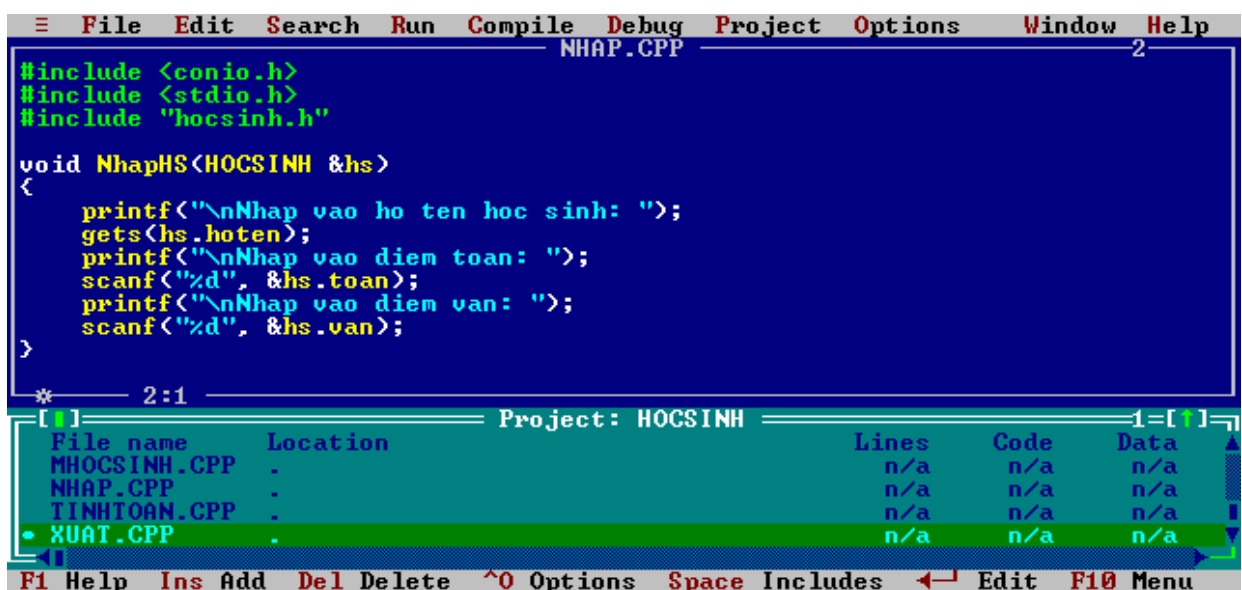
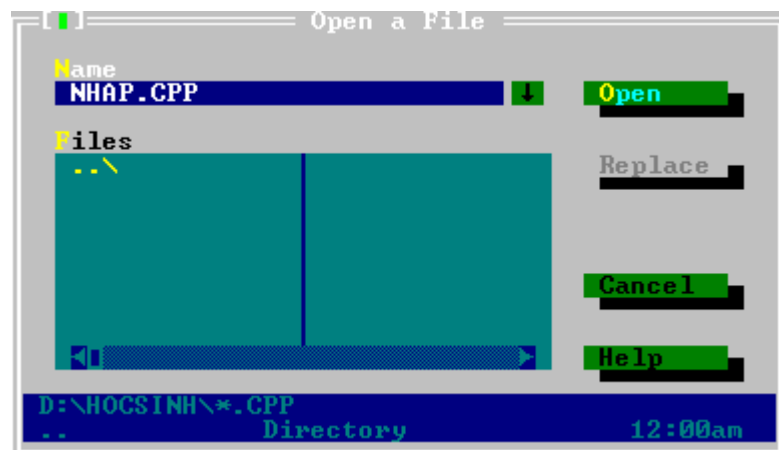
```
void NhapHS(HOCSINH &hs);
float TinhDTB(HOCSINH hs);
void XuatHS(HOCSINH hs);
```

```
#endif
```

Bước 5: Thêm file vào Project.



- ❖ Nhấn F3, đặt tên file mới là **nhap.cpp** và viết hàm nhập. Tương tự cho những file: **xuat.cpp**, **tinhtoan.cpp** và file **mhocsinh.cpp**.



- ❖ Nội dung file **nhap.cpp**.

```
#include <conio.h>
#include <stdio.h>

// Sử dụng kiểu dữ liệu HOCSINH và khai báo nguyên mẫu hàm
#include "hocsinh.h"

void NhapHS(HOCSINH &hs)
{
    printf("\nNhập vào họ tên học sinh: ");
    gets(hs.hoten);
    printf("\nNhập vào điểm toán: ");
    scanf("%d", &hs.toan);
    printf("\nNhập vào điểm văn: ");
    scanf("%d", &hs.van);
}
```

- ❖ Nội dung file **xuat.cpp**.

```
#include <conio.h>
#include <stdio.h>

// Sử dụng kiểu dữ liệu HOCSINH và khai báo nguyên mẫu hàm
#include "hocsinh.h"

void XuatHS(HOCSINH hs)
{
    printf("\nHọ tên học sinh: %s", hs.hoten);
    printf("\nĐiểm toán: %d \nĐiểm văn: %d", hs.toan, hs.van);
    printf("\nĐiểm trung bình: %.2f", TinhDTB(hs));
}
```

- ❖ Nội dung file **tinhtoan.cpp**.

```
// Sử dụng kiểu dữ liệu HOCSINH và khai báo nguyên mẫu hàm
#include "hocsinh.h"

float TinhDTB(HOCSINH hs)
{
    return (hs.toan + hs.van)/2.0;
}
```

- ❖ Nội dung file **mhocsinh.cpp**.

```
#include <conio.h>
#include <stdio.h>

// Sử dụng kiểu dữ liệu HOCSINH và khai báo nguyên mẫu hàm
#include "hocsinh.h"
```

```
void main()  
{  
    clrscr();  
    HOCSINH hs;  
  
    NhapHS(hs);  
    printf("\nKet qua:\n");  
    XuatHS(hs);  
  
    getch();  
}
```

- ❖ Nhấn F9 để biên dịch và kiểm lỗi.
- ❖ Nhấn Ctrl + F9 để thực thi chương trình.

Ví dụ kết quả chạy chương trình

Nhap vào họ tên học sinh: Nguyen Van A
Nhap vào điểm toán: 6
Nhap vào điểm văn: 5

Ket qua:

Ho ten học sinh: Nguyen Van A
Diem toán: 6
Diem văn: 5
Diem trung bình: 5.50

III. BÀI TẬP

Cài đặt các bài tập ở chương mảng cấu trúc bằng phương pháp tạo **project**.

PHỤ LỤC 1 **ĐỀ THI MẪU**

ĐỀ SỐ 01

Thời gian: 120 phút
(Không tham khảo tài liệu)
⊕ ⊕ ⊕ ⊕

Câu 1: Viết chương trình tính tổng: $S(n) = 1! + 2! + \dots + n!$

Câu 2: Viết chương trình thực hiện các yêu cầu sau:

- Nhập mảng một chiều các số nguyên.
- Đếm số lượng giá trị chẵn âm trong mảng.
- Tìm số lẻ cuối cùng trong mảng.

Câu 3: Cho ma trận các số thực. Viết hàm tìm giá trị trong ma trận xa giá trị x nhất.

float **xanhat**(float a[][100], int m, int n, float x);

Câu 4: Hãy khai báo kiểu dữ liệu biểu diễn khái niệm điểm trong mặt phẳng Oxy (DIEM).

- Viết hàm nhập tọa độ điểm.

void **nhap**(DIEM &P);

- Viết hàm xuất tọa độ điểm.

void **xuat**(DIEM P);

- Viết hàm tính khoảng cách giữa 2 điểm.

float **khoangcach**(DIEM P, DIEM Q);

ĐỀ SỐ 02

Thời gian: 120 phút
(Không tham khảo tài liệu)
⊕ ⊕ ⊕ ⊕

Câu 1: Viết chương trình tính tổng: $S(x, n) = x + x^2 + \dots + x^n$

Câu 2: Viết chương trình thực hiện các yêu cầu sau:

- Nhập mảng một chiều các số nguyên.

- b. Đếm số lượng giá trị lẻ dương trong mảng.
- c. Tìm số chẵn cuối cùng trong mảng.

Câu 3: Cho ma trận các số thực. Viết hàm tìm giá trị trong ma trận gần giá trị x nhất.

float **gannhat**(float a[][100], int m, int n, float x);

Câu 4: Hãy khai báo kiểu dữ liệu biểu diễn khái niệm phân số (**PHANSO**)

- a. Viết hàm nhập phân số.

void **nhap**(PHANSO &x);

- b. Viết hàm xuất phân số.

void **xuat**(PHANSO x);

- c. Viết hàm tính tổng hai phân số.

PHANSO **tong**(PHANSO x, PHANSO y);

ĐỀ SỐ 03

Thời gian: 120 phút

(Không tham khảo tài liệu)

⊕ ⊕ ⊕ ⊕

Câu 1:

$$S_n = \frac{1}{2} + \frac{3}{4} + \frac{5}{6} + \dots + \frac{2n+1}{2n+2} \text{ Với } n \text{ nguyên dương } (n > 0)$$

1. Vẽ lưu đồ thuật toán (Flowchart) tính tổng trên.
2. Viết hàm tính tổng trên bằng **phương pháp đệ quy**.

Câu 2:

Cho mảng một chiều các số thực A kích thước n ($0 < n \leq 100$). Hãy xây dựng hàm thực hiện các yêu cầu sau:

1. Nhập giá trị các phần tử vào mảng.
2. Tìm và trả về vị trí của phần tử có giá trị **âm đầu tiên** trong mảng. Nếu không có giá trị âm thì trả về -1.
3. Tìm và trả về giá trị phần tử **âm lớn nhất** trong mảng a. Nếu mảng không có phần tử chứa giá trị âm thì trả về 0.

Câu 3:

Cho ma trận vuông các số nguyên A kích thước $n \times n$ ($3 < n < 10$). Hãy xây dựng các hàm cho phép thực hiện các yêu cầu sau:

1. Nhập giá trị các phần tử vào ma trận.
2. Đếm và trả về số lượng các phần tử là **số nguyên tố** trong ma trận.
3. Tính **trung bình cộng** các phần tử trên đường chéo chính.

Câu 4:

Hãy khai báo kiểu dữ liệu để biểu diễn thông tin của một nhân viên (NHANVIEN).

Biết một nhân viên gồm:

- Mã nhân viên (MaNV): Chuỗi tối đa 5 ký tự.
- Tên nhân viên (TenNV): Chuỗi tối đa 30 ký tự.
- Chức vụ (ChucVu): Chuỗi tối đa 20 ký tự
(gồm các chức vụ: “Truong phong”, “Nhan vien”, “Giam doc”, “Pho giam doc”, ...).
- Số năm làm việc (SoNam): Số nguyên 1 byte.
- Hệ số lương (HeSo): Kiểu số thực.

Cho danh sách gồm n (n>0) nhân viên. Viết các hàm sau:

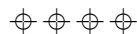
1. Liệt kê các nhân viên có số năm làm việc từ **3 năm trở lên**.
2. Đếm số nhân viên có chức vụ là “**Truong phong**”.

Sắp xếp danh sách nhân viên tăng dần theo **hệ số lương** nhân viên.

ĐỀ SỐ 04

Thời gian: 120 phút

(Không tham khảo tài liệu)



Câu 1:

$S_n = 1 - 2 + 3 - 4 + \dots + (-1)^{n+1} n$ Với n nguyên dương (n>0)

1. Vẽ lưu đồ thuật toán (Flowchart) tính tổng trên.
2. Viết hàm tính tổng trên bằng **phương pháp đệ quy**.

Câu 2:

Cho mảng một chiều các số nguyên A kích thước n ($0 < n \leq 100$). Hãy xây dựng hàm thực hiện các yêu cầu sau:

1. Nhập giá trị các phần tử vào mảng.

2. Tìm và trả về vị trí của phần tử có giá trị là **số nguyên tố đầu tiên** trong mảng. Nếu không có giá trị là số nguyên tố thì trả về -1.
3. Tìm và trả về giá trị phần tử là **số nguyên tố lớn nhất** trong mảng a. Nếu mảng không có phần tử là số nguyên tố thì trả về 0.

Câu 3:

Cho ma trận vuông các số thực A kích thước $n \times n$ ($3 < n < 10$). Hãy xây dựng các hàm cho phép thực hiện các yêu cầu sau:

1. Nhập giá trị các phần tử vào ma trận.
2. Liệt kê những phần tử tại những dòng lẻ trong ma trận.
3. Tính và trả về giá trị trung bình cộng của những phần tử âm trong ma trận.

Câu 4:

Hãy khai báo kiểu dữ liệu để biểu diễn thông tin của một mặt hàng (**MATHANG**).

Biết một mặt hàng gồm:

- Mã hàng (**MaHang**): Chuỗi tối đa 5 ký tự.
- Tên hàng (**TenHang**): Chuỗi tối đa 30 ký tự.
- Số lượng (**SoLuong**): Số nguyên 2 byte.
- Đơn vị tính (**DonViTinh**): Chuỗi tối đa 5 ký tự.
- Đơn giá (**DonGia**): Kiểu số thực.

Cho danh sách gồm n ($n > 0$) mặt hàng. Viết các hàm sau:

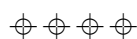
1. Liệt kê các mặt hàng có số lượng **lớn hơn 100**.
2. Tìm và trả về mặt hàng có **thành tiền lớn nhất**
(*thành tiền = số lượng * đơn giá*).

Sắp xếp danh sách các mặt hàng theo thứ tự giảm dần của **đơn giá**.

ĐỀ SỐ 05

Thời gian: 120 phút

(Không tham khảo tài liệu)



Bài 1: Nhập số nguyên n ($0 < n \leq 20$). Viết chương trình xuất n phần tử đầu tiên của hai mảng A và B, cho biết các giá trị được xác định như sau:

$$\begin{cases} A_1 = 1, B_1 = 1 \\ A_i = \sqrt{A_{i-1}^2 + B_{i-1}^2} \\ B_i = 2A_i + \frac{B_{i-1}}{2} \end{cases}$$

Bài 2: Viết chương trình nhập vào ma trận vuông cấp n với n nhập từ bàn phím. Hãy kiểm tra ma trận này có phải là ma trận tam giác dưới hoặc tam giác trên theo đường chéo phụ không?

Ví dụ:

```
1 0 1 0
3 2 0 0
4 0 0 0
5 0 0 0
```

Ma trận tam giác trên

```
0 0 0 4
0 0 5 1
0 1 2 3
1 0 1 0
```

Ma trận tam giác dưới

Bài 3: Mỗi hồ sơ nhân viên gồm:

- họ tên
- năm sinh
- lương cơ bản

Viết chương trình thực hiện các công việc sau:

- Nhập n hồ sơ với n nhập từ bàn phím.
- In ra họ tên và lương cơ bản của nhân viên có lương cơ bản thấp nhất và nhân viên có lương cơ bản cao nhất.
- Ghi xuống file văn bản (với tên file là hoso.txt) danh sách gồm họ tên, lương cơ bản, phụ cấp và thực lãnh của các nhân viên (mỗi nhân viên một dòng) biết rằng:

Phụ cấp = 30% lương cơ bản

Thực lãnh = lương cơ bản + phụ cấp

ĐỀ SỐ 06

Thời gian: 120 phút

(Không tham khảo tài liệu)

⊕ ⊕ ⊕ ⊕

Bài 1: Nhập vào một dãy số thực kết thúc bởi 0 hoặc đã đủ 20 phần tử

- Sắp xếp dãy theo thứ tự tăng dần.

- b. Cho biết dãy có hội tụ không? (Dãy được hội tụ khi có nửa phần tử trở lên nhỏ hơn trung bình cộng của dãy).

Bài 2: Nhập vào ma trận cấp $m \times n$ với m và n nhập từ bàn phím. Hãy kiểm tra xem ma trận có cân bằng theo cột hay không? (Ma trận cân bằng theo cột khi tổng các giá trị của các cột bên trái bằng tổng các giá trị của các cột bên phải, nếu số cột lẻ thì không tính cột giữa).

Ví dụ:

8	4	5	8	9
3	5	7	4	6
4	9	7	5	1

Tổng bên trái = 33 Tổng bên phải = 33

Kết luận: Ma trận cân bằng theo cột.

Bài 3: Một Album ca nhạc MP3 gồm tối đa 150 ca khúc. Thông tin mỗi ca khúc gồm:

- Tên ca khúc
- Tên nhạc sỹ
- Tên ca sỹ
- Thời gian (tính bằng giây)

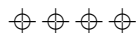
Viết chương trình thực hiện các công việc sau:

- Nhập n ca khúc với n nhập từ bàn phím.
- Xuất tổng thời gian của các ca khúc (hiển thị theo dạng hh:mm:ss) và cho biết tên ca khúc nào có thời gian dài nhất.

Ghi xuống file văn bản (với tên file là mp3.txt) danh sách gồm tên ca khúc, tên nhạc sỹ, tên ca sỹ và thời gian (hiển thị theo dạng hh:mm:ss), mỗi ca khúc chiếm một dòng.

ĐỀ SỐ 07

*Thời gian: 120 phút
(Không tham khảo tài liệu)*



Câu 1. Tính số hạng thứ n của hệ thức truy hồi như sau

$$f(0)=1, f(1)=2$$

$$f(n)=3f(n-1)+2f(n-2) \quad (n \geq 2)$$

bằng hai cách

- Dùng đệ qui
- Khử đệ qui, dùng vòng lặp

Câu 2. Xây dựng một cấu trúc có các thành phần sau

- Mã số học sinh
- Họ và tên học sinh
- Điểm Toán
- Điểm Văn
- Điểm trung bình=(Điểm Toán+Điểm Văn)/2

Viết chương trình nhập dữ liệu của n học sinh và lưu vào một tập tin có tên là HOSOHS.DOC (hay mảng 1 chiều có cấu trúc). Sau đó đọc dữ liệu từ tập tin HOSOHS.DOC (hay mảng 1 chiều có cấu trúc), sắp xếp theo thứ tự Điểm trung bình giảm dần và xuất dữ liệu của từng học sinh ra màn hình

Câu 3. Cho n là một số nguyên dương, tính giá trị biểu thức sau bằng cách viết chương trình sử dụng vòng lặp và tối ưu vòng lặp.

$$1 + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \dots + \frac{1}{n!} \quad (n = 1, 2, 3, \dots)$$

ĐỀ SỐ 08

Thời gian: 120 phút

(Không tham khảo tài liệu)

⊕ ⊕ ⊕ ⊕

PHẦN I: Chọn câu trả lời đúng nhất (5 điểm)

Đánh dấu chéo vào câu trả lời đúng nhất trong các câu trả lời cho mỗi câu hỏi.

Câu 1. Đoạn chương trình sau sẽ cho giá trị của t:

```
for (t=i=0; (i<10) && (t<100); i++, t += 2*i);
```

- a. 90 b. 100 c. 110 d. 120

Câu 2. Cho dãy gồm 12 phần tử $a_0, a_1, a_2, \dots, a_{11}$ như sau:

-9 -9 -5 -2 0 3 7 7 10 15

Dùng thuật toán tìm nhị phân để tìm vị trí phần tử $x = -9$, vị trí tìm được sẽ là:

- a. -1 b. 0 c. 1 d. 2

Câu 3. Chương trình sau:

```
#include<conio.h>  
#include<stdio.h>  
int a=1, b=2, c=3;  
int A(int &a, int b)  
{  
          a += c + 2;  
          b -= a;
```

```
        return a;
    }
    void main()
    {
        printf(" %d %d", A(b, c), a+c);
    }
```

Sẽ in ra:

- a. 7 5 b. 7 4 c. 7 3 d. 7 2

4. Chương trình sau:

```
#include <conio.h>
#include <stdio.h>
int A(int a, int &b)
{
    a += b + 2;
    b -= a;
    return a;
}
void main()
{
    int x = 5;
    printf(" %d %d", A(A(3, x), x), x);
}
```

Sẽ in ra:

- a. 5 5 b. 5 7 c. 7 7 d. 7 5

5. Đoạn chương trình dưới đây khi thực thi sẽ:

```
char buf1[100], buf2[100], *strptr1, *strptr2;
strcpy(buf1, "abcdefghijklmnopqrstvwxyz");
strcpy(buf2, "Hello");
strptr1 = buf1 + 6;
strcpy(strptr1, buf2);
strptr2 = (strptr1 + 4);
strncpy(strptr2, buf2, 4);
printf("%s\n", buf1);
```

Sẽ in ra màn hình:

- a. abcdefHellHelloworldvwxyz b. ghijklmnHellotvwxyz
c. abcdefghijklmnopqrstvwxyz d. abcdefHelloImnopqrstvwxyz

PHẦN II: Lập trình (5 điểm)

Câu 1. Hãy viết hàm kiểm tra một số nguyên không n có phải là số nguyên tố hay không, hàm thực hiện sẽ trả về: 1 nếu n là số nguyên tố, 0 nếu n không là số nguyên tố

```
int LaSNT(unsigned int n);
```

Câu 2. Hãy viết hàm tìm tổng các số nguyên tố nằm trong mảng một chiều a có n phần tử (unsigned int a[100], int n).

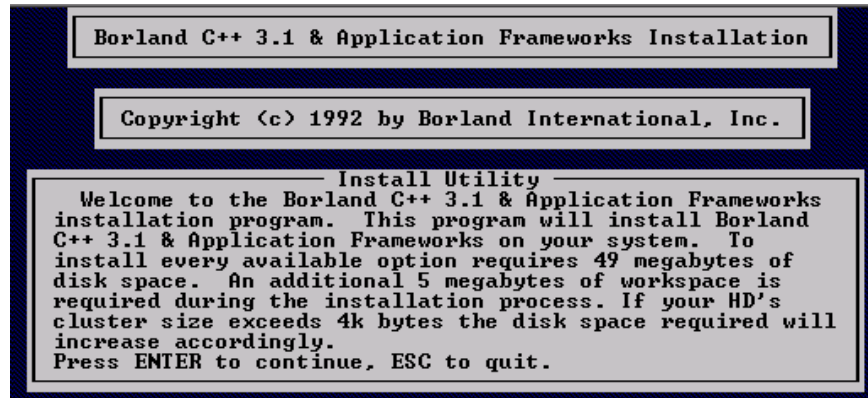
Câu 3. Viết hàm xác định vị trí của số nguyên tố lớn nhất trên mảng a có n phần tử (unsigned int a[100], int n)

(Lưu ý: Có thể làm các câu 2 và 3 mà không cần làm câu 1).

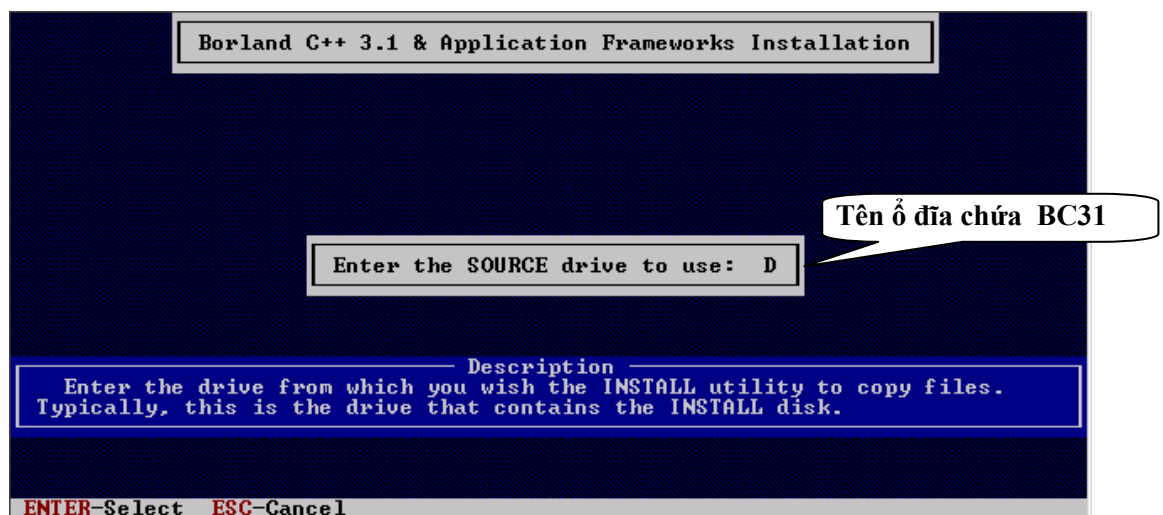
PHỤ LỤC 2 HƯỚNG DẪN VIẾT CHƯƠNG TRÌNH TRÊN MÔI TRƯỜNG BORLAND C++ 3.1 (BC31)

I. CÀI ĐẶT BC3.1

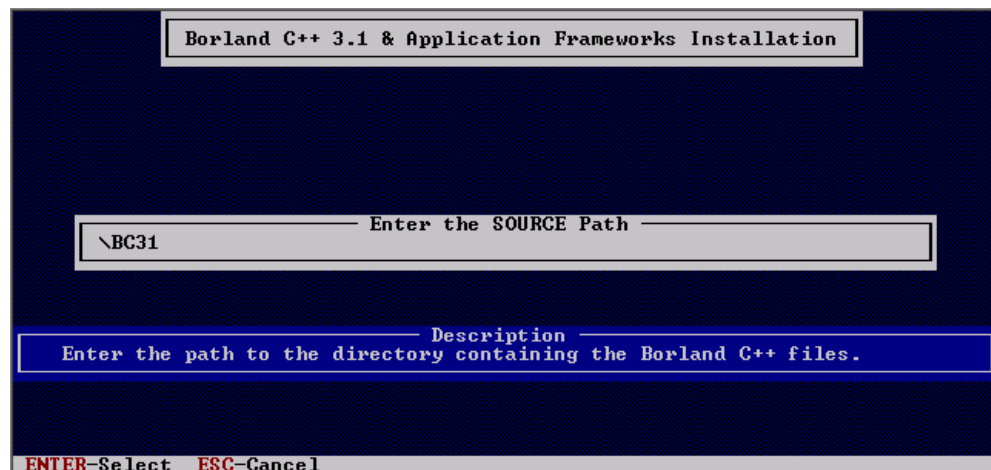
- ❖ Vào thư mục BC3.1 trên đĩa CD chạy file install.exe.



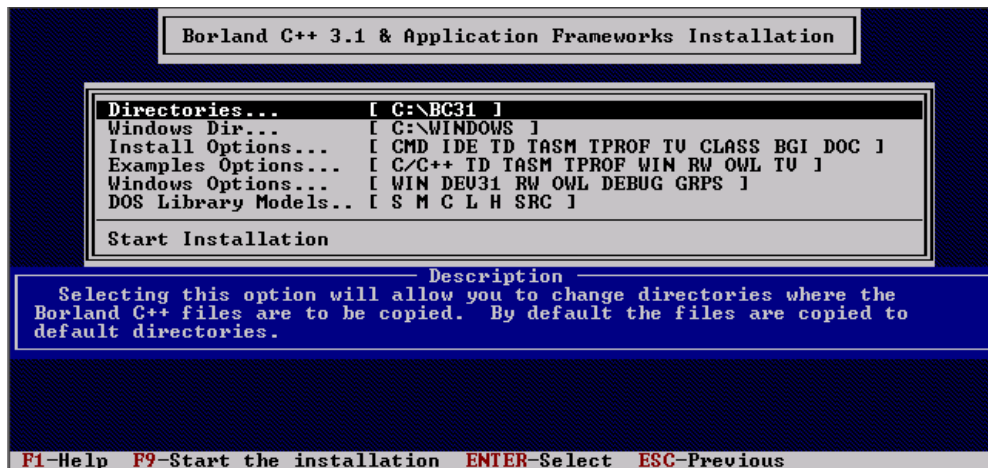
- ❖ Nhấn Enter.



- ❖ Gõ vào ô chứa thư mục nguồn chứa BC3.1 (Ví dụ trên:Giả sử ổ đĩa chứa thư mục BC3.1 trên đĩa CD là D:) → Nhấn Enter.



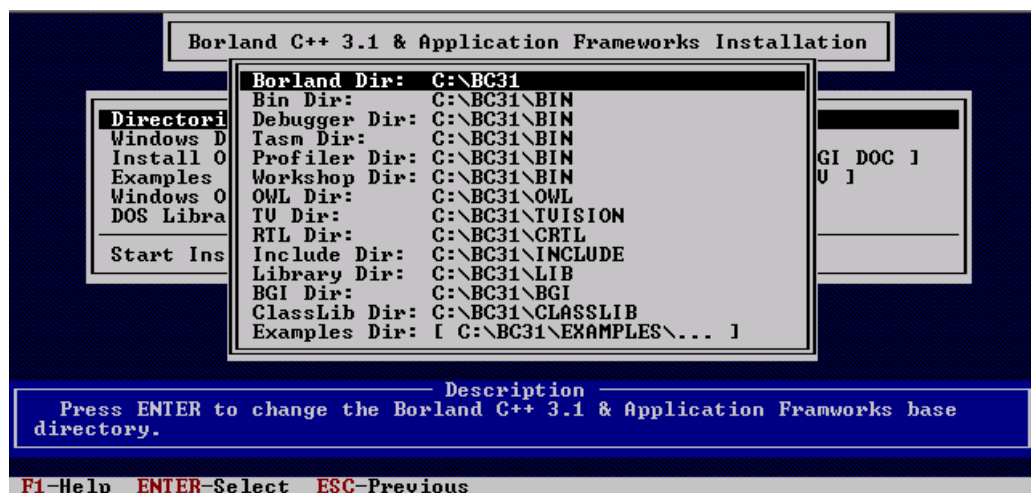
- ❖ Enter tiếp (có thể gỡ lại đường dẫn chứa BC3.1 nếu bước trên gõ sai).



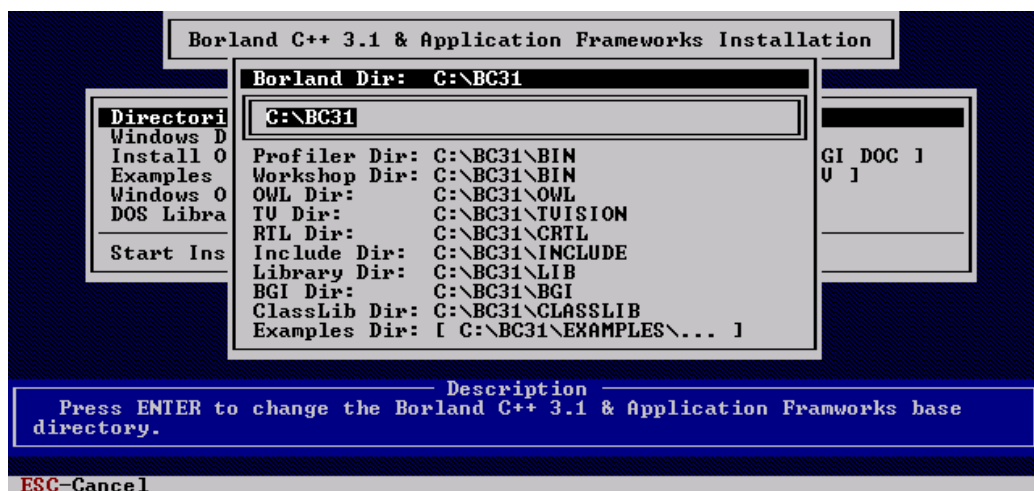
- ❖ Chọn đường dẫn và tên thư mục cần cài đặt BC3.1 lên đĩa cứng.

Ví dụ: Cần cài BC3.1 lên ổ đĩa C: tên thư mục là BC31.

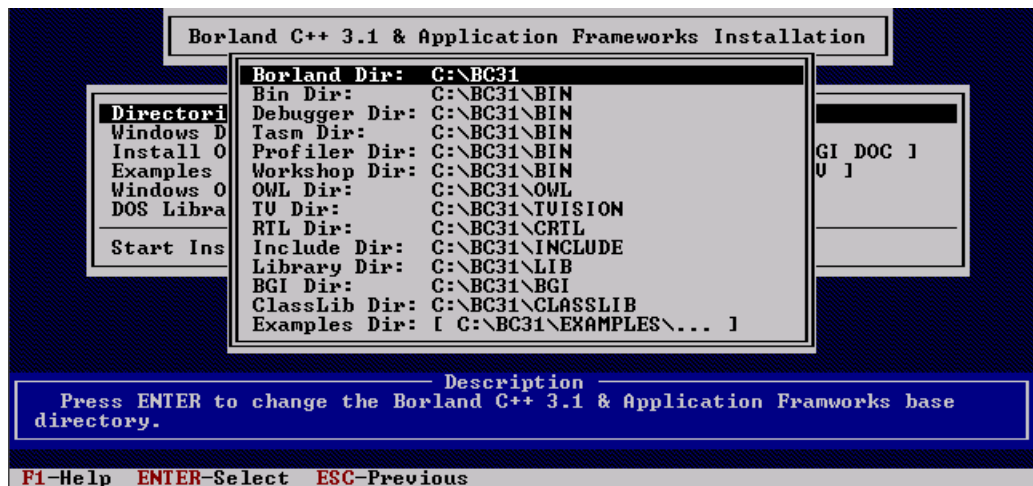
- Để thay đổi thư mục và ổ đĩa cài đặt → di chuyển vệt sáng (dùng phím mũi tên) đến dòng Directories ... như hình trên → sau đó nhấn Enter.



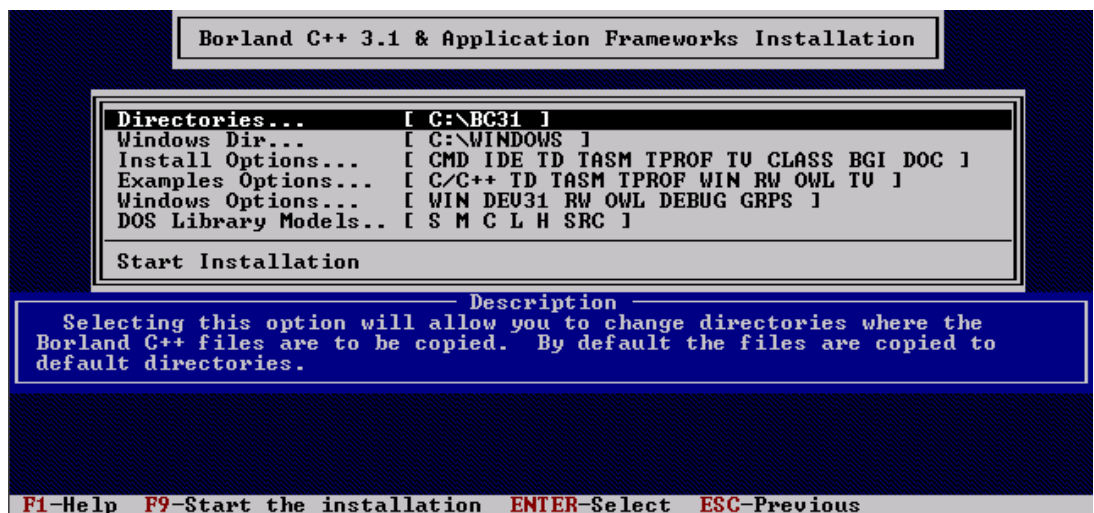
- Nhấn tiếp Enter.



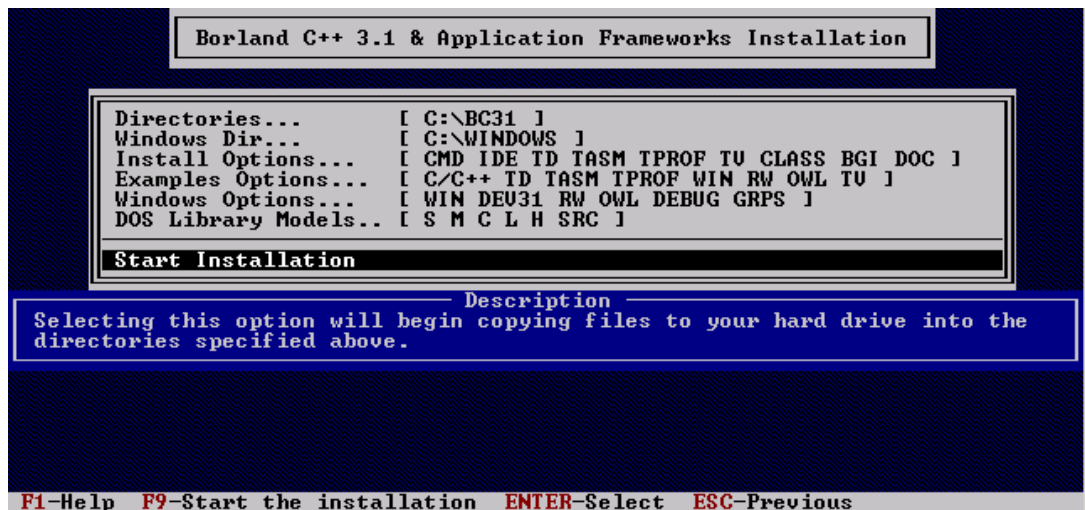
- Gõ tên ổ đĩa và tên thư mục cần cài đặt trong textbox → Enter.
- Sau đó nhấn ESC.



- Tiếp tục nhấn phím ESC.

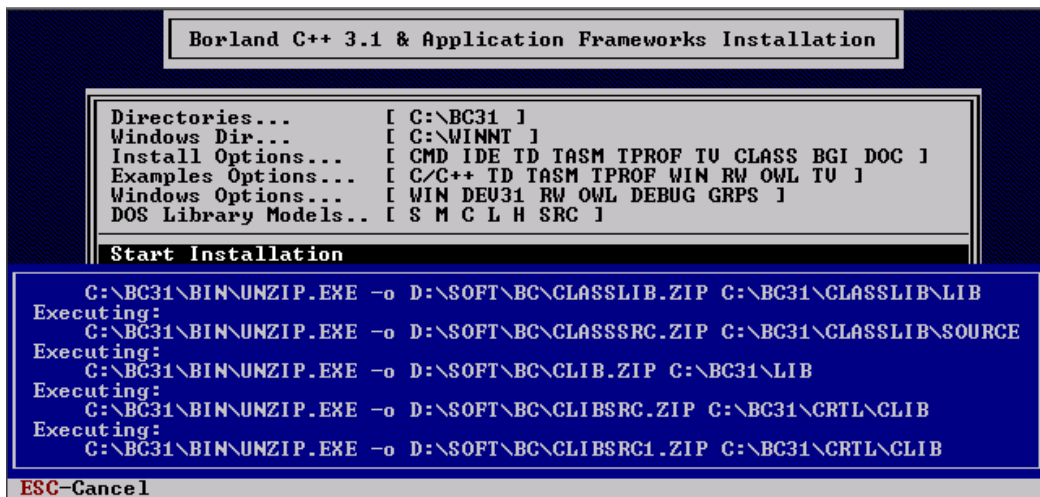


- Tiếp theo kiểm tra xem thư mục cài Windows có đúng đường dẫn như dòng **Windows Dir** hay không (dòng thứ 2). Nếu không đúng thì thay đổi thư mục cho đúng, di chuyển vệt sáng đến đó, thao tác tương tự như thay đổi thư mục BC3.1.
- Thường thì không cần thay đổi vì các máy có cài Windows mặc định là C:\Windows.
- Di chuyển vệt sáng đến dòng **Start Installation** nhấn Enter bắt đầu quá trình cài đặt.

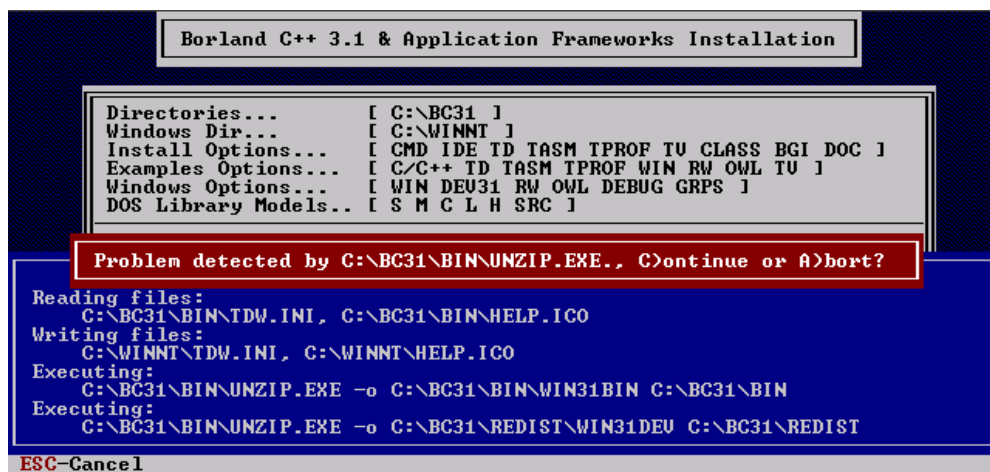


Lưu ý: Ở bước này chỉ thay đổi thư mục cài đặt BC3.1, thư mục Windows (nếu có) còn những mục khác không thay đổi.

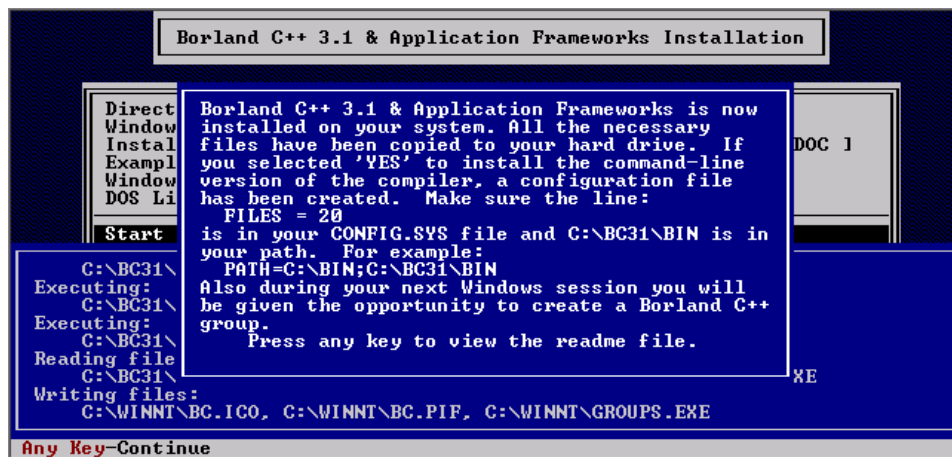
- ❖ Quá trình cài đặt đang thực hiện.



- ❖ Nếu trong quá trình cài đặt gặp thông báo sau:



→ Nhấn phím C để tiếp tục.



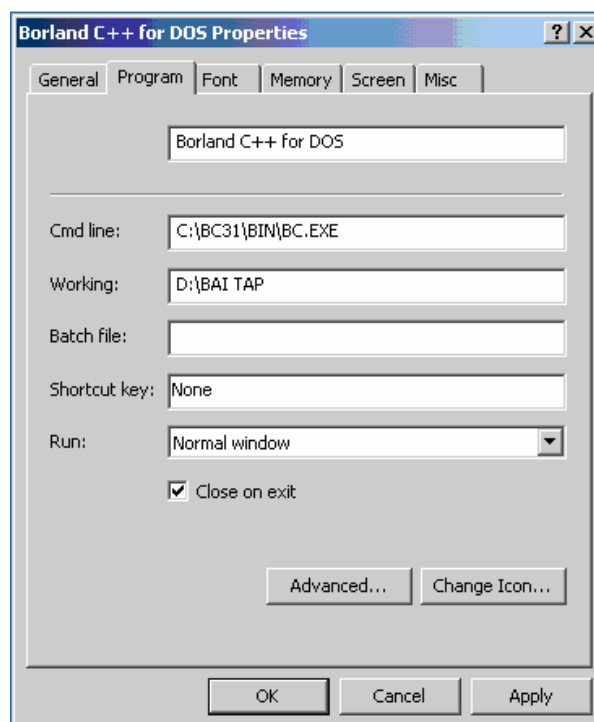
Quá trình cài đặt hoàn tất, nhấn phím **ESC** cho đến khi mất màn hình cài đặt.

❖ Tạo một thư mục để lưu bài tập, chẳng hạn **D:\BaiTap** để làm thư mục làm việc của C, trong quá trình làm bài hay biên dịch chạy chương trình thì tất cả các file đó đều nằm trong thư mục **BaiTap** cho dễ quản lý.

❖ Tạo Shortcut Borland C++3.1 (File bc.exe trong thư mục BIN của thư mục BC31 vừa cài đặt) → Chọn Properties → Chọn Tab Program gõ vào mục Cmd line và Working giống như hình sau nếu cài đặt BC3.1 trên ổ đĩa C:\BC3.1. Nhấn OK.

- Cmd line (đường dẫn đến file chạy BC): C:\BC3.1\BIN\BC.EXE.
- Working (thư mục mới vừa tạo để lưu bài làm): D:\Bai tap

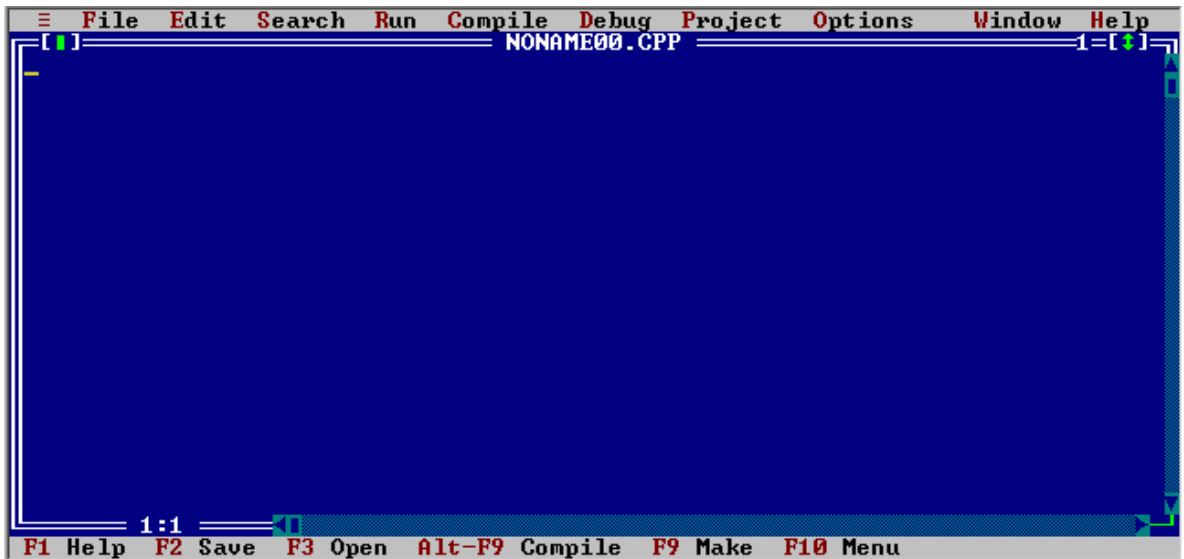
Lưu ý: Đúng đường dẫn thư mục.



II. CÁC BƯỚC VIẾT CHƯƠNG TRÌNH

a. Chuẩn bị viết chương trình

- Khởi động BC3.1



- Vào menu Options\Environment\Editor chỉnh lại Tab size là 4.



b. Các phím chức năng chính

- F3: Mở file chương trình có sẵn.
- F2: Lưu file
- Lưu ý: Chọn đường dẫn và đặt tên file cho đúng. Tên có tối đa 8 ký tự, phần đuôi không cần nhập vào (mặc định là *.cpp).
- F5: Phóng to hoặc trở về kích thước bình thường của cửa sổ soạn thảo.
- F6: Chuyển qua lại các cửa sổ soạn thảo (nếu mở nhiều cửa sổ).
- F9: Biên dịch chương trình. Mục đích là kiểm tra lỗi chương trình.
- Ctr+F9: Thực thi chương trình (Run) khi chương trình không có lỗi.
- Alt+F5: Xem lại màn hình kết quả chương trình đã chạy trước đó.

c. Viết chương trình

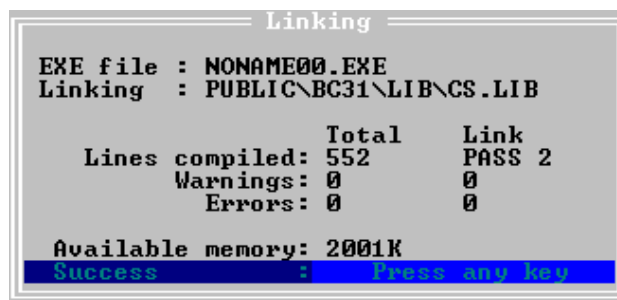
Cấu trúc cơ bản của chương trình gồm phần.

- i. Phần khai báo thư viện hàm.
- ii. Phần khai báo biến toàn cục, khai báo kiểu dữ liệu, khai báo hàm hay khai báo hằng (nếu có).
- iii. Các hàm con (nếu có).
- iv. Hàm main().

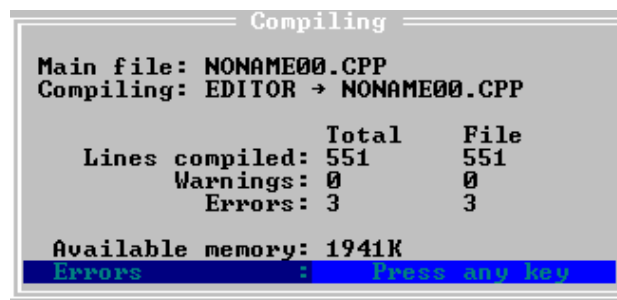
→ Lưu ý trình bày chương trình.

d. Biên dịch và sửa lỗi

- Sau khi soạn thảo xong chương trình nhấn F2 đặt tên chương trình, để đảm bảo chương trình có thể thực thi được, ta phải nhấn F9 để biên dịch.
- Nếu không có lỗi, ta có thể nhấn Ctrl+F9 để thực thi chương trình.



- Nếu máy bị loop nhấn Ctrl+Break+Enter để trở về màn hình soạn thảo.
- Ngược lại, ta cần phải sửa lỗi cho đến khi hết lỗi.



- Các bước thực hiện khi có lỗi:
 - i. Khi hiển thị màn hình báo lỗi, ta phải nhấn phím Enter để xuất hiện cửa sổ mô tả lỗi (**không nhấn phím ESC**).

- ii. Sử dụng phím mũi tên lên xuống để duyệt lên xuống và xem mô tả lỗi. Khi di chuyển để ý quan sát vệt sáng bên trên khung cửa sổ soạn thảo chương trình. Thông thường vệt sáng sẽ cho biết vị trí lỗi (có thể ngay chính tại dòng có lỗi hoặc trên hoặc dưới một dòng). Có nhiều cách sửa lỗi, nhưng để đơn giản chúng ta nên sửa lỗi từ trên xuống.

e. Một số lỗi thường gặp

STT	LỖI	MÔ TẢ	KHẮC PHỤC	VÍ DỤ
LỖI CÚ PHÁP				
1	Statement missing;	Thiếu dấu; khi kết thúc 1 lệnh	Bổ sung thêm dấu ; vào sau khai báo biến hay kết thúc một lệnh.	Sai: int a scanf("%d",&a) Sửa thành: int a; scanf("%d",&a);
2	Compound statement missing }	Thiếu dấu } khi kết thúc khối lệnh hay làm.	Bổ sung thêm dấu } vào tương ứng	Sai : void main() { int a; scanf("%d",&a); if(a>0) printf("Duong"); Sửa thành: void main() { int a; scanf("%d",&a); if(a>0) printf("Duong"); }
3	Unexpected }	Thiếu dấu { khi bắt đầu khối lệnh, hàm hay dư dấu }	Kiểm tra xem có dư dấu } hoặc	Sai : void main()

			thiếu dấu { và sửa tương ứng.	<pre>{ int a; scanf("%d",&a); if(a>0) printf("Duong"); }</pre> <p>Sửa thành:</p> <pre>void main() { int a; scanf("%d",&a); if(a>0) printf("Duong"); }</pre>
4	Misplaced else	Chấm phẩy sau phát biểu if hoặc khối lệnh thực hiện trong phát biểu if chưa đặt trong cặp dấu ngoặc {}		<p>Sai :</p> <pre>if (a%2); printf("a le"); else if(a>10) printf("a chan"); printf(", > 10"); else printf("a < 10");</pre> <p>Sửa thành:</p> <pre>if (a%2) printf("a le"); else if(a>10) { printf("a chan "); printf(", > 10"); } else printf("a < 10");</pre>
5	For statement missing ;	Thiếu thành phần trong cú pháp của vòng lặp for hoặc quên dùng dấu chấm phẩy (;) để ngăn cách các thành phần , ... (Phải có đủ 2 dấu chấm phẩy)	Kiểm tra cho đúng cú pháp: for(<Biểu thức khởi>; <biểu thức điều kiện dừng>; <biểu thức tăng giảm>) Trong biểu thức gán hay nhiều thành phần thì mỗi phần cách nhau bởi dấu phẩy (,)	<p>Sai :</p> <pre>for(int i=0, i<n; i++) { printf("a[%d]: ",i); scanf("%d",&a[i]); }</pre> <p>Sửa thành:</p> <pre>for(int i=0; i<n; i++) { printf("a[%d]: ",i); scanf("%d",&a[i]); }</pre>
6	Function call missing)	Thiếu dấu phẩy phân cách giữa phần định dạng và danh sách biến trong hàm printf và scanf.	Thêm dấu phẩy giữa phần định dạng và danh sách biến.	<p>Sai :</p> <pre>for(int i=0; i<n; i++) { printf("a[%d]: "i); scanf("%d",&a[i]); }</pre>

				<pre> } Sửa thành: for(int i=0; i<n; i++) { printf("a[%d]: ",i); scanf("%d",&a[i]); } </pre>
LỖI KHAI BÁO				
1	Declaration terminated incorrectly	Khai báo tên biến trùng với tên hằng đã định nghĩa trước.	Đổi tên biến.	<pre> Sai : #define MAX 100 void main() { int MAX; } Sửa thành: #define MAX 100 void main() { int x; } </pre>
2	Multiple declaration for 'i'	Khai báo biến trùng tên, khai báo nhiều lần.	Kiểm tra và bỏ bớt khai báo lại biến hoặc đổi tên biến khác.	<pre> Sai : int i; for(int i=0; i<n; i++) scanf("%d",&a[i][j]); Sửa thành: int x; for(i=0; i<n; i++) scanf("%d",&a[i][j]) </pre>
3	Undefined symbol 'a'	Sử dụng biến chưa khai báo.	Khai báo biến.	<pre> Sai : printf("Nhap vào n:"); scanf("%d", &n); Sửa thành: int n; printf("Nhap vào n:"); scanf("%d", &n); </pre>
4	Declaration syntax error	Thiếu dấu ; sau khai báo biến.	Bổ sung dấu ; sau khi kết thúc khai báo biến.	<pre> Sai : int n Sửa thành: Int n; </pre>
THƯ VIỆN HÀM HOẶC SAI TÊN HÀM				
1	Function 'printf' should have a prototype Function 'scanf' should have a prototype Function 'XXX' should have a prototype	Thiếu sai báo thư viện hàm nếu sử dụng hàm thư viện, ngược lại phải kiểm tra xem có khai báo nguyên mẫu hàm, hoặc gọi sai tên hàm.	Bổ sung #include <stdio.h> #include <conio.h>	

CÁC CẢNH BÁO				
1	Possibly incorrect assignment	Dùng ký hiệu trong phép toán quan hệ.	Dùng ký hiệu trong phép so sánh	Sai : <code>if(n%2=0)</code> <code>printf("n chan);</code> Sửa thành: <code>if(n%2==0)</code> <code>printf("n chan");</code>
2	Code has no effect	Dùng ký hiệu phép toán quan hệ cho phép gán.	Dùng ký hiệu phép toán số học.	

f. Debug

Mặc dù chương trình không còn lỗi nhưng khi chạy chương trình vẫn ra kết quả sai, những lỗi đó có thể là:

- Dùng chấm phẩy sau: if, else, for, while, ... mà chưa thực hiện lệnh.
- Định dạng nhập xuất sai hay khai báo sai kiểu dữ liệu.
- Chia cho 0.
- Không có điều kiện dừng (điều kiện dừng sai).
- Phân tích thuật toán thiếu (chưa vét hết các trường hợp) hoặc sai.

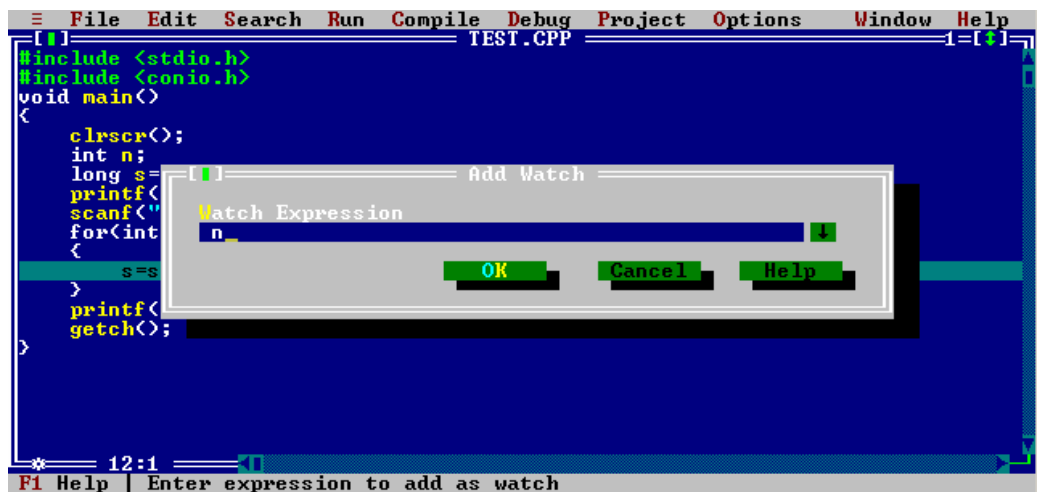
Các thao tác debug:

- Nhấn F7 hoặc F8 để chạy từng bước (nếu không có lỗi khi biên dịch)

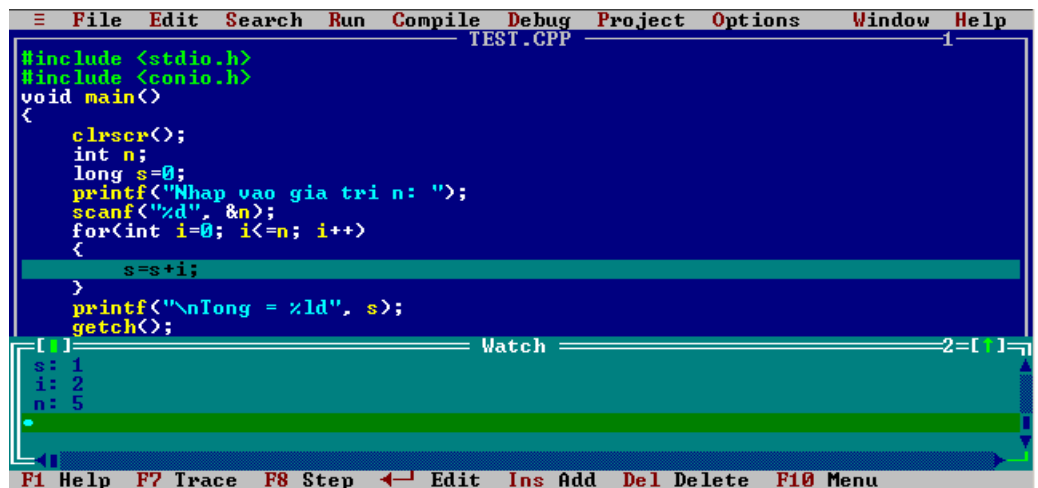
```

#include <stdio.h>
#include <conio.h>
void main()
{
    clrscr();
    int n;
    long s=0;
    printf("Nhap vao gia tri n: ");
    scanf("%d", &n);
    for(int i=0; i<=n; i++)
    {
        s=s+i;
    }
    printf("\nTong = %ld", s);
    getch();
}
    
```

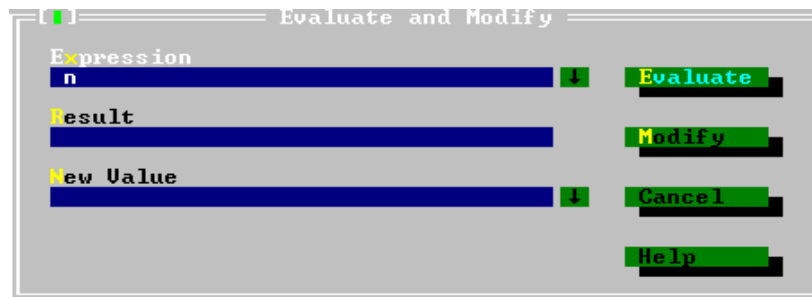
- F7: Đi từng lệnh của hàm con nếu có gọi hàm.
 - F8: không vào chi tiết từng lệnh khi gọi đến hàm con (chỉ đưa ra kết quả của hàm con).
- Quan sát vệt sáng để biết chương trình đang thực hiện đến vị trí lệnh nào.
- Nhấn Ctrl+F7 (hoặc nhấn phím Insert nếu đã có cửa sổ Watch): Nhập vào biến cần theo dõi giá trị các biến khi thực hiện xong lệnh hay hàm nào đó.



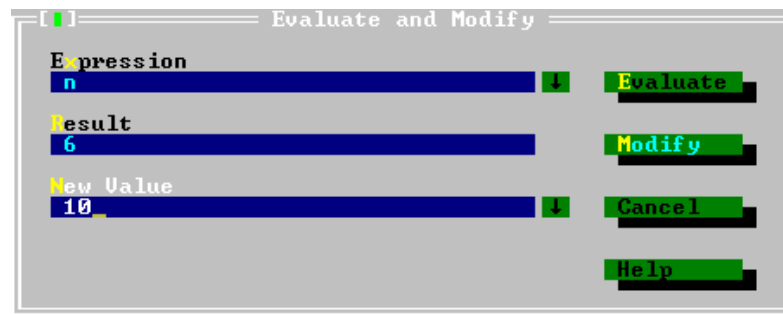
- Có thể xóa biến trên cửa sổ Watch bằng cách chọn biến trên cửa sổ Watch và nhấn phím Delete.
- Nếu không thấy cửa sổ hiển thị giá trị biến (Watch) nhấn Alt+W+W hoặc vào menu Window chọn Watch.



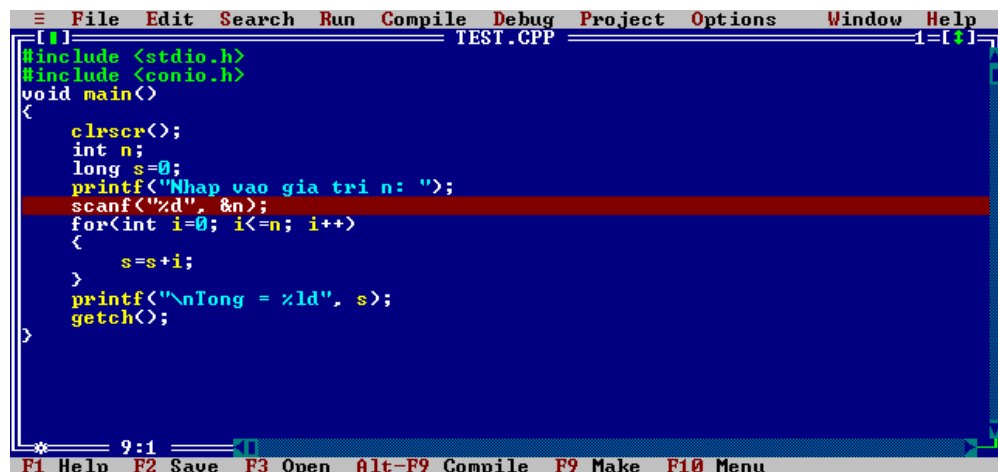
- Nếu muốn bỏ qua một đoạn nào đó (tức không cần kiểm tra đoạn đó) thì nhấn F4 để chương trình thực thi tới vị trí dòng của dấu nhảy rồi dừng lại đó (dấu nhảy phải tại vị trí những dòng phía sau của vệt sáng, nhấn F6 để chuyển qua lại các cửa sổ).
- Muốn thay đổi giá trị của biến ta dùng phím Ctrl+F4 để hiển thị cửa sổ.



- Nhập vào tên biến ở ô Expression, chọn nút Evaluate (hoặc nhấn Enter), ô Result sẽ hiển thị kết quả tại thời điểm đó, sau đó nhập giá trị mới cho biến tại ô New Value → Enter (dùng phím tab để di chuyển vị trí chọn).



- Ngoài ra có thể đánh dấu để chương trình thực thi đến vị trí đánh dấu (khi chưa chạy từng bước) dùng phím F8 để đánh dấu ngay vị trí dấu nhảy. Vị trí đánh dấu sẽ có vệt sáng màu đỏ.



- Có thể đánh dấu nhiều vị trí khác nhau. Nhấn **Ctrl+F9** để chương trình thực thi đến vị trí đánh dấu theo thứ tự từ trên xuống dưới, đồng thời cũng có thể dùng phím **F7** hoặc **F8** giống như trên để chạy từng bước.
- Ngoài ra, có thể dùng phím **ALT+F5** để xem kết quả xuất trong quá trình debug (để kiểm tra nhập xuất).
- Trong quá trình chạy từng bước có thể kết thúc bằng cách nhấn **Ctrl+F2**.

g. Các thao tác liên quan đến cửa sổ Watch

- Di chuyển cửa sổ Watch: Chọn cửa sổ Watch, nhấn **Ctrl+F5**. Sau đó dùng phím mũi tên để di chuyển cửa sổ tới vị trí mới. Nhấn phím Enter.
- Thay đổi kích thước cửa sổ Watch (khi đang chọn bằng **Ctrl+F5** trên cửa sổ Watch) nhấn **Shift** + phím mũi tên rồi nhấn phím Enter.

TÀI LIỆU THAM KHẢO



1. **PHẠM VĂN ÁT**: “Kỹ thuật lập trình C: cơ sở và nâng cao”. Nhà Xuất Bản Khoa Học Kỹ Thuật – 1996.
2. **LÊ HOÀI BẮC – LÊ HOÀNG THÁI – NGUYỄN TẤN TRẦN MINH KHANG – NGUYỄN PHƯƠNG THẢO**: “Giáo trình ngôn ngữ C”. Nhà Xuất Bản Đại Học Quốc Gia Tp. Hồ Chí Minh – 2003.
3. **NGUYỄN TẤN TRẦN MINH KHANG**: “Bài tập Kỹ thuật lập trình – Tập 1”. Nhà Xuất Bản Đại Học Quốc Gia Tp. Hồ Chí Minh – 2004.
4. **NGUYỄN ĐÌNH TÊ – HOÀNG ĐỨC HẢI**: “Giáo trình lý thuyết & Bài tập ngôn ngữ C”. Nhà Xuất Bản Mũi Cà Mau.
5. **HUỲNH TẤN DŨNG – HOÀNG ĐỨC HẢI**: “Bài tập ngôn ngữ C từ A đến Z”. Nhà Xuất Bản Lao Động – Xã Hội.
6. **NGUYỄN THANH SƠN**: “Tập bài giảng Kỹ thuật lập trình” – 2004.
7. **TRẦN MINH THÁI**: “Tập bài giảng Kỹ thuật lập trình” – 2005.
8. **SANFORD LEESTMA LARRY NYHOFF**: “Pascal Programming and Solving”. Macmillan Publishing Company – 1990.
9. **JOHN R. HUBBARD**: “455 Bài tập cấu trúc dữ liệu cài đặt bằng C++”. Bản dịch của Minh Trung, Gia Việt – Nhà Xuất Bản Thống Kê.

MỤC LỤC

LỜI MỞ ĐẦU	1
LỊCH TRÌNH THỰC HÀNH.....	2
CHƯƠNG 1 LƯU ĐỒ THUẬT TOÁN (FLOWCHART)	3
I. TÓM TẮT LÝ THUYẾT	3
<i>I.1. Khái niệm.....</i>	3
<i>I.2. Phương pháp duyệt</i>	3
<i>I.3. Các ký hiệu</i>	3
<i>I.4. Các cấu trúc điều khiển cơ bản</i>	4
<i>a. Cấu trúc tuần tự</i>	<i>4</i>
<i>b. Cấu trúc lựa chọn.....</i>	<i>5</i>
<i>c. Cấu trúc lặp.....</i>	<i>6</i>
<i>d. Các ví dụ.....</i>	<i>8</i>
II. BÀI TẬP	11
<i>II.1. Bài tập cơ bản</i>	11
<i>II.2. Bài tập luyện tập và nâng cao.....</i>	12
III. KẾT LUẬN.....	12
CHƯƠNG 2 CẤU TRÚC ĐIỀU KHIỂN.....	13
I. TÓM TẮT LÝ THUYẾT	13
<i>I.1. Các ký hiệu</i>	13
<i>I.2. Các kiểu dữ liệu cơ bản trong C.....</i>	13
<i>I.3. Bảng ký hiệu các phép toán.....</i>	14
<i>I.4. Các hàm cơ bản.....</i>	15
<i>I.5. Cấu trúc rẽ nhánh</i>	15
<i>a. Cấu trúc if.....</i>	<i>15</i>
<i>b. Cấu trúc if ... else.....</i>	<i>16</i>
<i>I.6. Cấu trúc lựa chọn switch</i>	16
<i>I.7. Cấu trúc lặp</i>	18
<i>a. for</i>	<i>18</i>
<i>b. while</i>	<i>19</i>

c. <i>do ... while</i>	20
I.8. <i>break và continue</i>	20
a. <i>break</i>	20
b. <i>continue</i>	21
II. BÀI TẬP	21
II.1. <i>Phương pháp chạy tay từng bước để tìm kết quả chương trình</i>	21
II.2. <i>Bài tập cơ bản</i>.....	23
a. <i>Cấu trúc if/ if..else và switch</i>	23
b. <i>Cấu trúc lặp</i>	25
II.3. <i>Bài tập luyện tập và nâng cao</i>.....	29
III. KẾT LUẬN.....	30
CHƯƠNG 3 HÀM CON.....	31
I. TÓM TẮT LÝ THUYẾT	31
I.1. <i>Khái niệm</i>.....	31
I.2. <i>Ví dụ</i>.....	31
I.3. <i>Cấu trúc một chương trình C</i>	33
a. <i>Khởi khai báo</i>	33
b. <i>Hàm chính (main())</i>	33
c. <i>Các hàm con</i>	33
d. <i>Nguyên mẫu hàm</i>	33
I.4. <i>Cách xây dựng một hàm con</i>	34
a. <i>Kiểu dữ liệu của hàm</i>	34
b. <i>Tham số</i>	34
c. <i>Tên hàm</i>	35
d. <i>Ví dụ</i>	35
II. BÀI TẬP	37
II.1. <i>Bài tập cơ bản</i>.....	37
II.2. <i>Bài tập luyện tập và nâng cao</i>.....	39
III. KẾT LUẬN.....	39
CHƯƠNG 4 MẢNG MỘT CHIỀU	41
I. TÓM TẮT LÝ THUYẾT	41
I.1. <i>Khái niệm</i>.....	41
I.2. <i>Khai báo mảng</i>.....	41

I.3.	<i>Truy xuất phần tử của mảng</i>	42
II.	BÀI TẬP	43
II.1.	<i>Một số kỹ thuật cơ bản</i>	43
a.	<i>Kỹ thuật đặt cờ hiệu</i>	43
b.	<i>Kỹ thuật đặt lính canh</i>	44
II.2.	<i>Bài tập cơ bản</i>	45
a.	<i>Nhập xuất mảng một chiều</i>	45
b.	<i>Tìm kiếm trên mảng một chiều</i>	46
c.	<i>Đếm – Tần suất</i>	47
d.	<i>Tính tổng – Trung bình có điều kiện</i>	48
e.	<i>Sắp xếp</i>	49
f.	<i>Xoá</i>	50
g.	<i>Chèn</i>	50
h.	<i>Tách / ghép mảng</i>	51
II.3.	<i>Bài tập luyện tập và nâng cao</i>	53
III.	KẾT LUẬN	56
CHƯƠNG 5	CHUỖI KÝ TỰ	57
I.	TÓM TẮT LÝ THUYẾT	57
I.1.	<i>Khái niệm</i>	57
I.2.	<i>Khai báo chuỗi</i>	57
I.3.	<i>Các thao tác trên chuỗi</i>	57
a.	<i>Nhập chuỗi</i>	57
b.	<i>Xuất chuỗi</i>	58
c.	<i>Các hàm thư viện (string.h)</i>	58
d.	<i>Ví dụ</i>	60
II.	BÀI TẬP	60
II.1.	<i>Bài tập cơ bản</i>	60
II.2.	<i>Bài tập luyện tập và nâng cao</i>	62
III.	KẾT LUẬN	63
CHƯƠNG 6	MẢNG HAI CHIỀU	64
I.	TÓM TẮT LÝ THUYẾT	64
I.1.	<i>Khái niệm</i>	64
I.2.	<i>Khai báo mảng</i>	64

I.3.	<i>Truy xuất phần tử của mảng</i>	64
I.4.	<i>Ma trận vuông và các khái niệm liên quan</i>	65
a.	<i>Khái niệm</i>	65
b.	<i>Tính chất của ma trận vuông</i>	65
II.	BÀI TẬP	66
II.1.	<i>Một số kỹ thuật cơ bản</i>	67
II.2.	<i>Bài tập cơ bản</i>	69
a.	<i>Bài tập nhập xuất</i>	69
b.	<i>Bài tập tính tổng</i>	69
c.	<i>Bài tập tìm kiếm</i>	70
d.	<i>Bài tập đếm</i>	70
e.	<i>Bài tập sắp xếp</i>	71
f.	<i>Bài tập Thêm – Xoá – Thay thế</i>	72
II.3.	<i>Bài tập luyện tập và nâng cao</i>	73
III.	KẾT LUẬN	77
CHƯƠNG 7 KIỂU DỮ LIỆU CÓ CẤU TRÚC		78
I.	TÓM TẮT LÝ THUYẾT	78
I.1.	<i>Khái niệm</i>	78
I.2.	<i>Định nghĩa kiểu dữ liệu</i>	78
I.3.	<i>Khai báo</i>	79
I.4.	<i>Truy xuất</i>	80
I.5.	<i>Ví dụ minh họa</i>	81
I.6.	<i>Mảng cấu trúc</i>	82
I.7.	<i>Nguyên tắc viết chương trình có mảng cấu trúc</i>	82
II.	BÀI TẬP	91
II.1.	<i>Bài tập cơ bản</i>	91
II.2.	<i>Bài Tập Luyện Tập</i>	92
III.	KẾT LUẬN	96
CHƯƠNG 8 TẬP TIN		97
I.	TÓM TẮT LÝ THUYẾT	97
I.1.	<i>Khái niệm</i>	97
I.2.	<i>Thao tác với tập tin</i>	97
a.	<i>Khai báo</i>	97

<i>b.</i>	<i>Mở tập tin</i>	97
<i>c.</i>	<i>Các hàm đọc ghi nội dung tập tin</i>	98
<i>d.</i>	<i>Đóng tập tin</i>	99
<i>e.</i>	<i>Các thao tác khác trên tập tin</i>	99
<i>f.</i>	<i>Ví dụ minh hoạ</i>	99
I.3.	<i>Các ví dụ minh hoạ</i>	100
<i>a.</i>	<i>Tập tin văn bản</i>	100
<i>b.</i>	<i>Tập tin nhị phân</i>	102
II.	BÀI TẬP	103
II.1.	<i>Bài tập cơ bản</i>	103
II.2.	<i>Bài tập luyện tập và nâng cao</i>	105
III.	KẾT LUẬN	108
CHƯƠNG 9	ĐỆ QUI	109
I.	TÓM TẮT LÝ THUYẾT	109
I.1.	<i>Khái niệm</i>	109
I.2.	<i>Phân loại đệ qui</i>	109
<i>a.</i>	<i>Đệ qui tuyến tính</i>	109
<i>b.</i>	<i>Đệ qui nhị phân</i>	110
<i>c.</i>	<i>Đệ qui phi tuyến</i>	112
<i>d.</i>	<i>Đệ qui hỗ tương</i>	113
I.3.	<i>Tìm hiểu cách hoạt động của hàm đệ qui</i>	114
I.4.	<i>Ví dụ</i>	115
II.	BÀI TẬP	116
II.1.	<i>Bài tập cơ bản</i>	116
II.2.	<i>Bài tập luyện tập và nâng cao</i>	117
III.	KẾT LUẬN	117
CHƯƠNG 10	LẬP TRÌNH THEO PHƯƠNG PHÁP PROJECT	118
I.	MỤC TIÊU	118
II.	PHƯƠNG PHÁP	118
II.1.	<i>Tạo một project mới</i>	118
II.2.	<i>Mở project có sẵn</i>	118
II.3.	<i>Một số lưu ý</i>	119
II.4.	<i>Ví dụ minh hoạ</i>	119

III. BÀI TẬP	123
PHỤ LỤC 1 ĐỀ THI MẪU	124
PHỤ LỤC 2 HƯỚNG DẪN VIẾT CHƯƠNG TRÌNH TRÊN MÔI TRƯỜNG BORLAND C++ 3.1 (BC31)	133
I. CÀI ĐẶT BC3.1.....	133
II. CÁC BƯỚC VIẾT CHƯƠNG TRÌNH	138
<i>a. Chuẩn bị viết chương trình</i>	<i>138</i>
<i>b. Các phím chức năng chính.....</i>	<i>138</i>
<i>c. Viết chương trình.....</i>	<i>139</i>
<i>d. Biên dịch và sửa lỗi.....</i>	<i>139</i>
<i>e. Một số lỗi thường gặp</i>	<i>140</i>
<i>f. Debug</i>	<i>143</i>
<i>g. Các thao tác liên quan đến cửa sổ Watch.....</i>	<i>145</i>
TÀI LIỆU THAM KHẢO	146
MỤC LỤC	i

Ngôn Ngữ Lập Trình Căn Bản

KẾT BẠN GIÚP NHAU TIẾN BỘ TRONG HỌC TẬP

Xin chào bạn!

Cám ơn bạn đã quan tâm đến tài liệu, tải xuống và sử dụng nó. Mình cũng chỉ là người đang tìm hiểu về cái ngôn ngữ C này. Mình đã tìm các ebook, tài liệu có liên quan, rồi upload lên mạng chia sẻ...hy vọng phần nào giúp ích cho bạn mới làm quen với lập trình. Rất mong được làm quen với các bạn gần xa yêu thích môn lập trình để mình học kinh nghiệm.

Liên lạc:

Mình ở TP.HCM

Yahoo: *laptrinh_123*

Email : *dung.softwares@gmail.com*

Hướng lâu dài của mình là theo ngành "Lập trình ứng dụng web". Theo ngành này cần phải biết khá nhiều: Lập trình (PHP,MySQL...) - Đồ họa (Flash, Photoshop...), chưa kể những thứ tổng hợp linh tinh khác có liên quan như là các ngôn ngữ lập trình khác... Nói chung là có nhiều thứ để học hỏi lắm.

Mình luôn sẵn sàng đón nhận sự giúp đỡ của các bạn có kinh nghiệm chuyên môn, giao lưu với các bạn có cùng sở thích như mình, trao đổi các tài liệu có liên quan đến các lĩnh vực trên...

Không biết bạn nghĩ sao, chứ mình thấy tự học mà chỉ theo cái cách lên mạng tải ebook về rồi in ra và ngâm kiu thì... chậm tiến lắm. Mình nghĩ phải có sự học hỏi từ những người có K/Nghiệm khác thì mới tốt. Bạn bè sẽ bổ sung cho nhau những lỗ hổng kiến thức... có tác dụng tích cực hơn. "HỌC THẦY KO BẰNG HỌC BẠN" mà lại...

TP.HCM, ngày 20/07/2009

PHẦN 2

GIỚI THIỆU VỀ MỘT NGÔN NGỮ LẬP TRÌNH - NGÔN NGỮ LẬP TRÌNH C

Chương 1

GIỚI THIỆU VỀ NGÔN NGỮ C & MÔI TRƯỜNG TURBO C 3.0

Học xong chương này, sinh viên sẽ nắm được các vấn đề sau:

- Tổng quan về ngôn ngữ lập trình C.
- Môi trường làm việc và cách sử dụng Turbo C 3.0.

I. TỔNG QUAN VỀ NGÔN NGỮ LẬP TRÌNH C

C là ngôn ngữ lập trình cấp cao, được sử dụng rất phổ biến để lập trình hệ thống cùng với Assembler và phát triển các ứng dụng.

Vào những năm cuối thập kỷ 60 đầu thập kỷ 70 của thế kỷ XX, Dennish Ritchie (làm việc tại phòng thí nghiệm Bell) đã phát triển ngôn ngữ lập trình C dựa trên ngôn ngữ BCPL (do Martin Richards đưa ra vào năm 1967) và ngôn ngữ B (do Ken Thompson phát triển từ ngôn ngữ BCPL vào năm 1970 khi viết hệ điều hành UNIX đầu tiên trên máy PDP-7) và được cài đặt lần đầu tiên trên hệ điều hành UNIX của máy DEC PDP-11.

Năm 1978, Dennish Ritchie và B.W Kernighan đã cho xuất bản quyển “Ngôn ngữ lập trình C” và được phổ biến rộng rãi đến nay.

Lúc ban đầu, C được thiết kế nhằm lập trình trong môi trường của hệ điều hành Unix nhằm mục đích hỗ trợ cho các công việc lập trình phức tạp. Nhưng về sau, với những nhu cầu phát triển ngày một tăng của công việc lập trình, C đã vượt qua khuôn khổ của phòng thí nghiệm Bell và nhanh chóng hội nhập vào thế giới lập trình để rồi các công ty lập trình sử dụng một cách rộng rãi. Sau đó, các công ty sản xuất phần mềm lần lượt đưa ra các phiên bản hỗ trợ cho việc lập trình bằng ngôn ngữ C và chuẩn ANSI C cũng được khai sinh từ đó.

Ngôn ngữ lập trình C là một ngôn ngữ lập trình hệ thống rất mạnh và rất “mềm dẻo”, có một thư viện gồm rất nhiều các hàm (function) đã được tạo sẵn. Người lập trình có thể tận dụng các hàm này để giải quyết các bài toán mà không cần phải tạo mới. Hơn thế nữa, ngôn ngữ C hỗ trợ rất nhiều phép toán nên phù hợp cho việc giải quyết các bài toán kỹ thuật có nhiều công thức phức tạp. Ngoài ra, C cũng cho phép người lập trình tự định nghĩa thêm các kiểu dữ liệu trừu tượng khác. Tuy nhiên, điều mà người mới vừa học lập trình C thường gặp “rắc rối” là “hơi khó hiểu” do sự “mềm dẻo” của C. Dù vậy, C được phổ biến khá rộng rãi và đã trở thành một công cụ lập

trình khá mạnh, được sử dụng như là một ngôn ngữ lập trình chủ yếu trong việc xây dựng những phần mềm hiện nay.

Ngôn ngữ C có những đặc điểm cơ bản sau:

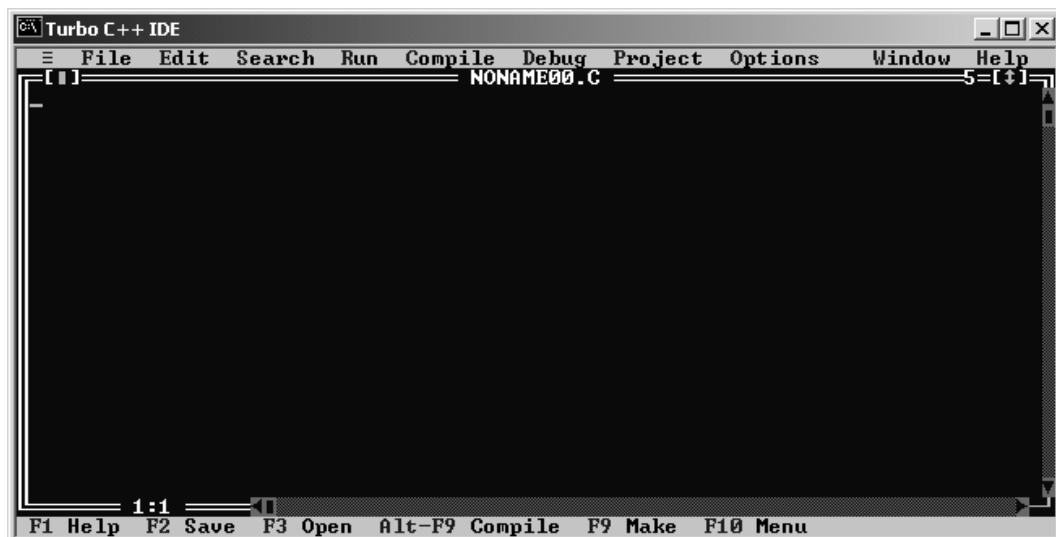
- *Tính cô đọng (compact)*: C chỉ có 32 từ khóa chuẩn và 40 toán tử chuẩn, nhưng hầu hết đều được biểu diễn bằng những chuỗi ký tự ngắn gọn.
- *Tính cấu trúc (structured)*: C có một tập hợp những chỉ thị của lập trình như cấu trúc lựa chọn, lặp... Từ đó các chương trình viết bằng C được tổ chức rõ ràng, dễ hiểu.
- *Tính tương thích (compatible)*: C có bộ tiền xử lý và một thư viện chuẩn vô cùng phong phú nên khi chuyển từ máy tính này sang máy tính khác các chương trình viết bằng C vẫn hoàn toàn tương thích.
- *Tính linh động (flexible)*: C là một ngôn ngữ rất uyển chuyển và cú pháp, chấp nhận nhiều cách thể hiện, có thể thu gọn kích thước của các mã lệnh làm chương trình chạy nhanh hơn.
- *Biên dịch (compile)*: C cho phép biên dịch nhiều tập tin chương trình riêng rẽ thành các tập tin đối tượng (object) và liên kết (link) các đối tượng đó lại với nhau thành một chương trình có thể thực thi được (executable) thống nhất.

II. MÔI TRƯỜNG LẬP TRÌNH TURBO C

Turbo C là môi trường hỗ trợ lập trình C do hãng Borland cung cấp. Môi trường này cung cấp các chức năng như: soạn thảo chương trình, dịch, thực thi chương trình... Phiên bản được sử dụng ở đây là Turbo C 3.0.

II.1. Gọi Turbo C

Chạy Turbo C cũng giống như chạy các chương trình khác trong môi trường DOS hay Windows, màn hình sẽ xuất hiện menu của Turbo C có dạng như sau:



Dòng trên cùng gọi là thanh menu (menu bar). Mỗi mục trên thanh menu lại có thể có nhiều mục con nằm trong một menu kéo xuống.

Dòng dưới cùng ghi chức năng của một số phím đặc biệt. Chẳng hạn khi gõ phím F1 thì ta có được một hệ thống trợ giúp mà ta có thể tham khảo nhiều thông tin bổ ích.

Muốn vào thanh menu ngang ta gõ phím F10. Sau đó dùng các phím mũi tên qua trái hoặc phải để di chuyển vùng sáng tới mục cần chọn rồi gõ phím Enter. Trong menu kéo xuống ta lại dùng các phím mũi tên lên xuống để di chuyển vùng sáng tới mục cần chọn rồi gõ Enter.

Ta cũng có thể chọn một mục trên thanh menu bằng cách giữ phím Alt và gõ vào một ký tự đại diện của mục đó (ký tự có màu sắc khác với các ký tự khác). Chẳng hạn để chọn mục File ta gõ Alt-F (F là ký tự đại diện của File)

II.2. Soạn thảo chương trình mới

Muốn soạn thảo một chương trình mới ta chọn mục New trong menu File (File ->New)

Trên màn hình sẽ xuất hiện một vùng trống để cho ta soạn thảo nội dung của chương trình. Trong quá trình soạn thảo chương trình ta có thể sử dụng các phím sau:

Các phím xem thông tin trợ giúp:

- F1: Xem toàn bộ thông tin trong phần trợ giúp.
- Ctrl-F1: Trợ giúp theo ngữ cảnh (tức là khi con trỏ đang ở trong một từ nào đó, chẳng hạn int mà bạn gõ phím Ctrl-F1 thì bạn sẽ có được các thông tin về kiểu dữ liệu int)

Các phím di chuyển con trỏ trong vùng soạn thảo chương trình:

Phím	Ý nghĩa	Phím tắt (tổ hợp phím)
Enter	Đưa con trỏ xuống dòng	
Mũi tên đi lên	Đưa con trỏ lên hàng trước	Ctrl-E
Mũi tên đi xuống	Đưa con trỏ xuống hàng sau	Ctrl-X
Mũi tên sang trái	Đưa con trỏ sang trái một ký tự	Ctrl-S
Mũi tên sang phải	Đưa con trỏ sang phải một ký tự	Ctrl-D
End	Đưa con trỏ đến cuối dòng	
Home	Đưa con trỏ đến đầu dòng	
PgUp	Đưa con trỏ lên trang trước	Ctrl-R
PgDn	Đưa con trỏ xuống trang sau	Ctrl-C
	Đưa con trỏ sang từ bên trái	Ctrl-A
	Đưa con trỏ sang từ bên phải	Ctrl-F

Các phím xóa ký tự/ dòng:

Phím	Ý nghĩa	Phím tắt
Delete	Xóa ký tự tại vị trí con trỏ	Ctrl-G
BackSpace	Di chuyển sang trái đồng thời xóa ký tự đứng trước con trỏ	Ctrl-H
	Xóa một dòng chứa con trỏ	Ctrl-Y
	Xóa từ vị trí con trỏ đến cuối dòng	Ctrl-Q-Y
	Xóa ký tự bên phải con trỏ	Ctrl-T

Các phím chèn ký tự/ dòng:

Insert	Thay đổi viết xen hay viết chồng
Ctrl-N	Xen một dòng trống vào trước vị trí con trỏ

Sử dụng khối :

Khối là một đoạn văn bản chương trình hình chữ nhật được xác định bởi đầu khối là góc trên bên trái và cuối khối là góc dưới bên phải của hình chữ nhật. Khi một khối đã được xác định (trên màn hình khối có màu sắc khác chỗ bình thường) thì ta có

thể chép khối, di chuyển khối, xoá khối... Sử dụng khối cho phép chúng ta soạn thảo chương trình một cách nhanh chóng. sau đây là các thao tác trên khối:

Phím tắt	Ý nghĩa
Ctrl-K-B	Đánh dấu đầu khối
Ctrl-K-K	Đánh dấu cuối khối
Ctrl-K-C	Chép khối vào sau vị trí con trỏ
Ctrl-K-V	Chuyển khối tới sau vị trí con trỏ
Ctrl-K-Y	Xoá khối
Ctrl-K-W	Ghi khối vào đĩa như một tập tin
Ctrl-K-R	Đọc khối (tập tin) từ đĩa vào sau vị trí con trỏ
Ctrl-K-H	Tắt/mở khối
Ctrl-K-T	Đánh dấu từ chứa con trỏ
Ctrl-K-P	In một khối

Các phím, phím tắt thực hiện các thao tác khác:

Phím	Ý nghĩa	Phím tắt
F10	Kích hoạt menu chính	Ctrl-K-D, Ctrl-K-Q
F2	Lưu chương trình đang soạn vào đĩa	Ctrl-K-S
F3	Tạo tập tin mới	
Tab	Di chuyển con trỏ một khoảng đồng thời đẩy dòng văn bản	Ctrl-I
ESC	Hủy bỏ thao tác lệnh	Ctrl-U
	Đóng tập tin hiện tại	Alt-F3
	Hiện hộp thoại tìm kiếm	Ctrl-Q-F
	Hiện hộp thoại tìm kiếm và thay thế	Ctrl-Q-A
	Tìm kiếm tiếp tục	Ctrl-L

Ví dụ: Bạn hãy gõ đoạn chương trình sau:

```
#include <stdio.h>
#include <conio.h>
int main ()
{
    char ten[50];
    printf("Xin cho biet ten cua ban !");
    scanf("%s",ten);
    printf("Xin chao ban %s",ten);
    getch();
    return 0;
}
```

II.3. Ghi chương trình đang soạn thảo vào đĩa

Sử dụng File/Save hoặc gõ phím F2. Có hai trường hợp xảy ra:

- Nếu chương trình chưa được ghi lần nào thì một hội thoại sẽ xuất hiện cho phép bạn xác định tên tập tin (FileName). Tên tập tin phải tuân thủ quy cách đặt tên của DOS và không cần có phần mở rộng (sẽ tự động có phần mở rộng là .C hoặc .CPP sẽ nói thêm trong phần Option). Sau đó gõ phím Enter.

- Nếu chương trình đã được ghi một lần rồi thì nó sẽ ghi những thay đổi bổ sung lên tập tin chương trình cũ.

Chú ý: Để đề phòng mất điện trong khi soạn thảo chương trình thỉnh thoảng bạn nên gõ phím F2.

Quy tắc đặt tên tập tin của DOS: Tên của tập tin gồm 2 phần: Phần tên và phần mở rộng.

○ Phần tên của tập tin phải bắt đầu là 1 ký tự từ a..z (không phân biệt hoa thường), theo sau có thể là các ký tự từ a..z, các ký số từ 0..9 hay dấu gạch dưới (_), phần này dài tối đa là 8 ký tự.

○ Phần mở rộng: phần này dài tối đa 3 ký tự.

Ví dụ: Ghi chương trình vừa soạn thảo trên lên đĩa với tên là CHAO.C

II.4. Thực hiện chương trình

Để thực hiện chương trình hãy dùng Ctrl-F9 (giữ phím Ctrl và gõ phím F9).

Ví dụ: Thực hiện chương trình vừa soạn thảo xong và quan sát trên màn hình để thấy kết quả của việc thực thi chương trình sau đó gõ phím bất kỳ để trở lại với Turbo.

II.5. Mở một chương trình đã có trên đĩa

Với một chương trình đã có trên đĩa, ta có thể mở nó ra để thực hiện hoặc sửa chữa bổ sung. Để mở một chương trình ta dùng File/Open hoặc gõ phím F3. Sau đó gõ tên tập tin vào hộp File Name hoặc lựa chọn tập tin trong danh sách các tập tin rồi gõ Enter.

Ví dụ: Mở tập tin CHAO.C sau đó bổ sung để có chương trình mới như sau:

```
#include <stdio.h>
#include<conio.h>
int main ()
{
    char ten[50];
    printf("Xin cho biet ten cua ban !");
    scanf("%s",ten);
    printf("Xin chao ban %s\n ",ten);
    printf("Chao mung ban den voi Ngon ngu lap trinh C");
    getch();
    return 0;
}
```

Ghi lại chương trình này (F2) và cho thực hiện (Ctrl-F9). Hãy so sánh xem có gì khác trước?

II.6. Thoát khỏi Turbo C và trở về DOS (hay Windows)

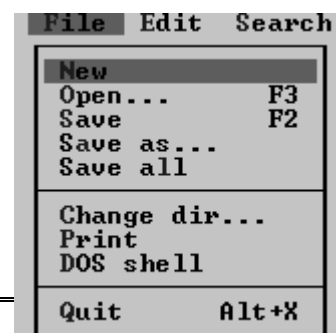
Dùng File/Exit hoặc Alt-X.

II.7. Sử dụng một số lệnh trên thanh menu

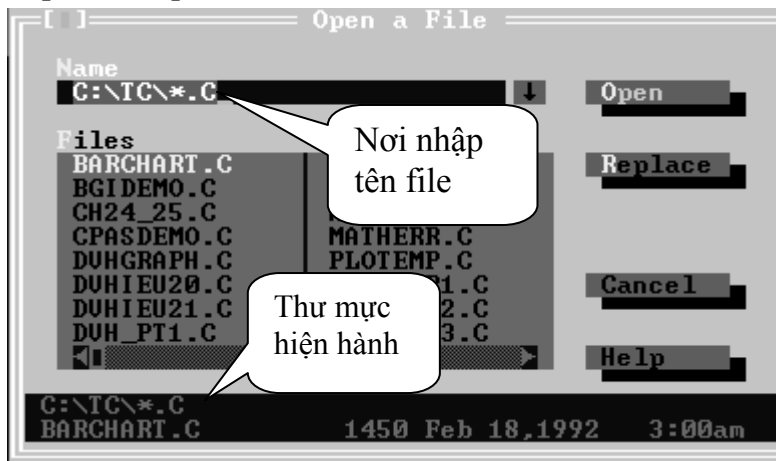
II.7.1. Các lệnh trên menu File (Alt -F)

- Lệnh New : Dùng để tạo mới một chương trình. Tên ngầm định của chương trình là NONAMEXX.C (XX là 2 số từ 00 đến 99).

- Lệnh Open : Dùng để mở một chương trình đã có sẵn trên đĩa để sửa chữa, bổ sung hoặc để thực hiện chương trình



đó. Khi tập tin được mở thì văn bản chương trình được trình bày trong vùng soạn thảo; hộp thoại Open như sau:

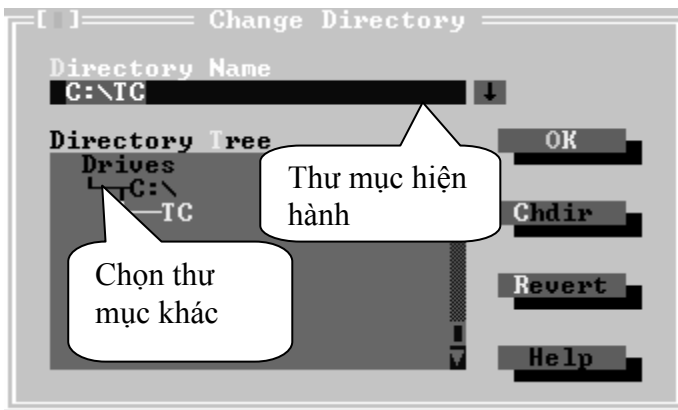


Trong trường hợp ta nhập vào tên tập tin chưa tồn tại thì chương trình được tạo mới và sau này khi ta lưu trữ, chương trình được lưu với tên đó.

- Lệnh Save : Dùng để lưu chương trình đang soạn thảo vào đĩa.
- Lệnh Save as... : Dùng để lưu chương trình đang soạn thảo với tên khác, hộp thoại lưu tập tin đang soạn với tên khác như sau:



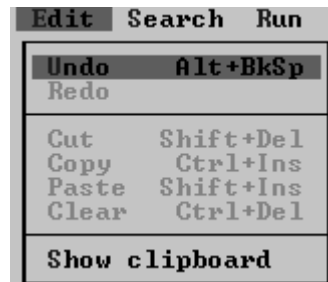
- Lệnh Save All: Trong lúc làm việc với Turbo C, ta có thể mở một lúc nhiều chương trình để sửa chữa, bổ sung. Lệnh Save All dùng để lưu lại mọi thay đổi trên tất cả các chương trình đang mở ấy..
- Lệnh Change Dir ... : Dùng để đổi thư mục hiện hành



- Lệnh Print : Dùng để in chương trình đang soạn thảo ra máy in.
- Lệnh Printer Setup ...: Dùng để thiết đặt một số thông số cho máy in.
- Lệnh Dos Shell : Dùng để thoát tạm thời về Dos, để trở lại Turbo C ta đánh EXIT.
- Lệnh Exit : Dùng để thoát khỏi C.

II.7.2. Các lệnh trên menu Edit (Alt -E)

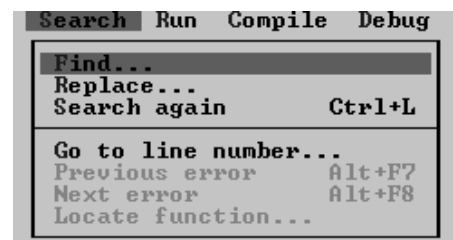
- Lệnh Undo : Dùng để hủy bỏ thao tác soạn thảo cuối cùng trên cửa sổ soạn thảo.
- Lệnh Redo : Dùng để phục hồi lại thao tác đã bị Undo cuối cùng.
- Lệnh Cut : Dùng để xóa một phần văn bản đã được đánh dấu khối, phần dữ liệu bị xóa sẽ được lưu vào một vùng nhớ *đặc biệt* gọi là Clipboard.



- Lệnh Copy : Dùng để chép phần chương trình đã được đánh dấu khối vào Clipboard.
- Lệnh Paste : Dùng để dán phần chương trình đang được lưu trong Clipboard vào cửa sổ đang soạn thảo, bắt đầu tại vị trí của con trỏ.
- Lệnh Clear : Dùng để xóa phần dữ liệu đã được đánh dấu khối, dữ liệu bị xóa không được lưu vào Clipboard.
- Lệnh Show clipboard : Dùng để hiển thị phần chương trình đang được lưu trong Clipboard trong một cửa sổ mới.

II.7.3. Các lệnh trên menu Search (Alt -S)

- Lệnh Find ...: Dùng để tìm kiếm một cụm từ trong văn bản chương trình. Nếu tìm thấy thì con trỏ sẽ di chuyển đến đoạn văn bản trùng với cụm từ cần tìm; hộp thoại Find như sau:



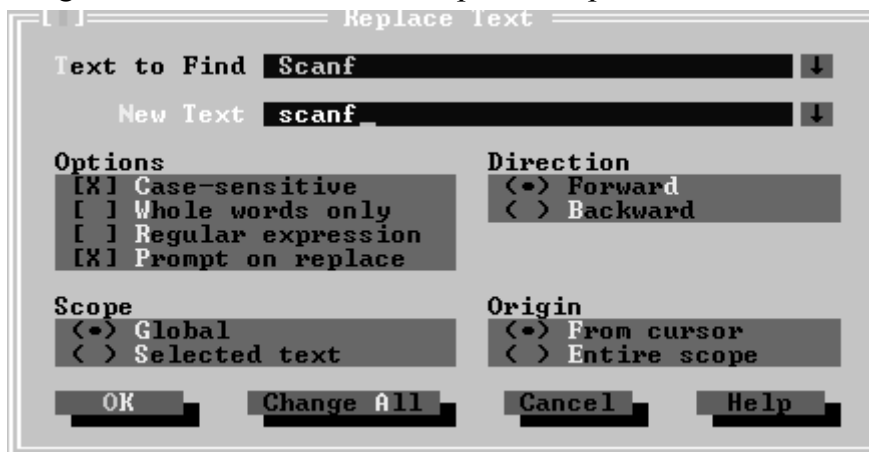


Nhập cụm từ cần tìm vào đây

Ý nghĩa các lựa chọn trong hộp thoại trên như sau:

- Case sensitive : Phân biệt chữ IN HOA với chữ in thường trong khi so sánh cụm từ cần tìm với văn bản chương trình.
- Whole word only: Một đoạn văn bản chương trình trùng với toàn bộ cụm từ cần tìm thì mới được xem là tìm thấy.
- Regular expression: Tìm theo biểu thức
- Global: Tìm trên tất cả tập tin.
- Forward : Tìm đến cuối tập tin.
- Selected text: Chỉ tìm trong khối văn bản đã được đánh dấu.
- Backward: Tìm đến đầu tập tin.
- From cursor : Bắt đầu từ vị trí con nháy.
- Entire scope: Bắt đầu tại vị trí đầu tiên của khối hoặc tập tin.

- Lệnh Replace...: Dùng để tìm kiếm một đoạn văn bản nào đó, và tự động thay bằng một đoạn văn bản khác, hộp thoại replace như sau:

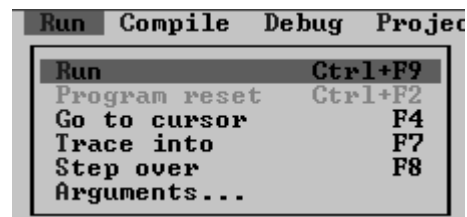


Tìm các cụm từ Scanf và thay thế bằng scanf

- Lệnh Search again : Dùng để thực hiện lại việc tìm kiếm.
- Các lệnh còn lại trên menu Search, các bạn sẽ tìm hiểu thêm khi thực hành trực tiếp trên máy tính.

II.7.4. Các lệnh trên menu Run (Alt -R)

- Lệnh Run : Dùng để thực thi hay "chạy" một chương trình.
- Lệnh Step over : Dùng để "chạy" chương trình từng bước.



- Lệnh Trace into : Dùng để chạy chương trình từng bước. *Khác với lệnh Step over ở chỗ:* Lệnh Step over không cho chúng ta xem từng bước "chạy" trong chương trình con, còn lệnh Trace into cho chúng ta xem từng bước trong chương trình con.

- Các lệnh còn lại, các bạn sẽ tìm hiểu thêm khi thực hành trên máy.

II.7.5. Các lệnh trên menu Compile (Alt C)

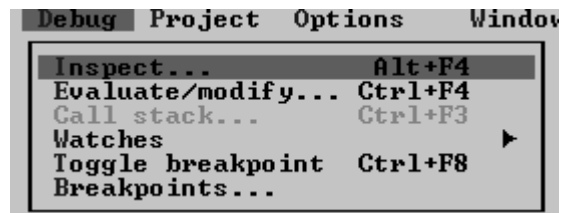
- Lệnh Compile: Biên dịch một chương trình.
- Lệnh Make , Build, ... : Các lệnh này bạn sẽ tìm hiểu thêm khi thực hành trực tiếp trên máy tính.
- Lệnh Information : Dùng để hiển thị các thông tin về chương trình, Mode, môi trường .



II.7.6. Các lệnh trên menu Debug (Alt-D)

Trên menu Debug bao gồm một số lệnh giúp người lập trình "gỡ rối" chương trình . Người lập trình sử dụng chức năng "gỡ rối" khi gặp một số "lỗi" về thuật toán, sử dụng biến nhớ...

- Lệnh Breakpoints: Dùng để đặt "điểm dừng" trong chương trình. Khi chương trình thực thi đến "điểm dừng thì nó sẽ dừng lại" .
- Lệnh Watch : Dùng để mở một cửa sổ hiển thị kết quả trung gian của một biến nhớ nào đó khi chạy chương trình từng bước.
- Lệnh Evaluate/Modify: Bạn sẽ tìm hiểu khi thực hành trực tiếp trên máy.



II.7.7. Các lệnh trên menu Project (Alt- P)

Trên menu Project bao gồm các lệnh liên quan đến dự án như : đóng, mở, thêm , xóa các mục,...

II.7.8. Các lệnh trên menu Option (Alt -O)

Trên menu Option bao gồm các lệnh giúp người lập trình thiết đặt một số tự chọn khi chạy chương trình. Thông thường, người lập trình không cần phải thiết đặt lại các tự chọn.

- Lệnh Compiler ...: Dùng để thiết đặt lại một số thông số khi biên dịch chương trình như hình sau

Phần trình bày dưới đây thuộc về 3 mục: Directories, Enviroment và Save; các phần khác sinh viên tự tìm hiểu.

- Lệnh Directories...: Dùng để đặt lại đường dẫn tìm đến các tập tin cần thiết khi biên dịch chương trình như hình sau:





- Include directory: Thư mục chứa các tập tin mà chúng ta muốn đưa vào chương trình (các tập tin .h trong dòng #include).
- Library directory : Thư mục chứa các tập tin thư viện (các tập tin .Lib)
- Output directory: Thư mục chứa các tập tin “đối tượng “ (có phần mở rộng là .OBJ), tập tin thực thi (.exe) khi biên dịch chương trình.
- Source directory: Thư mục chứa các tập tin “nguồn” (có phần mở rộng là .obj, .lib).

- Lệnh Environment: dùng để thiết lập môi trường làm việc như:

- Reference...: Các tham chiếu.
- Editor: Môi trường soạn thảo gồm: tạo tập tin dự phòng khi có sự chỉnh sửa (create backup file), chế độ viết đè (insert mode), tự động thụt đầu dòng (indent), đổi màu từ khóa (Syntax highlighting)... Đặc biệt, trong phần này là thiết lập phần mở rộng mặc định (Default Extension) của tập tin chương trình là C hay CPP (C Plus Plus: C++).



- Mouse...: Đặt chuột.
- Colors...: Đặt màu.

II.7.9. Các lệnh trên menu Window (Alt- W)

Trên menu Window bao gồm các lệnh thao tác đến cửa sổ như:

- Lệnh Cascade : Dùng để sắp xếp các cửa sổ.
- Lệnh Close all : Dùng để đóng tất cả các cửa sổ.
- Lệnh Zoom: Dùng để phóng to/ thu nhỏ cửa sổ.
- Các lệnh Tile, Refresh display, Size/ Move, Next, Previous, Close, List...: Các bạn sẽ tìm hiểu thêm khi thực hành trực tiếp trên máy tính.

II.7.10. Các lệnh trên menu Help (Alt- H)

Trên menu Help bao gồm các lệnh gọi trợ giúp khi người lập trình cần giúp đỡ một số vấn đề nào đó như: Cú pháp câu lệnh, cách sử dụng các hàm có sẵn...

- Lệnh Contents: Hiện thị toàn bộ nội dung của phần help.

- Lệnh Index : Hiển thị bảng tìm kiếm theo chỉ mục.
- Các lệnh còn lại, bạn sẽ tìm hiểu khi thực hành trên máy.

Chương 2

CÁC THÀNH PHẦN CƠ BẢN CỦA NGÔN NGỮ C

Học xong chương này, sinh viên sẽ nắm được các vấn đề sau:

- Bộ chữ viết trong C.
- Các từ khóa.
- Danh biểu.
- Các kiểu dữ liệu
- Biến và các biểu thức trong C.
- Cấu trúc của một chương trình viết bằng ngôn ngữ lập trình C

I. BỘ CHỮ VIẾT TRONG C

Bộ chữ viết trong ngôn ngữ C bao gồm những ký tự, ký hiệu sau: (*phân biệt chữ in hoa và in thường*):

- 26 chữ cái latin lớn A,B,C...Z
- 26 chữ cái latin nhỏ a,b,c ...z.
- 10 chữ số thập phân 0,1,2...9.
- Các ký hiệu toán học: +, -, *, /, =, <, >, (,)
- Các ký hiệu đặc biệt: ., ;, " ' _ @ # \$! ^ [] { } ...
- Dấu cách hay khoảng trống.

ii. Các từ khoá trong c

Từ khóa là các từ dành riêng (reserved words) của C mà người lập trình có thể sử dụng nó trong chương trình tùy theo ý nghĩa của từng từ. Ta không được dùng từ khóa để đặt cho các tên của riêng mình. Các từ khóa của Turbo C 3.0 bao gồm:

asm	auto	break	case	cdecl	char
class	const	continue	_cs	default	delete
do	double	_ds	else	enum	_es
extern	_export	far	_fastcall	float	for
friend	goto	huge	if	inline	int
interrupt	_loadds	long	near	new	operator
pascal	private	protected	public	register	return
_saveregs	_seg	short	signed	sizeof	_ss
static	struct	switch	template	this	typedef
union	unsigned	virtual	void	volatile	while

iii. Cặp dấu ghi chú thích

Khi viết chương trình đôi lúc ta cần phải có vài lời ghi chú về 1 đoạn chương trình nào đó để dễ nhớ và dễ điều chỉnh sau này; nhất là phần nội dung ghi chú phải

không thuộc về chương trình (khi biên dịch phần này bị bỏ qua). Trong ngôn ngữ lập trình C, nội dung chú thích phải được viết trong cặp dấu /* và */.

Ví dụ :

```
#include <stdio.h>
#include<conio.h>
int main ()
{
    char ten[50]; /* khai bao bien ten kieu char 50 ky tu */
                /*Xuat chuoai ra man hinh*/
    printf("Xin cho biet ten cua ban !");
    scanf("%s",ten); /*Doc vao 1 chuoai la ten cua ban*/
    printf("Xin chao ban %s\n ",ten);
    printf("Chao mung ban den voi Ngon ngu lap trinh C");
    /*Dung chuong trinh, cho go phim*/
    getch();
    return 0;
}
```

IV. CÁC KIỂU DỮ LIỆU SƠ CẤP CHUẨN TRONG C

Các kiểu dữ liệu sơ cấp chuẩn trong C có thể được chia làm 2 dạng : kiểu số nguyên, kiểu số thực.

IV.1. Kiểu số nguyên

Kiểu số nguyên là kiểu dữ liệu dùng để lưu các giá trị nguyên hay còn gọi là kiểu đếm được. Kiểu số nguyên trong C được chia thành các kiểu dữ liệu con, mỗi kiểu có một miền giá trị khác nhau

IV.1.1. Kiểu số nguyên 1 byte (8 bits)

Kiểu số nguyên một byte gồm có 2 kiểu sau:

STT	Kiểu dữ liệu	Miền giá trị (Domain)
1	unsigned char	Từ 0 đến 255 (tương đương 256 ký tự trong bảng mã ASCII)
2	char	Từ -128 đến 127

Kiểu unsigned char: lưu các số nguyên dương từ 0 đến 255.

=> Để khai báo một biến là kiểu ký tự thì ta khai báo biến kiểu unsigned char. Mỗi số trong miền giá trị của kiểu unsigned char tương ứng với một ký tự trong bảng mã ASCII .

Kiểu char: lưu các số nguyên từ -128 đến 127. Kiểu char sử dụng bit trái nhất để làm bit dấu.

=> Nếu gán giá trị > 127 cho biến kiểu char thì giá trị của biến này có thể là số âm (?).

IV.1.2. Kiểu số nguyên 2 bytes (16 bits)

Kiểu số nguyên 2 bytes gồm có 4 kiểu sau:

STT	Kiểu dữ liệu	Miền giá trị (Domain)
1	enum	Từ -32,768 đến 32,767
2	unsigned int	Từ 0 đến 65,535

3	short int	Từ -32,768 đến 32,767
4	int	Từ -32,768 đến 32,767

Kiểu enum, short int, int : Lưu các số nguyên từ -32768 đến 32767. Sử dụng bit bên trái nhất để làm bit dấu.

=> Nếu gán giá trị >32767 cho biến có 1 trong 3 kiểu trên thì giá trị của biến này có thể là số âm.

Kiểu unsigned int: Kiểu unsigned int lưu các số nguyên dương từ 0 đến 65535.

IV.1.3. Kiểu số nguyên 4 byte (32 bits)

Kiểu số nguyên 4 bytes hay còn gọi là số nguyên dài (long) gồm có 2 kiểu sau:

STT	Kiểu dữ liệu	Miền giá trị (Domain)
1	unsigned long	Từ 0 đến 4,294,967,295
2	long	Từ -2,147,483,648 đến 2,147,483,647

Kiểu long : Lưu các số nguyên từ -2147483658 đến 2147483647. Sử dụng bit bên trái nhất để làm bit dấu.

=> Nếu gán giá trị >2147483647 cho biến có kiểu long thì giá trị của biến này có thể là số âm.

Kiểu unsigned long: Kiểu unsigned long lưu các số nguyên dương từ 0 đến 4294967295

IV.2. Kiểu số thực

Kiểu số thực dùng để lưu các số thực hay các số có dấu chấm thập phân gồm có 3 kiểu sau:

STT	Kiểu dữ liệu	Kích thước (Size)	Miền giá trị (Domain)
1	float	4 bytes	Từ $3.4 * 10^{-38}$ đến $3.4 * 10^{38}$
2	double	8 bytes	Từ $1.7 * 10^{-308}$ đến $1.7 * 10^{308}$
3	long double	10 bytes	Từ $3.4 * 10^{-4932}$ đến $1.1 * 10^{4932}$

Mỗi kiểu số thực ở trên đều có miền giá trị và độ chính xác (số số lẻ) khác nhau. Tùy vào nhu cầu sử dụng mà ta có thể khai báo biến thuộc 1 trong 3 kiểu trên.

Ngoài ra ta còn có kiểu dữ liệu **void**, kiểu này mang ý nghĩa là kiểu rỗng không chứa giá trị gì cả.

V. Tên và hằng trong C

V.1 Tên (danh biểu)

Tên hay còn gọi là danh biểu (identifier) được dùng để đặt cho chương trình, hằng, kiểu, biến, chương trình con... Tên có hai loại là tên chuẩn và tên do người lập trình đặt.

Tên chuẩn là tên do C đặt sẵn như tên kiểu: int, char, float,...; tên hàm: sin, cos...

Tên do người lập trình tự đặt để dùng trong chương trình của mình. Sử dụng bộ chữ cái, chữ số và dấu gạch dưới (_) để đặt tên, nhưng phải tuân thủ quy tắc:

- Bắt đầu bằng một chữ cái hoặc dấu gạch dưới.
- Không có khoảng trống ở giữa tên.

- Không được trùng với từ khóa.
- Độ dài tối đa của tên là không giới hạn, tuy nhiên chỉ có 31 ký tự đầu tiên là có ý nghĩa.
- Không cấm việc đặt tên trùng với tên chuẩn nhưng khi đó ý nghĩa của tên chuẩn không còn giá trị nữa.

Ví dụ: tên do người lập trình đặt: Chieu_dai, Chieu_Rong, Chu_Vi, Dien_Tich
Tên không hợp lệ: Do Dai, 12A2,...

V.2. Hằng (Constant)

Là đại lượng không đổi trong suốt quá trình thực thi của chương trình.

Hằng có thể là một chuỗi ký tự, một ký tự, một con số xác định. Chúng có thể được biểu diễn hay định dạng (Format) với nhiều dạng thức khác nhau.

V.2.1 Hằng số thực

Số thực bao gồm các giá trị kiểu float, double, long double được thể hiện theo 2 cách sau:

- *Cách 1:* Sử dụng cách viết thông thường mà chúng ta đã sử dụng trong các môn Toán, Lý, ... Điều cần lưu ý là sử dụng dấu thập phân là dấu chấm (.);

Ví dụ: 123.34 -223.333 3.00 -56.0

- *Cách 2:* Sử dụng cách viết theo số mũ hay số khoa học. Một số thực được tách làm 2 phần, cách nhau bằng ký tự e hay E

Phần giá trị: là một số nguyên hay số thực được viết theo cách 1.

Phần mũ: là một số nguyên

Giá trị của số thực là: *Phần giá trị nhân với 10 mũ phần mũ.*

Ví dụ: 1234.56e-3 = 1.23456 (là số 1234.56 * 10⁻³)

-123.45E4 = -1234500 (là -123.45 * 10⁴)

V.2.2 Hằng số nguyên

Số nguyên gồm các kiểu int (2 bytes), long (4 bytes) được thể hiện theo những cách sau.

- *Hằng số nguyên 2 bytes (int) hệ thập phân:* Là kiểu số mà chúng ta sử dụng thông thường, hệ thập phân sử dụng các ký số từ 0 đến 9 để biểu diễn một giá trị nguyên.

Ví dụ: 123 (một trăm hai mươi ba), -242 (trừ hai trăm bốn mươi hai).

- *Hằng số nguyên 2 byte (int) hệ bát phân:* Là kiểu số nguyên sử dụng 8 ký số từ 0 đến 7 để biểu diễn một số nguyên.

Cách biểu diễn: 0<các ký số từ 0 đến 7>

Ví dụ: 0345 (số 345 trong hệ bát phân)

-020 (số -20 trong hệ bát phân)

Cách tính giá trị thập phân của số bát phân như sau:

Số bát phân : 0d_nd_{n-1}d_{n-2}...d₁d₀ (d_i có giá trị từ 0 đến 7)

=> Giá trị thập phân = $\sum_{i=0}^n d_i * 8^i$

0345=229 , 020=16

- **Hằng số nguyên 2 byte (int) hệ thập lục phân:** Là kiểu số nguyên sử dụng 10 ký số từ 0 đến 9 và 6 ký tự A, B, C, D, E, F để biểu diễn một số nguyên.

Ký tự	giá trị
A	10
B	11
C	12
D	13
E	14
F	15

Cách biểu diễn: 0x<các ký số từ 0 đến 9 và 6 ký tự từ A đến F>

Ví dụ:

0x345 (số 345 trong hệ 16)

0x20 (số 20 trong hệ 16)

0x2A9 (số 2A9 trong hệ 16)

Cách tính giá trị thập phân của số thập lục phân như sau:

Số thập lục phân : 0xd_nd_{n-1}d_{n-2}...d₁d₀ (d_i từ 0 đến 9 hoặc A đến F)

$$\Rightarrow \text{Giá trị thập phân} = \sum_{i=0}^n d_i * 16^i$$

$$0x345=827 \quad , \quad 0x20=32 \quad , \quad 0x2A9= 681$$

- **Hằng số nguyên 4 byte (long):** Số long (số nguyên dài) được biểu diễn như số int trong hệ thập phân và kèm theo ký tự l hoặc L. Một số nguyên nằm ngoài miền giá trị của số int (2 bytes) là số long (4 bytes).

Ví dụ: 45345L hay 45345l hay 45345

- **Các hằng số còn lại:** Viết như cách viết thông thường (không có dấu phân cách giữa 3 số)

Ví dụ:

12 (mười hai)

12.45 (mười hai chấm 45)

1345.67 (một ba trăm bốn mươi lăm chấm sáu mươi bảy)

V.2.3. Hằng ký tự

Hằng ký tự là một ký tự riêng biệt được viết trong cặp dấu nháy đơn ('). Mỗi một ký tự tương ứng với một giá trị trong bảng mã ASCII. Hằng ký tự cũng được xem như trị số nguyên.

Ví dụ: 'a', 'A', '0', '9'

Chúng ta có thể thực hiện các phép toán số học trên 2 ký tự (thực chất là thực hiện phép toán trên giá trị ASCII của chúng)

V.2.4. Hằng chuỗi ký tự

Hằng chuỗi ký tự là một chuỗi hay một xâu ký tự được đặt trong cặp dấu nháy kép (").

Ví dụ: "Ngon ngu lap trinh C", "Khoa CNTT-DHCT", "NVLinh-DVHieu"

Chú ý:

1. Một chuỗi không có nội dung "" được gọi là chuỗi rỗng.

2. Khi lưu trữ trong bộ nhớ, một chuỗi được kết thúc bằng ký tự NULL ('\0': mã Ascii là 0).

3. Để biểu diễn ký tự đặc biệt bên trong chuỗi ta phải thêm dấu \ phía trước.

Ví dụ: "I'm a student" phải viết "I\'m a student"

"Day la ky tu "dac biet"" phải viết "Day la ky tu \"dac biet\""

VI. BIẾN VÀ BIỂU THỨC

VI.1. Biến

Biến là một đại lượng được người lập trình định nghĩa và được đặt tên thông qua việc khai báo biến. Biến dùng để chứa dữ liệu trong quá trình thực hiện chương trình và giá trị của biến có thể bị thay đổi trong quá trình này. Cách đặt tên biến giống như cách đặt tên đã nói trong phần trên.

Mỗi biến thuộc về một kiểu dữ liệu xác định và có giá trị thuộc kiểu đó.

VI.1.1. Cú pháp khai báo biến:

<Kiểu dữ liệu> **Danh sách các tên biến cách nhau bởi dấu phẩy;**

Ví dụ:

```
int    a, b, c;           /*Ba biến a, b,c có kiểu int*/
long int chu_vi;        /*Biến chu_vi có kiểu long*/
float  nua_chu_vi;      /*Biến nua_chu_vi có kiểu float*/
double dien_tich;      /*Biến dien_tich có kiểu double*/
```

Lưu ý: Để kết thúc 1 lệnh phải có dấu chấm phẩy (;) ở cuối lệnh.

VI.1.2. Vị trí khai báo biến trong C

Trong ngôn ngữ lập trình C, ta phải khai báo biến đúng vị trí. Nếu khai báo (đặt các biến) không đúng vị trí sẽ dẫn đến những sai sót ngoài ý muốn mà người lập trình không lường trước (hiệu ứng lè). Chúng ta có 2 cách đặt vị trí của biến như sau:

a) Khai báo biến ngoài: Các biến này được đặt bên ngoài tất cả các hàm và nó có tác dụng hay ảnh hưởng đến toàn bộ chương trình (còn gọi là biến toàn cục).

Ví dụ:

```
int    i;                /*Bien ben ngoai */
float  pi;               /*Bien ben ngoai*/
int    main()
{ ... }
```

b) Khai báo biến trong: Các biến được đặt ở bên trong hàm, chương trình chính hay một khối lệnh. Các biến này chỉ có tác dụng hay ảnh hưởng đến hàm, chương trình hay khối lệnh chứa nó. Khi khai báo biến, phải đặt các **biến này ở đầu của khối lệnh**, trước các lệnh gán, ...

Ví dụ 1:

```
#include <stdio.h>
#include<conio.h>
int bienngoai;           /*khai bao bien ngoai*/
int main ()
{   int j,i; /*khai bao bien ben trong chuong trinh chinh*/
    clrscr();
    i=1; j=2;
    bienngoai=3;
```

```

printf("\n Gia7 tri cua i la %d",i);
    /*%d là số nguyên, sẽ biết sau */
printf("\n Gia tri cua j la %d",j);
printf("\n Gia tri cua bienngoai la %d",bienngoai);
getch();
return 0;
}

```

Ví dụ 2:

```

#include <stdio.h>
#include<conio.h>
int main ()
{ int i, j;                                /*Bien ben trong*/
  clrscr();
  i=4; j=5;
  printf("\n Gia tri cua i la %d",i);
  printf("\n Gia tri cua j la %d",j);
  if(j>i)
  {
    int hieu=j-i;                          /*Bien ben trong */
    printf("\n Hieu so cua j tru i la %d",hieu);
  }
  else
  {
    int hieu=i-j                            ; /*Bien ben trong*/
    printf("\n Gia tri cua i tru j la %d",hieu);
  }
  getch();
  return 0;
}

```

VI.2. Biểu thức

Biểu thức là một sự kết hợp giữa các toán tử (operator) và các toán hạng (operand) theo đúng một trật tự nhất định.

Mỗi toán hạng có thể là một hằng, một biến hoặc một biểu thức khác.

Trong trường hợp, biểu thức có nhiều toán tử, ta dùng cặp dấu ngoặc đơn () để chỉ định toán tử nào được thực hiện trước.

Ví dụ: Biểu thức nghiệm của phương trình bậc hai:

$$(-b + \sqrt{\Delta})/(2*a)$$

Trong đó 2 là hằng; a, b, Delta là biến.

VI.2.1 Các toán tử số học

Trong ngôn ngữ C, các toán tử +, -, *, / làm việc tương tự như khi chúng làm việc trong các ngôn ngữ khác. Ta có thể áp dụng chúng cho đa số kiểu dữ liệu có sẵn được cho phép bởi C. Khi ta áp dụng phép / cho một số nguyên hay một ký tự, bất kỳ phần dư nào cũng bị cắt bỏ. Chẳng hạn, 5/2 bằng 2 trong phép chia nguyên.

Toán tử	Ý nghĩa
+	Cộng
-	Trừ
*	Nhân
/	Chia

%	Chia lấy phần dư
--	Giảm 1 đơn vị
++	Tăng 1 đơn vị

Tăng và giảm (++ & --)

Toán tử ++ thêm 1 vào toán hạng của nó và – trừ bớt 1. Nói cách khác:

$$x = x + 1 \text{ giống như } ++x$$

$$x = x - 1 \text{ giống như } x--$$

Cả 2 toán tử tăng và giảm đều có thể tiền tố (đặt trước) hay hậu tố (đặt sau) toán hạng. Ví dụ: $x = x + 1$ có thể viết $x++$ (hay $++x$)

Tuy nhiên giữa tiền tố và hậu tố có sự khác biệt khi sử dụng trong 1 biểu thức. Khi 1 toán tử tăng hay giảm đứng trước toán hạng của nó, C thực hiện việc tăng hay giảm trước khi lấy giá trị dùng trong biểu thức. Nếu toán tử đi sau toán hạng, C lấy giá trị toán hạng trước khi tăng hay giảm nó. Tóm lại:

$$x = 10$$

$$y = ++x // y = 11$$

Tuy nhiên:

$$x = 10$$

$$x = x++ // y = 10$$

Thứ tự ưu tiên của các toán tử số học:

++ -- sau đó là * / % rồi mới đến + -

VI.2.2 Các toán tử quan hệ và các toán tử Logic

Ý tưởng chính của toán tử quan hệ và toán tử Logic là đúng hoặc sai. Trong C mọi giá trị khác 0 được gọi là đúng, còn sai là 0. Các biểu thức sử dụng các toán tử quan hệ và Logic trả về 0 nếu sai và trả về 1 nếu đúng.

Toán tử	Ý nghĩa
Các toán tử quan hệ	
>	Lớn hơn
>=	Lớn hơn hoặc bằng
<	Nhỏ hơn
<=	Nhỏ hơn hoặc bằng
==	Bằng
!=	Khác
Các toán tử Logic	
&&	AND
	OR
!	NOT

Bảng chân trị cho các toán tử Logic:

P	q	p&&q	p q	!p
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

Các toán tử quan hệ và Logic đều có độ ưu tiên thấp hơn các toán tử số học. Do đó một biểu thức như: $10 > 1 + 12$ sẽ được xem là $10 > (1 + 12)$ và kết quả là sai (0).

Ta có thể kết hợp vài toán tử lại với nhau thành biểu thức như sau:

$10 > 5 \&\& !(10 < 9) || 3 <= 4$ Kết quả là đúng

Thứ tự ưu tiên của các toán tử quan hệ là Logic

Cao nhất: !
 > >= < <=
 == !=
 &&
 Thấp nhất: ||

VI.2.3 Các toán tử Bitwise:

Các toán tử Bitwise ý nói đến kiểm tra, gán hay sự thay đổi các Bit thật sự trong 1 Byte của Word, mà trong C chuẩn là các kiểu dữ liệu và biến char, int. Ta không thể sử dụng các toán tử Bitwise với dữ liệu thuộc các kiểu float, double, long double, void hay các kiểu phức tạp khác.

Toán tử	Ý nghĩa
&	AND
	OR
^	XOR
~	NOT
>>	Dịch phải
<<	Dịch trái

Bảng chân trị của toán tử ^ (XOR)

p	q	p^q
0	0	0
0	1	1
1	0	1
1	1	0

VI.2.4 Toán tử ? cùng với :

C có một toán tử rất mạnh và thích hợp để thay thế cho các câu lệnh của If-Then-Else. Cú pháp của việc sử dụng toán tử ? là:

$E1 \ ? \ E2 \ : \ E3$

Trong đó E1, E2, E3 là các biểu thức.

Ý nghĩa: Trước tiên E1 được ước lượng, nếu đúng E2 được ước lượng và nó trở thành giá trị của biểu thức; nếu E1 sai, E2 được ước lượng và trở thành giá trị của biểu thức.

Ví dụ:

$X = 10$
 $Y = X > 9 \ ? \ 100 \ : \ 200$

Thì Y được gán giá trị 100, nếu X nhỏ hơn 9 thì Y sẽ nhận giá trị là 200. Đoạn mã này tương đương cấu trúc if như sau:

```
X = 10
if (X < 9) Y = 100
else Y = 200
```

VII.2.5 Toán tử con trỏ & và *

Một con trỏ là địa chỉ trong bộ nhớ của một biến. Một biến con trỏ là một biến được khai báo riêng để chứa một con trỏ đến một đối tượng của kiểu đã chỉ ra nó. Ta sẽ tìm hiểu kỹ hơn về con trỏ trong chương về con trỏ. Ở đây, chúng ta sẽ đề cập ngắn gọn đến hai toán tử được sử dụng để thao tác với các con trỏ.

Toán tử thứ nhất là &, là một toán tử quy ước trả về địa chỉ bộ nhớ của hệ số của nó.

Ví dụ: `m = &count`

Đặt vào biến m địa chỉ bộ nhớ của biến count.

Chẳng hạn, biến count ở vị trí bộ nhớ 2000, giả sử count có giá trị là 100. Sau câu lệnh trên m sẽ nhận giá trị 2000.

Toán tử thứ hai là *, là một bổ sung cho &; đây là một toán tử quy ước trả về giá trị của biến được cấp phát tại địa chỉ theo sau đó.

Ví dụ: `q = *m`

Sẽ đặt giá trị của count vào q. Bây giờ q sẽ có giá trị là 100 vì 100 được lưu trữ tại địa chỉ 2000.

VI.2.6 Toán tử dấu phẩy ,

Toán tử dấu , được sử dụng để kết hợp các biểu thức lại với nhau. Bên trái của toán tử dấu , luôn được xem là kiểu void. Điều đó có nghĩa là biểu thức bên phải trở thành giá trị của tổng các biểu thức được phân cách bởi dấu phẩy.

Ví dụ: `x = (y=3,y+1);`

Trước hết gán 3 cho y rồi gán 4 cho x. Cặp dấu ngoặc đơn là cần thiết vì toán tử dấu , có độ ưu tiên thấp hơn toán tử gán.

VI.2.7 Xem các dấu ngoặc đơn và cặp dấu ngoặc vuông là toán tử

Trong C, cặp dấu ngoặc đơn là toán tử để tăng độ ưu tiên của các biểu thức bên trong nó.

Các cặp dấu ngoặc vuông thực hiện thao tác truy xuất phần tử trong mảng.

VI.2.8 Tổng kết về độ ưu tiên

Cao nhất	() []
	! ~ ++ -- (Kiểu) * &
	* / %
	+ -
	<< >>
	< <= > >=
	&
	^

	&&
	?:
	= += -= *= /=
Thấp nhất	,

VI.2.9 Cách viết tắt trong C

Có nhiều phép gán khác nhau, đôi khi ta có thể sử dụng viết tắt trong C nữa. Chẳng hạn:

$x = x + 10$ được viết thành $x += 10$

Toán tử += báo cho chương trình dịch biết để tăng giá trị của x lên 10.

Cách viết này làm việc trên tất cả các toán tử nhị phân (phép toán hai ngôi) của C. Tổng quát:

(Biến) = (Biến) (Toán tử) (Biểu thức)

có thể được viết:

(Biến) (Toán tử) = (Biểu thức)

VII. Cấu trúc của một chương trình C

VII.1. Tiền xử lý và biên dịch

Trong C, việc dịch (translation) một tập tin nguồn được tiến hành trên hai bước hoàn toàn độc lập với nhau:

- Tiền xử lý.
- Biên dịch.

Hai bước này trong phần lớn thời gian được nối tiếp với nhau một cách tự động theo cách thức mà ta có ấn tượng rằng nó đã được thực hiện như là một xử lý duy nhất. Nói chung, ta thường nói đến việc tồn tại của một bộ tiền xử lý (preprocessor?) nhằm chỉ rõ chương trình thực hiện việc xử lý trước. Ngược lại, các thuật ngữ trình biên dịch hay sự biên dịch vẫn còn nhập nhằng bởi vì nó chỉ ra khi thì toàn bộ hai giai đoạn, khi thì lại là giai đoạn thứ hai.

Bước tiền xử lý tương ứng với việc cập nhật trong văn bản của chương trình nguồn, chủ yếu dựa trên việc diễn giải các mã lệnh rất đặc biệt gọi là các chỉ thị dẫn hướng của bộ tiền xử lý (destination directive of preprocessor); các chỉ thị này được nhận biết bởi chúng bắt đầu bằng ký hiệu (symbol) #.

Hai chỉ thị quan trọng nhất là:

- Chỉ thị sự gộp vào của các tập tin nguồn khác: #include
- Chỉ thị việc định nghĩa các macros hoặc ký hiệu: #define

Chỉ thị đầu tiên được sử dụng trước hết là nhằm gộp vào nội dung của các tập tin cần có (header file), không thể thiếu trong việc sử dụng một cách tốt nhất các hàm của thư viện chuẩn, phổ biến nhất là:

```
#include <stdio.h>
```

Chỉ thị thứ hai rất hay được sử dụng trong các tập tin thư viện (header file) đã được định nghĩa trước đó và thường được khai thác bởi các lập trình viên trong việc định nghĩa các ký hiệu như là:

```
#define NB_COUPS_MAX 100
#define SIZE 25
```

VII.2 Cấu trúc một chương trình C

Một chương trình C bao gồm các phần như: Các chỉ thị tiền xử lý, khai báo biến ngoài, các hàm tự tạo, chương trình chính (hàm main).

Cấu trúc có thể như sau:

Các chỉ thị tiền xử lý (Preprocessor directives)

```
#include <Tên tập tin thư viện>
#define ....
```

Định nghĩa kiểu dữ liệu (phần này không bắt buộc): dùng để đặt tên lại cho một kiểu dữ liệu nào đó để gọi nhớ hay đặt 1 kiểu dữ liệu cho riêng mình dựa trên các kiểu dữ liệu đã có.

Cú pháp: `typedef <Tên kiểu cũ> <Tên kiểu mới>`

Ví dụ: `typedef int SoNguyen; // Kiểu SoNguyen là kiểu int`

Khai báo các prototype (tên hàm, các tham số, kiểu kết quả trả về,... của các hàm sẽ cài đặt trong phần sau, phần này không bắt buộc): phần này chỉ là các khai báo đầu hàm, không phải là phần định nghĩa hàm.

Khai báo các biến ngoài (các biến toàn cục) *phần này không bắt buộc*: phần này khai báo các biến toàn cục được sử dụng trong cả chương trình.

Chương trình chính phần này bắt buộc phải có

```
<Kiểu dữ liệu trả về> main()
```

```
{
```

Các khai báo cục bộ trong hàm main: Các khai báo này chỉ tồn tại trong hàm mà thôi, có thể là khai báo biến hay khai báo kiểu.

Các câu lệnh dùng để định nghĩa hàm

```
return <kết quả trả về>; // Hàm phải trả về kết quả
```

```
}
```

Cài đặt các hàm

```
<Kiểu dữ liệu trả về> function1( các tham số)
```

```
{
```

Các khai báo cục bộ trong hàm.

Các câu lệnh dùng để định nghĩa hàm

```
return <kết quả trả về>;
```

```
}
```

```
...
```


Một chương trình C bắt đầu thực thi từ hàm main (thông thường là từ câu lệnh đầu tiên đến câu lệnh cuối cùng).

VII.3 Các tập tin thư viện thông dụng

Đây là các tập tin chứa các hàm thông dụng khi lập trình C, muốn sử dụng các hàm trong các tập tin header này thì phải khai báo `#include <Tên tập tin>` ở phần đầu của chương trình

1) stdio.h: Tập tin định nghĩa các hàm vào/ra chuẩn (standard input/output). Gồm các hàm in dữ liệu (`printf()`), nhập giá trị cho biến (`scanf()`), nhận ký tự từ bàn phím (`getc()`), in ký tự ra màn hình (`putc()`), nhận một dãy ký tự từ bàn phím (`gets()`), in chuỗi ký tự ra màn hình (`puts()`), xóa vùng đệm bàn phím (`fflush()`), `fopen()`, `fclose()`, `fread()`, `fwrite()`, `getchar()`, `putchar()`, `getw()`, `putw()`...

2) conio.h : Tập tin định nghĩa các hàm vào ra trong chế độ DOS (DOS console). Gồm các hàm `clrscr()`, `getch()`, `getche()`, `getpass()`, `cgets()`, `cputs()`, `putch()`, `clreol()`,...

3) math.h: Tập tin định nghĩa các hàm tính toán gồm các hàm `abs()`, `sqrt()`, `log()`, `log10()`, `sin()`, `cos()`, `tan()`, `acos()`, `asin()`, `atan()`, `pow()`, `exp()`,...

4) alloc.h: Tập tin định nghĩa các hàm liên quan đến việc quản lý bộ nhớ. Gồm các hàm `calloc()`, `realloc()`, `malloc()`, `free()`, `farmalloc()`, `farcalloc()`, `farfree()`, ...

5) io.h: Tập tin định nghĩa các hàm vào ra cấp thấp. Gồm các hàm `open()`, `_open()`, `read()`, `_read()`, `close()`, `_close()`, `creat()`, `_creat()`, `creatnew()`, `eof()`, `filelength()`, `lock()`,...

6) graphics.h: Tập tin định nghĩa các hàm liên quan đến đồ họa. Gồm `initgraph()`, `line()`, `circle()`, `putpixel()`, `getpixel()`, `setcolor()`, ...

Còn nhiều tập tin khác nữa.

VII.4 Cú pháp khai báo các phần bên trong một chương trình C

VII.4.1. Chỉ thị `#include` để sử dụng tập tin thư viện

Cú pháp:

```
#include <Tên tập tin> // Tên tập tin được đặt trong dấu <>
hay #include "Tên đường dẫn"
```

Menu Option của Turbo C có mục INCLUDE DIRECTORIES, mục này dùng để chỉ định các tập tin thư viện được lưu trữ trong thư mục nào.

Nếu ta dùng `#include<Tên tập tin>` thì Turbo C sẽ tìm tập tin thư viện trong thư mục đã được xác định trong INCLUDE DIRECTORIES.

Ví dụ: `include <stdio.h>`

Nếu ta dùng `#include"Tên đường dẫn"` thì ta phải chỉ rõ tên ở đâu, tên thư mục và tập tin thư viện.

Ví dụ: `#include"C:\\TC\\math.h"`

Trong trường hợp tập tin thư viện nằm trong thư mục hiện hành thì ta chỉ cần đưa tên tập tin thư viện. Ví dụ: `#include"math.h"`.

Ví dụ:

```
#include <stdio.h>
#include <conio.h>
#include "math.h"
```

VII.4.2. Chỉ thị #define để định nghĩa hằng số

Cú pháp:

```
#define <Tên hằng> <Giá trị>
```

Ví dụ:

```
#define MAXINT 32767
```

VII.4.3. Khai báo các prototype của hàm

Cú pháp:

```
<Kiểu kết quả trả về> Tên hàm (danh sách đối số)
```

Ví dụ:

```
long giaiithua( int n); //Hàm tính giai thừa của số nguyên n
double x_mu_y(float x, float y); /*Hàm tính x mũ y*/
```

VII.4.4. Cấu trúc của hàm “bình thường”

Cú pháp:

```
<Kiểu kết quả trả về> Tên hàm (các đối số)
```

```
{
```

```
    Các khai báo và các câu lệnh định nghĩa hàm
    return kết quả;
```

```
}
```

Ví dụ:

```
int tong(int x, int y) /*Hàm tính tổng 2 số nguyên*/
{
    return (x+y);
}
float tong(float x, float y) /*Hàm tính tổng 2 số thực*/
{
    return (x+y);
}
```

VII.4.5. Cấu trúc của hàm main

Hàm main chính là chương trình chính, gồm các lệnh xử lý, các lời gọi các hàm khác.

Cú pháp:

```
<Kết quả trả về> main( đối số)
```

```
{
```

```
    Các khai báo và các câu lệnh định nghĩa hàm
    return <kết quả>;
```

```
}
```

Ví dụ 1:

```
int main()
{
    printf("Day la chuong trinh chinh");
    getch();
}
```

```
    return 0;  
}
```

Ví dụ 2:

```
int main()  
{  
    int a=5, b=6, c;  
    float x=3.5, y=4.5, z;  
    printf("Day la chuong trinh chinh");  
    c=tong(a,b);  
    printf("\n Tong cua %d va %d la %d", a,b,c);  
    z=tong(x,y);  
    printf("\n Tong cua %f và %f là %f", x,y,z);  
    getch();  
    return 0;  
}
```

VIII. BÀI TẬP

Bài 1: Biểu diễn các hằng số nguyên 2 byte sau đây dưới dạng số nhị phân, bát phân, thập lục phân

a) 12 b) 255 c) 31000 d) 32767 e) -32768

Bài 2: Biểu diễn các hằng ký tự sau đây dưới dạng số nhị phân, bát phân.

a) 'A' b) 'a' c) 'Z' d) 'z'

Chương 3

CÁC CÂU LỆNH ĐƠN TRONG C

Học xong chương này, sinh viên sẽ nắm rõ các vấn đề sau:

- Câu lệnh là gì?
- Cách sử dụng câu lệnh gán giá trị của một biểu thức cho một biến.
- Cách sử dụng lệnh `scanf` để nhập giá trị cho biến.
- Cách sử dụng lệnh `printf` để xuất giá trị của biểu thức lên màn hình và cách định dạng dữ liệu.

I. Câu lệnh

I.1. Khái niệm câu lệnh

Một câu lệnh (statement) xác định một công việc mà chương trình phải thực hiện để xử lý dữ liệu đã được mô tả và khai báo. Các câu lệnh được ngăn cách với nhau bởi dấu chấm phẩy (;).

I.2. Phân loại

Có hai loại lệnh: lệnh đơn và lệnh có cấu trúc.

Lệnh đơn là một lệnh không chứa các lệnh khác. Các lệnh đơn gồm: lệnh gán, các câu lệnh nhập xuất dữ liệu...

Lệnh có cấu trúc là lệnh trong đó chứa các lệnh khác. Lệnh có cấu trúc bao gồm: cấu trúc điều kiện rẽ nhánh, cấu trúc điều kiện lựa chọn, cấu trúc lặp và cấu trúc lệnh hợp thành. Lệnh hợp thành (khối lệnh) là một nhóm bao gồm nhiều khai báo biến và các lệnh được gom vào trong cặp dấu {}.

II. CÁC LỆNH ĐƠN

II.1. Lệnh gán

Lệnh gán (assignment statement) dùng để gán giá trị của một biểu thức cho một biến.

Cú pháp: <Tên biến> = <biểu thức>

Ví dụ:

```
int main() {
    int x, y;
    x = 10; /*Gán hằng số 10 cho biến x*/
    y = 2*x; /*Gán giá trị 2*x=2*10=20 cho x*/
    return 0;
}
```

Nguyên tắc khi dùng lệnh gán là kiểu của biến và kiểu của biểu thức phải giống nhau, gọi là có sự tương thích giữa các kiểu dữ liệu. Chẳng hạn ví dụ sau cho thấy một sự không tương thích về kiểu:

```
int main() {
    int x, y;
    x = 10; /*Gán hằng số 10 cho biến x*/
    y = "Xin chao";
        /*y có kiểu int, còn "Xin chao" có kiểu char* */
    return 0;
}
```

Khi biên dịch chương trình này, C sẽ báo lỗi "*Cannot convert 'char *' to 'int'*" tức là C không thể tự động chuyển đổi kiểu từ char * (chuỗi ký tự) sang int.

Tuy nhiên trong đa số trường hợp sự tự động biến đổi kiểu để sự tương thích về kiểu sẽ được thực hiện. *Ví dụ:*

```
int main() {
    int x, y;
    float r;
    char ch;
    r = 9000;
    x = 10; /* Gán hằng số 10 cho biến x */
    y = 'd'; /* y có kiểu int, còn 'd' có kiểu char*/
    r = 'e'; /* r có kiểu float, 'e' có kiểu char*/
    ch = 65.7; /* ch có kiểu char, còn 65.7 có kiểu float*/
    return 0;
}
```

Trong nhiều trường hợp để tạo ra sự tương thích về kiểu, ta phải sử dụng đến cách thức chuyển đổi kiểu một cách tường minh. Cú pháp của phép toán này như sau:

(Tên kiểu) <Biểu thức>

Chuyển đổi kiểu của <Biểu thức> thành kiểu mới <Tên kiểu>. Chẳng hạn như:

```
float f;
f = (float) 10 / 4; /* f lúc này là 2.5*/
```

Chú ý:

- Khi một biểu thức được gán cho một biến thì giá trị của nó sẽ thay thế giá trị cũ mà biến đã lưu giữ trước đó.

- Trong câu lệnh gán, dấu = là một toán tử; do đó nó có thể được sử dụng là một thành phần của biểu thức. Trong trường hợp này giá trị của biểu thức gán chính là giá trị của biến.

Ví dụ:

```
int x, y;
y = x = 3; /* y lúc này cùng bằng 3*/
```

- Ta có thể gán trị cho biến lúc biến được khai báo theo cách thức sau:

<Tên kiểu> <Tên biến> = <Biểu thức>;

Ví dụ: int x = 10, y=x;

II.2. Lệnh nhập giá trị từ bàn phím cho biến (hàm scanf)

Là hàm cho phép đọc dữ liệu từ bàn phím và gán cho các biến trong chương trình khi chương trình thực thi. Trong ngôn ngữ C, đó là hàm scanf nằm trong thư viện stdio.h.

Cú pháp:

scanf("Chuỗi định dạng", địa chỉ của các biến);

Giải thích:

- *Chuỗi định dạng*: dùng để qui định kiểu dữ liệu, cách biểu diễn, độ rộng, số chữ số thập phân... Một số định dạng khi nhập kiểu số nguyên, số thực, ký tự.

Định dạng	Ý nghĩa
%[số ký số]d	Nhập số nguyên có tối đa <số ký số>
%[số ký số]f	Nhập số thực có tối đa <số ký số> tính cả dấu chấm
%c	Nhập một ký tự
<i>Ví dụ:</i>	
%d	Nhập số nguyên
%4d	Nhập số nguyên tối đa 4 ký số, nếu nhập nhiều hơn 4 ký số thì chỉ nhận được 4 ký số đầu tiên
%f	Nhập số thực
%6f	Nhập số thực tối đa 6 ký số (tính luôn dấu chấm), nếu nhập nhiều hơn 6 ký số thì chỉ nhận được 6 ký số đầu tiên (hoặc 5 ký số với dấu chấm)

- *Địa chỉ của các biến*: là địa chỉ (&) của các biến mà chúng ta cần nhập giá trị cho nó. Được viết như sau: **&<tên biến>**.

Ví dụ:

```
scanf("%d",&bien1); /*Doc gia tri cho bien1 co kieu nguyen*/
scanf("%f",&bien2); /*Doc gia tri cho bien2 co kieu thuc*/
scanf("%d%f",&bien1,&bien2);
/*Doc gia tri cho bien1 co kieu nguyen, bien2 co kieu thuc*/
scanf("%d%f%c",&bien1,&bien2,&bien3);
/*bien3 co kieu char*/
```

Lưu ý:

- Chuỗi định dạng phải đặt trong cặp dấu nháy kép ("").
- Các biến (địa chỉ biến) phải cách nhau bởi dấu phẩy (,).
- Có bao nhiêu biến thì phải có bấy nhiêu định dạng.
- Thứ tự của các định dạng phải phù hợp với thứ tự của các biến.
- Để nhập giá trị kiểu char được chính xác, nên dùng hàm *fflush(stdin)* để loại bỏ các ký tự còn nằm trong vùng đệm bàn phím trước hàm scanf().
- Để nhập vào một chuỗi ký tự (không chứa khoảng trắng hay kết thúc bằng khoảng trắng), chúng ta phải khai báo kiểu *mảng ký tự* hay *con trỏ ký tự*, sử dụng định dạng %s và tên biến thay cho địa chỉ biến.
- Để đọc vào một chuỗi ký tự có chứa khoảng trắng (kết thúc bằng phím Enter) thì phải dùng hàm gets().

Ví dụ:

```
int    biennguyen;
float  bienthuc;
char   bienchar;
char   chuoil[20], *chuois;
```

Nhập giá trị cho các biến:

```
scanf("%3d", &biennghuyen);
```

Nếu ta nhập 1234455 thì giá trị của `biennghuyen` là 3 ký số đầu tiên (123). Các ký số còn lại sẽ còn nằm lại trong vùng đệm.

```
scanf("%5f", &bienthuc);
```

Nếu ta nhập 123.446 thì giá trị của `bienthuc` là 123.4, các ký số còn lại sẽ còn nằm trong vùng đệm.

```
scanf("%2d%5f", &biennghuyen, &bienthuc);
```

Nếu ta nhập liên tiếp 2 số cách nhau bởi khoảng trắng như sau: 1223 3.142325

- 2 ký số đầu tiên (12) sẽ được đọc vào cho `biennghuyen`.

- 2 ký số tiếp theo trước khoảng trắng (23) sẽ được đọc vào cho `bienthuc`.

```
scanf("%2d%5f%c", &biennghuyen, &bienthuc, &bienchar)
```

Nếu ta nhập liên tiếp 2 số cách nhau bởi khoảng trắng như sau: 12345 3.142325:

- 2 ký số đầu tiên (12) sẽ được đọc vào cho `biennghuyen`.

- 3 ký số tiếp theo trước khoảng trắng (345) sẽ được đọc vào cho `bienthuc`.

- Khoảng trắng sẽ được đọc cho `bienchar`.

Nếu ta chỉ nhập 1 số gồm nhiều ký số như sau: 123456789:

- 2 ký số đầu tiên (12) sẽ được đọc vào cho `biennghuyen`.

- 5 ký số tiếp theo (34567) sẽ được đọc vào cho `bienthuc`.

- `bienchar` sẽ có giá trị là ký số tiếp theo '8'.

```
scanf("%s", chuoi1); hoặc scanf("%s", chuoi2)
```

Nếu ta nhập chuỗi như sau: *Nguyen Van Linh* ↵ thì giá trị của biến `chuoi1` hay `chuoi2` chỉ là *Nguyen*.

```
scanf("%s%s", chuoi1, chuoi2);
```

Nếu ta nhập chuỗi như sau: *Duong Van Hieu* ↵ thì giá trị của biến `chuoi1` là *Duong* và giá trị của biến `chuoi2` là *Van*.

Vì sao như vậy? C sẽ đọc từ đầu đến khi gặp khoảng trắng và gán giá trị cho biến đầu tiên, phần còn lại sau khoảng trắng là giá trị của các biến tiếp theo.

```
gets(chuoi1);
```

Nếu nhập chuỗi : *Nguyen Van Linh* ↵ thì giá trị của biến `chuoi1` là *Nguyen Van Linh*

II.3. Lệnh xuất giá trị của biểu thức lên màn hình (hàm printf)

Hàm `printf` (nằm trong thư viện `stdio.h`) dùng để xuất giá trị của các biểu thức lên màn hình.

Cú pháp:

```
printf("Chuỗi định dạng", Các biểu thức);
```

Giải thích:

- *Chuỗi định dạng*: dùng để qui định kiểu dữ liệu, cách biểu diễn, độ rộng, số chữ số thập phân... Một số định dạng khi đối với số nguyên, số thực, ký tự.

Định dạng	Ý nghĩa
-----------	---------

%d	Xuất số nguyên
%[.số chữ số thập phân] f	Xuất số thực có <số chữ số thập phân> theo quy tắc làm tròn số.
%o	Xuất số nguyên hệ bát phân
%x	Xuất số nguyên hệ thập lục phân
%c	Xuất một ký tự
%s	Xuất chuỗi ký tự
%e hoặc %E hoặc %g hoặc %G	Xuất số nguyên dạng khoa học (nhân 10 mũ x)
<i>Ví dụ</i>	
%d	In ra số nguyên
%4d	In số nguyên tối đa 4 ký số, nếu số cần in nhiều hơn 4 ký số thì in hết
%f	In số thực
%6f	In số thực tối đa 6 ký số (tính luôn dấu chấm), nếu số cần in nhiều hơn 6 ký số thì in hết
%.3f	In số thực có 3 số lẻ, nếu số cần in có nhiều hơn 3 số lẻ thì làm tròn.

- Các biểu thức: là các biểu thức mà chúng ta cần xuất giá trị của nó lên màn hình, mỗi biểu thức phân cách nhau bởi dấu phẩy (,).

Ví dụ:

```
include<stdio.h>
int main() {
    int    bien_nguyen=1234, i=65;
    float  bien_thuc=123.456703;
    printf("Gia tri nguyen cua bien nguyen =%d\n",bien_nguyen);
    printf("Gia tri thuc cua bien thuc =%f\n",bien_thuc);
    printf("Truoc khi lam tron=%f \n
        Sau khi lam tron=%.2f",bien_thuc, bien_thuc);
    return 0;
}
```

Kết quả in ra màn hình như sau:

```
Gia tri nguyen cua bien nguyen =1234
Gia tri thuc cua bien thuc =123.456703
Truoc khi lam tron=123.456703
Sau khi lam tron=123.46
```

Nếu ta thêm vào dòng sau trong chương trình:

```
printf("\n Ky tu co ma ASCII %d la %c",i,i);
```

Kết quả ta nhận được thêm:

```
Ky tu co ma ASCII 65 la A_
```

```
printf(" So nguyen la %d \n So thuc la %f",i, (float)i );
```

```
So nguyen la 65
So thuc la 65.000000
```

```
printf("\n So thuc la %f \n So nguyen la %d",bien_thuc,
(int)bien_thuc);
```

```
So thuc la 123.456703
So nguyen la 123_
```

```
printf("\n Viet binh thuong =%f \n Viet kieu khoa
hoc=%e",bien_thuc, bien_thuc);
```

Kết quả in ra màn hình:


```
Viet binh thuong=123.456703
Viet kieu khoa hoc=1.234567e+02
```

Lưu ý: Đối với các ký tự điều khiển, ta không thể sử dụng cách viết thông thường để hiển thị chúng.

Ký tự điều khiển là các ký tự dùng để điều khiển các thao tác xuất, nhập dữ liệu.

Một số ký tự điều khiển được mô tả trong bảng:

Ký tự điều khiển	Giá trị thập lục phân	Ký tự được hiển thị	Ý nghĩa
\a	0x07	BEL	Phát ra tiếng chuông
\b	0x08	BS	Di chuyển con trỏ sang trái 1 ký tự và xóa ký tự bên trái (backspace)
\f	0x0C	FF	Sang trang
\n	0x0A	LF	Xuống dòng
\r	0x0D	CR	Trở về đầu dòng
\t	0x09	HT	Tab theo cột (giống gõ phím Tab)
\\	0x5C	\	Dấu \
\'	0x2C	'	Dấu nháy đơn (')
\"	0x22	"	Dấu nháy kép (")
\?	0x3F	?	Dấu chấm hỏi (?)
\ddd	ddd	Ký tự có mã ACSII trong hệ bát phân là số ddd	
\xHHH	oxHHH	Ký tự có mã ACSII trong hệ thập lục phân là HHH	

Ví dụ:

```
#include <stdio.h>
#include<conio.h>
int main ()
{
    clrscr();
    printf("\n Tieng Beep \a");
    printf("\n Doi con tro sang trai 1 ky tu\b");
    printf("\n Dau Tab \tva dau backslash \\");
    printf("\n Dau nhay don \' va dau nhay kep '\"");
    printf("\n Dau cham hoi \?");
    printf("\n Ky tu co ma bat phan 101 la \101");
    printf("\n Ky tu co ma thap luc phan 41 la \x041");
    printf("\n Dong hien tai, xin go enter");
    getch();
    printf("\rVe dau dong");
    getch();
    return 0;
}
```

Kết quả trước khi gõ phím Enter:

```
Tieng Beep
Doi con tro sang trai 1 ky tu
Dau Tab      va dau backslash \
Dau nhay don ' va dau nhay kep "
Dau cham hoi ?
Ky tu co ma bat phan 101 la A
Ky tu co ma thay luc phan 41 la A
Dong hien tai, xin go enter
```

Kết quả sau khi gõ phím Enter:

```
Tieng Beep
Doi con tro sang trai 1 ky tu
Dau Tab      va dau backslash \
Dau nhay don ' va dau nhay kep "
Dau cham hoi ?
Ky tu co ma bat phan 101 la A
Ky tu co ma thay luc phan 41 la A
Ve dau dongtai, xin go enter
```

III. BÀI TẬP

III.1. Mục đích yêu cầu

Làm quen và nắm vững các lệnh đơn giản (printf, scanf), các kiểu dữ liệu chuẩn (int, long, char, float...), các phép toán và các hàm chuẩn của ngôn ngữ lập trình C. Thực hiện viết các chương trình hoàn chỉnh sử dụng các lệnh đơn giản và các kiểu dữ liệu chuẩn đó.

III.2. Nội dung

1. Viết chương trình in lên màn hình một thiệp mời dự sinh nhật có dạng:

```
*****
THIỆP MỜI
Thân mời bạn : Nguyễn Mạnh Hùng
Tới dự lễ sinh nhật của mình
Vào lúc 19h ngày 12/10/2005
Tại 05/42 Trần Phú - Cần Thơ
Rất mong được đón tiếp !
HỒ Thu Hương
*****
```

2. Viết chương trình nhập vào bán kính r của một hình tròn. Tính chu vi và diện tích của hình tròn theo công thức :

Chu vi $CV = 2 * \pi * r$
Diện tích $S = \pi * r * r$

In các kết quả lên màn hình

3. Viết chương trình nhập vào độ dài 3 cạnh a, b, c của một tam giác. Tính chu vi và diện tích của tam giác theo công thức:

Chu vi $CV = a + b + c$

Diện tích $S = \sqrt{p(p-a)(p-b)(p-c)}$

Trong đó: $p = CV/2$

In các kết quả lên màn hình

4. Viết chương trình tính $\log_a x$ với a, x là các số thực nhập vào từ bàn phím, và $x > 0$, $a > 0$, $a \neq 1$. (dùng $\log_a x = \ln x / \ln a$)

5. Viết chương trình nhập vào tọa độ của hai điểm (x_1, y_1) và (x_2, y_2)

a) Tính hệ số góc của đường thẳng đi qua hai điểm đó theo công thức:

$$\text{Hệ số góc} = (y_2 - y_1) / (x_2 - x_1)$$

b) Tính khoảng cách giữa hai điểm theo công thức:

$$\text{Khoảng cách} = \sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2}$$

6. Viết chương trình nhập vào một ký tự:

a) In ra mã Ascii của ký tự đó.

b) In ra ký tự kế tiếp của nó.

7. Viết chương trình nhập vào các giá trị điện trở R_1, R_2, R_3 của một mạch điện :

Tính tổng trở theo công thức: $\frac{1}{R} = \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3}$

8. Viết chương trình nhập vào điểm ba môn Toán, Lý, Hóa của một học sinh. In ra điểm trung bình của học sinh đó với hai số lẻ thập phân.

9. Viết chương trình nhập vào ngày, tháng, năm. In ra ngày tháng năm theo dạng dd/mm/yy. (dd: ngày, mm: tháng, yy : năm. Ví dụ: 20/11/99)

10. Viết chương trình đảo ngược một số nguyên dương có đúng 3 chữ số.

Chương 4

CÁC LỆNH CÓ CẤU TRÚC

Học xong chương này, sinh viên sẽ nắm được các vấn đề sau:

- Khối lệnh trong C.
- Cấu trúc rẽ nhánh.
- Cấu trúc lựa chọn.
- Cấu trúc vòng lặp.
- Các câu lệnh “đặc biệt”.

I. KHỐI LỆNH

Một dãy các khai báo cùng với các câu lệnh nằm trong cặp dấu ngoặc móc { và } được gọi là một khối lệnh.

Ví dụ 1:

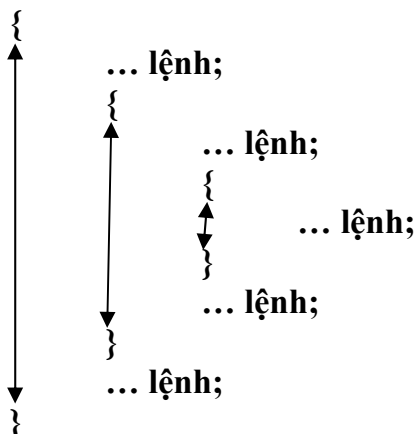
```
{
    char ten[30];
    printf("\n Nhập vào ten của bạn:");
    scanf("%s", ten);
    printf("\n Chào Bạn  %s",ten);
}
```

Ví dụ 2:

```
#include <stdio.h>
#include<conio.h>
int main ()
{ /*đây là đầu khối*/
    char ten[50];
    printf("Xin cho biet ten của bạn !");
    scanf("%s",ten);
    getch();
    return 0;
} /*đây là cuối khối*/
```

Một khối lệnh có thể chứa bên trong nó nhiều khối lệnh khác gọi là khối lệnh lồng nhau. Sự lồng nhau của các khối lệnh là không hạn chế.

Minh họa:



Lưu ý về phạm vi tác động của biến trong khối lệnh lồng nhau:

- Trong các khối lệnh khác nhau hay các khối lệnh lồng nhau có thể khai báo các biến cùng tên.

Ví dụ 1:

```
{
  ... lệnh;
  {
    int a,b;      /*biến a, b trong khối lệnh thứ nhất*/
    ... lệnh;
  }
  ...lệnh;
  {
    int a,b;      /*biến a,b trong khối lệnh thứ hai*/
    ... lệnh;
  }
}
```

Ví dụ 2:

```
{
  int a, b;      /*biến a,b trong khối lệnh “bên ngoài”*/
  ... lệnh;
  {
    int a,b;     /*biến a,b bên trong khối lệnh con*/
  }
}
```

- Nếu một biến được khai báo bên ngoài khối lệnh và không trùng tên với biến bên trong khối lệnh thì nó cũng được sử dụng bên trong khối lệnh.

- Một khối lệnh con có thể sử dụng các biến bên ngoài, các lệnh bên ngoài không thể sử dụng các biến bên trong khối lệnh con.

Ví dụ:

```
{
  int a, b, c;
  ...lệnh;
  {
    int c, d;
    ...lệnh;
  }
}
```

II. CẤU TRÚC RỄ NHÁNH

Cấu trúc rẽ nhánh là một cấu trúc được dùng rất phổ biến trong các ngôn ngữ lập trình nói chung. Trong C, có hai dạng: dạng không đầy đủ và dạng đầy đủ.

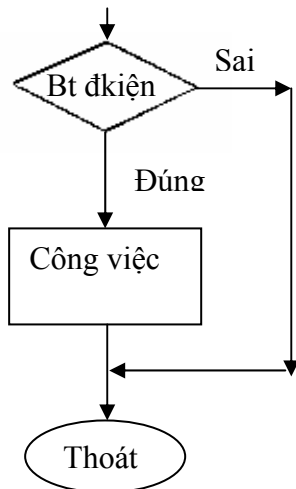
II.1. Dạng không đầy đủ

Cú pháp:

if (<Biểu thức điều kiện>)

<Công việc>

Lưu đồ cú pháp:



Giải thích:

<Công việc> được thể hiện bằng 1 câu lệnh hay 1 khối lệnh.

Kiểm tra *Biểu thức điều kiện* trước.

Nếu điều kiện đúng ($\neq 0$) thì thực hiện câu lệnh hoặc khối lệnh liền sau điều kiện.

Nếu điều kiện sai thì bỏ qua lệnh hoặc khối lệnh liền sau điều kiện (những lệnh và khối lệnh sau đó vẫn được thực hiện bình thường vì nó không phụ thuộc vào điều kiện sau if).

Ví dụ 1:

Yêu cầu người thực hiện chương trình nhập vào một số thực a. In ra màn hình kết quả nghịch đảo của a khi $a \neq 0$.

```

#include <stdio.h>
#include <conio.h>
int main ()
{
    float a;
    printf("Nhập a = "); scanf("%f",&a);
    if (a !=0 )
        printf("Nghich dao cua %f la %f",a,1/a);
    getch();
    return 0;
}
  
```

Giải thích:

- Nếu chúng ta nhập vào $a \neq 0$ thì câu lệnh `printf("Nghich dao cua %f la %f", a, 1/a)` được thực hiện, ngược lại câu lệnh này không được thực hiện.

- Lệnh `getch()` luôn luôn được thực hiện vì nó không phải là “lệnh liền sau” điều kiện if.

Ví dụ 2: Yêu cầu người chạy chương trình nhập vào giá trị của 2 số a và b, nếu a lớn hơn b thì in ra thông báo “Giá trị của a lớn hơn giá trị của b”, sau đó hiển thị giá trị cụ thể của 2 số lên màn hình.

```

#include <stdio.h>
#include <conio.h>
int main ()
{
    int a,b;
    printf("Nhập vào gia trị của 2 so a, b!");
    scanf("%d%d",&a,&b);
    if (a>b)
  
```

```

{
    printf("\n Gia tri cua a lon hon gia tri cua b");
    printf("\n a=%d, b=%d",a,b);
}
getch();
return 0;
}

```

Giải thích:

Nếu chúng ta nhập vào giá trị của a lớn hơn giá trị của b thì khối lệnh:

```

{
    printf("\n Gia tri cua a lon hon gia tri cua b");
    printf("\n a=%d, b=%d",a,b);
}

```

sẽ được thực hiện, ngược lại khối lệnh này không được thực hiện.

II.2. Dạng đầy đủ

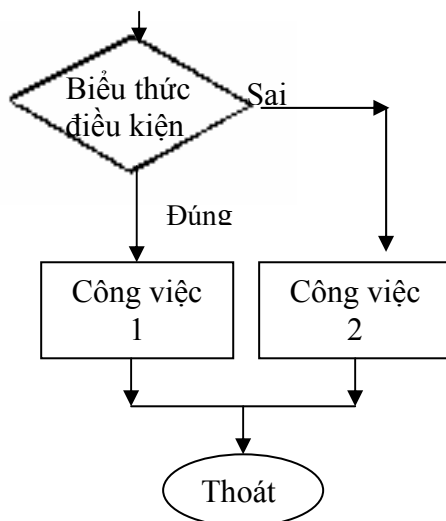
Cú pháp:

```

if (<Biểu thức điều kiện>
    <Công việc 1>
else
    <Công việc 2>

```

Lưu đồ cú pháp:



Giải thích:

Công việc 1, công việc 2 được thể hiện là 1 câu lệnh hay 1 khối lệnh.

Đầu tiên *Biểu thức điều kiện* được kiểm tra trước.

Nếu điều kiện đúng thì thực hiện công việc 1.

Nếu điều kiện sai thì thực hiện công việc 2.

Các lệnh phía sau công việc 2 không phụ thuộc vào điều kiện.

Ví dụ 1: Yêu cầu người thực hiện chương trình nhập vào một số thực a. In ra màn hình kết quả nghịch đảo của a khi $a \neq 0$, khi $a = 0$ in ra thông báo “Không thể tìm được nghịch đảo của a”

```

#include <stdio.h>
#include <conio.h>
int main ()
{
    float a;
    printf("Nhập a = "); scanf("%f",&a);
    if (a !=0 )
        printf("Nghich dao cua %f la %f",a,1/a);
    else
        printf("Khong the tim duoc nghich dao cua a");
}

```

```
    getch();
    return 0;
}
```

Giải thích:

- Nếu chúng ta nhập vào $a \neq 0$ thì câu lệnh `printf("Nghich dao cua %f la %f", a, 1/a)` được thực hiện, ngược lại câu lệnh `printf("Khong the tim duoc nghich dao cua a")` được thực hiện.

- Lệnh `getch()` luôn luôn được thực hiện.

Ví dụ 2: Yêu cầu người chạy chương trình nhập vào giá trị của 2 số a và b, nếu a lớn hơn b thì in ra thông báo “Giá trị của a lớn hơn giá trị của b, giá trị của 2 số”, ngược lại thì in ra màn hình câu thông báo “Giá trị của a nhỏ hơn hoặc bằng giá trị của b, giá trị của 2 số”.

```
#include <stdio.h>
#include <conio.h>
int main ()
{
    int a, b;
    printf("Nhap vao gia tri cua 2 so a va b !");
    scanf("%d%d", &a, &b);
    if (a>b)
    {
        printf("\n a lon hon b");
        printf("\n a=%d b=%d ", a, b);
    }
    else
    {
        printf("\n a nho hon hoac bang b");
        printf("\n a=%d b=%d", a, b);
    }
    printf("\n Thuc hien xong lenh if");
    getch();
    return 0;
}
```

Giải thích:

- Nếu chúng ta nhập vào 40 30 ↵ thì kết quả hiển ra trên màn hình là
a lon hon b
a=40 b=30

Thuc hien xong lenh if

- Còn nếu chúng ta nhập 40 50 ↵ thì kết quả hiển ra trên màn hình là
a nho hon hoac bang b
a=40 b=50

Thuc hien xong lenh if

Ví dụ 3: Yêu cầu người thực hiện chương trình nhập vào một số nguyên dương là tháng trong năm và in ra số ngày của tháng đó.

- Tháng có 31 ngày: 1, 3, 5, 7, 8, 10, 12

- Tháng có 30 ngày: 4, 6, 9, 10

- Tháng có 28 hoặc 29 ngày : 2

```
#include <stdio.h>
#include <conio.h>
```



```

int main ()
{
    int thg;
    printf("Nhap vao thang trong nam !");
    scanf("%d", &thg);
    if (thg==1||thg==3||thg==5||thg==7||thg==8||thg==10||thg==12)
        printf("\n Thang %d co 31 ngay ",thg);
    else if (thg==4||thg==6||thg==9||thg==11)
        printf("\n Thang %d co 30 ngay",thg);
    else if (thg==2)
        printf("\n Thang %d co 28 hoac 29 ngay",thg);
    else printf("Khong co thang %d",thg);
    printf("\n Thuc hien xong lenh if");
    getch();
    return 0;
}

```

Giải thích:

- Nếu chúng ta nhập vào một trong các số 1, 3, 5, 7, 8, 10, 12 thì kết quả xuất hiện trên màn hình sẽ là

Thang <số> co 31 ngay
Thuc hien xong lenh if

- Nếu chúng ta nhập vào một trong các số 4, 6, 9, 11 thì kết quả xuất hiện trên màn hình sẽ là

Thang <số> co 30 ngay
Thuc hien xong lenh if

- Nếu chúng ta nhập vào số 2 thì kết quả xuất hiện trên màn hình sẽ là

Thang 2 co 28 hoac 29 ngay
Thuc hien xong lenh if

- Nếu chúng ta nhập vào số nhỏ hơn 0 hoặc lớn hơn 12 thì kết quả xuất hiện trên màn hình sẽ là

Khong co thang <số>
Thuc hien xong lenh if

Trong đó <số> là con số mà chúng ta đã nhập vào.

Lưu ý:

- Ta có thể sử dụng các câu lệnh if...else lồng nhau. Trong trường hợp if...else lồng nhau thì *else sẽ kết hợp với if gần nhất chưa có else*.

- Trong trường hợp câu lệnh if “bên trong” không có else thì phải viết nó trong cặp dấu {} (coi như là khối lệnh) để tránh sự kết hợp else if sai.

Ví dụ 1:

```

if ( so1>0)
    if (so2 > so3)
        a=so2;
    else /*else của if (so2>so3) */
        a=so3;

```

Ví dụ 2:

```

if (so1>0)
{
    if (so2>so3) /*lệnh if này không có else*/
        a=so2;
}

```

```
else /*else của if (so1>0)*/  
a=so3;
```

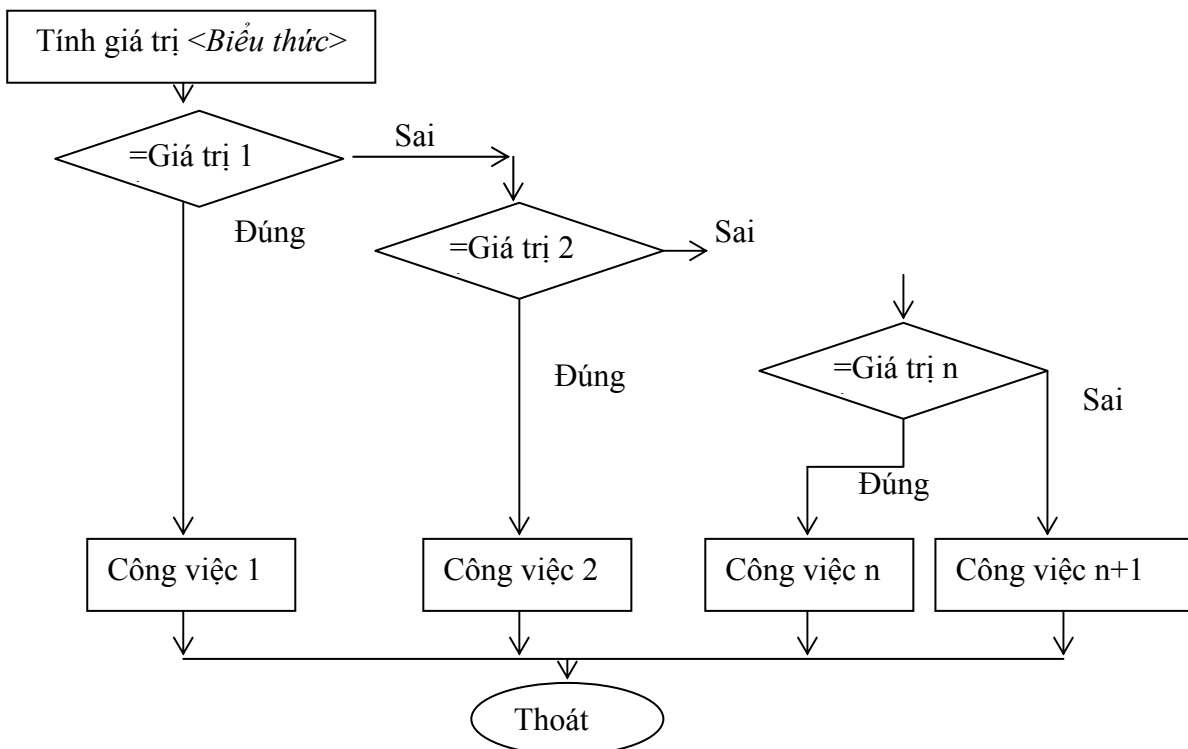
III CẤU TRÚC LỰA CHỌN

Cấu trúc lựa chọn cho phép lựa chọn một trong nhiều trường hợp. Trong C, đó là câu lệnh switch.

Cú pháp:

```
switch (<Biểu thức>  
{  
    case giá trị 1:  
        Khối lệnh thực hiện công việc 1;  
        break;  
    ...  
    case giá trị n:  
        Khối lệnh thực hiện công việc n;  
        break;  
    [default :  
        Khối lệnh thực hiện công việc mặc định;  
        break;]  
}
```

Lưu đồ:



Giải thích:

- Tính giá trị của biểu thức trước.
- Nếu giá trị của biểu thức bằng giá trị 1 thì thực hiện công việc 1 rồi thoát.

- Nếu giá trị của biểu thức khác giá trị 1 thì so sánh với giá trị 2, nếu bằng giá trị 2 thì thực hiện công việc 2 rồi thoát.

- Cứ như thế, so sánh tới giá trị n.

- Nếu tất cả các phép so sánh trên đều sai thì thực hiện công việc mặc định của trường hợp *default*.

Lưu ý:

- Biểu thức trong `switch()` phải có kết quả là giá trị kiểu số nguyên (`int`, `char`, `long`, ...).

- Các giá trị sau `case` cũng phải là kiểu số nguyên.

- Không bắt buộc phải có `default`.

Ví dụ 1: Nhập vào một số nguyên, chia số nguyên này cho 2 lấy phần dư. Kiểm tra nếu phần dư bằng 0 thì in ra thông báo “số chẵn”, nếu số dư bằng 1 thì in thông báo “số lẻ”.

```
#include <stdio.h>
#include <conio.h>
int main ()
{   int songuyen, phandu;
    clrscr();
    printf("\n Nhập vào số nguyên ");
    scanf("%d",&songuyen);
    phandu=(songuyen % 2);
    switch(phandu)
    {
        case 0: printf("%d là số chẵn ",songuyen);
                break;
        case 1: printf("%d là số lẻ ",songuyen);
                break;
    }
    getch();
    return 0;
}
```

Ví dụ 2: Nhập vào 2 số nguyên và 1 phép toán.

- Nếu phép toán là ‘+’, ‘-’, ‘*’ thì in ra kết quả là tổng, hiệu, tích của 2 số.

- Nếu phép toán là ‘/’ thì kiểm tra xem số thứ 2 có khác không hay không? Nếu khác không thì in ra thương của chúng, ngược lại thì in ra thông báo “không chia cho 0”.

```
#include <stdio.h>
#include <conio.h>
int main ()
{   int so1, so2;
    float thuong;
    char pheptoan;
    clrscr();
    printf("\n Nhập vào 2 số nguyên ");
    scanf("%d%d",&so1,&so2);
    fflush(stdin);
    /*Xóa ký tự enter trong vùng đệm trước khi nhập phép toán */
    printf("\n Nhập vào phép toán ");
    scanf("%c",&pheptoan);
    switch(pheptoan)
    {
```

```
    case '+':
        printf("\n %d + %d =%d",so1, so2, so1+so2);
        break;
    case '-':
        printf("\n %d - %d =%d",so1, so2, so1-so2);
        break;
    case '*':
        printf("\n %d * %d =%d",so1, so2, so1*so2);
        break;
    case '/':
        if (so2!=0)
        {
            thuong=float(so1)/float(so2);
            printf("\n %d / %d =%f", so1, so2, thuong);
        }
        else printf("Khong chia duoc cho 0");
        break;
    default :
        printf("\n Chua ho tro phep toan %c", pheptoan);
        break;
}
getch();
return 0;
}
```

Trong ví dụ trên, tại sao phải xóa ký tự trong vùng đệm trước khi nhập phép toán?

Ví dụ 3: Yêu cầu người thực hiện chương trình nhập vào một số nguyên dương là tháng trong năm và in ra số ngày của tháng đó.

- Tháng có 31 ngày: 1, 3, 5, 7, 8, 10, 12
- Tháng có 30 ngày: 4, 6, 9, 10
- Tháng có 28 hoặc 29 ngày : 2
- Nếu nhập vào số <1 hoặc >12 thì in ra câu thông báo “không có tháng này”.

```
#include <stdio.h>
#include<conio.h>
int main ()
{
    int thang;
    clrscr();
    printf("\n Nhập vào thangs 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 co 31 ngay ",thang);
            break;
        case 4:
        case 6:
        case 9:
        case 11:
            printf("\n Tháng %d co 30 ngay ",thang);
            break;
```

```

    case 2:
        printf ("\ Tháng 2 có 28 hoặc 29 ngày");
        break;
    default :
        printf("\n Không có tháng %d", thang);
        break;
}
getch();
return 0;
}

```

Trong ví dụ trên, tại sao phải sử dụng case 1:, case 3:, ...case 12: ?

IV. CẤU TRÚC VÒNG LẶP

Cấu trúc vòng lặp cho phép lặp lại nhiều lần 1 công việc (được thể hiện bằng 1 câu lệnh hay 1 khối lệnh) nào đó cho đến khi thỏa mãn 1 điều kiện cụ thể.

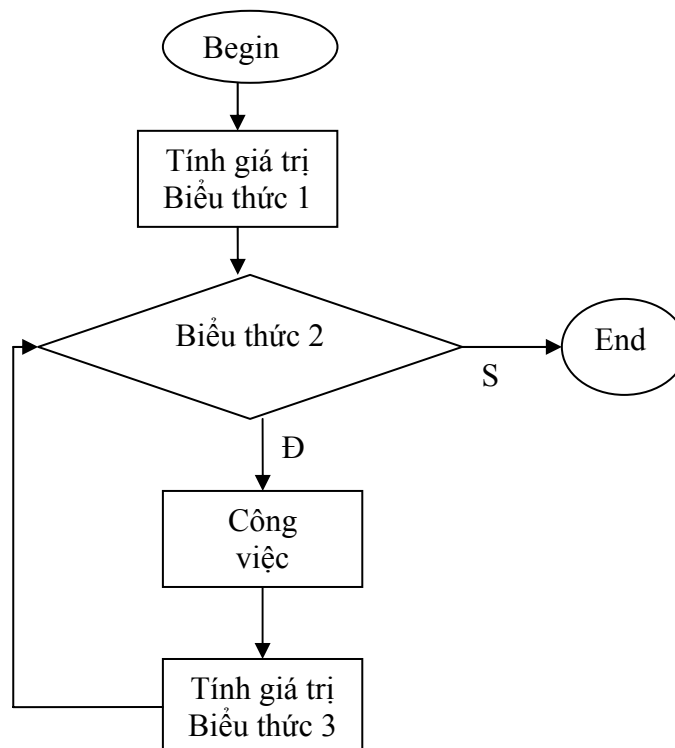
IV.1. Vòng lặp for

Lệnh for cho phép lặp lại công việc cho đến khi điều kiện sai.

Cú pháp:

for (Biểu thức 1; biểu thức 2; biểu thức 3)
<Công việc>

Lưu đồ:



Giải thích:

<Công việc>: được thể hiện là 1 câu lệnh hay 1 khối lệnh. Thứ tự thực hiện của câu lệnh for như sau:

B1: Tính giá trị của biểu thức 1.

B2: Tính giá trị của biểu thức 2.

- Nếu giá trị của biểu thức 2 là sai ($=0$): *thoát khỏi câu lệnh for*.
- Nếu giá trị của biểu thức 2 là đúng ($\neq 0$): <Công việc> được thực hiện.

B3: Tính giá trị của biểu thức 3 và quay lại B2.

Một số lưu ý khi sử dụng câu lệnh for:

- Khi biểu thức 2 vắng mặt thì nó được coi là luôn luôn đúng
- Biểu thức 1: thông thường là một phép gán để khởi tạo giá trị ban đầu cho biến điều kiện.
- Biểu thức 2: là một biểu thức kiểm tra điều kiện đúng sai để dừng vòng lặp.
- Biểu thức 3: thông thường là một phép gán để thay đổi giá trị của biến điều kiện.
- Trong mỗi biểu thức có thể có nhiều biểu thức con. Các biểu thức con được phân biệt bởi dấu phẩy.

Ví dụ 1: Viết đoạn chương trình in dãy số nguyên từ 1 đến 10.

```
#include <stdio.h>
#include <conio.h>
int main ()
{   int i;
    clrscr();
    printf("\n Day so tu 1 den 10 :");
    for (i=1; i<=10; i++)
        printf("%d ", i);
    getch();
    return 0;
}
```

Kết quả chương trình như sau:

```
Day so tu 1 den 10 :1 2 3 4 5 6 7 8 9 10
```

Ví dụ 2: Viết chương trình nhập vào một số nguyên n. Tính tổng của các số nguyên từ 1 đến n.

```
#include <stdio.h>
#include <conio.h>
int main ()
{   unsigned int n,i,tong;
    clrscr();
    printf("\n Nhap vao so nguyen duong n:"); scanf("%d",&n);
    tong=0;
    for (i=1; i<=n; i++)
        tong+=i;
    printf("\n Tong tu 1 den %d =%d ",n,tong);
    getch();
    return 0;
}
```

Nếu chúng ta nhập vào số 9 thì kết quả như sau:

```
Nhap vao so nguyen duong n:9
Tong tu 1 den 9 =45 _
```

Ví dụ 3: Viết chương trình in ra trên màn hình một ma trận có n dòng m cột như sau:

```

1   2   3   4   5   6   7
2   3   4   5   6   7   8
3   4   5   6   7   8   9
...

```

```

#include <stdio.h>
#include <conio.h>
int main ()
{   unsigned int dong, cot, n, m;
    clrscr();
    printf("\n Nhap vao so dong va so cot :");
    scanf("%d%d", &n, &m);
    for (dong=0; dong<n; dong++)
    {
        printf("\n");
        for (cot=1; cot<=m; cot++)
            printf("%d\t", dong+cot);

    }
    getch();
    return 0;
}

```

Kết quả khi nhập 3 dòng 6 cột như sau

```

Nhap vao so dong va so cot :3 6
1   2   3   4   5   6
2   3   4   5   6   7
3   4   5   6   7   8

```

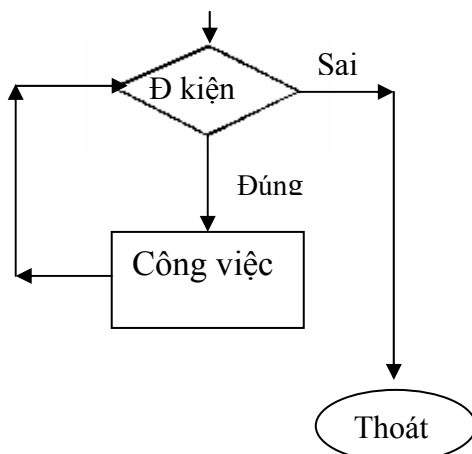
III.2. Vòng lặp while

Vòng lặp while giống như vòng lặp for, dùng để lặp lại một công việc nào đó cho đến khi điều kiện sai.

Cú pháp:

while (*Biểu thức điều kiện*)
 <Công việc>

Lưu đồ:



Giải thích:

- <Công việc>: được thể hiện bằng 1 câu lệnh hay 1 khối lệnh.
- Kiểm tra *Biểu thức điều kiện* trước.
- Nếu điều kiện sai ($=0$) thì thoát khỏi lệnh while.
- Nếu điều kiện đúng ($\neq 0$) thì thực hiện công việc rồi quay lại kiểm tra điều kiện tiếp.

Lưu ý:

- Lệnh while gồm có biểu thức điều kiện và thân vòng lặp (khối lệnh thực hiện công việc)

- Vòng lặp dừng lại khi nào điều kiện sai.

- Khối lệnh thực hiện công việc có thể rỗng, có thể làm thay đổi điều kiện.

Ví dụ 1: Viết đoạn chương trình in dãy số nguyên từ 1 đến 10.

```
#include <stdio.h>
#include<conio.h>
int main ()
{   int i;
    clrscr();
    printf("\n Day so tu 1 den 10 :");
    i=1;
    while (i<=10)
        printf("%d ",i++);
    getch();
    return 0;
}
```

Kết quả chương trình như sau:

```
Day so tu 1 den 10 :1 2 3 4 5 6 7 8 9 10
```

Ví dụ 2: Viết chương trình nhập vào một số nguyên n. Tính tổng của các số nguyên từ 1 đến n.

```
#include <stdio.h>
#include<conio.h>
int main ()
{   unsigned int n,i,tong;
    clrscr();
    printf("\n Nhap vao so nguyen duong n:");
    scanf("%d",&n);
    tong=0;
    i=1;
    while (i<=n)
    {
        tong+=i;
        i++;
    }
    printf("\n Tong tu 1 den %d =%d ",n,tong);
    getch();
    return 0;
}
```

Nếu chúng ta nhập vào số 9 thì kết quả như sau:

```
Nhap vao so nguyen duong n:9
Tong tu 1 den 9 =45 _
```

Ví dụ 3: Viết chương trình in ra trên màn hình một ma trận có n dòng m cột như sau:

1	2	3	4	5	6	7
2	3	4	5	6	7	8
3	4	5	6	7	8	9

...

```

#include <stdio.h>
#include <conio.h>
int main ()
{   unsigned int dong, cot, n, m;
    clrscr();
    printf("\n Nhap vao so dong va so cot :");
    scanf("%d%d", &n, &m);
    dong=0;
    while (dong<n)
    {
        printf("\n");
        cot=1;
        while (cot<=m)
        {
            printf("%d\t", dong+cot);
            cot++;
        }
        dong++;
    }
    getch();
    return 0;
}

```

Kết quả khi nhập 3 dòng 6 cột như sau

```

Nhap vao so dong va so cot :3 6
1      2      3      4      5      6
2      3      4      5      6      7
3      4      5      6      7      8

```

IV.3. Vòng lặp do... while

Vòng lặp do ... while giống như vòng lặp for, while, dùng để lặp lại một công việc nào đó khi điều kiện còn đúng.

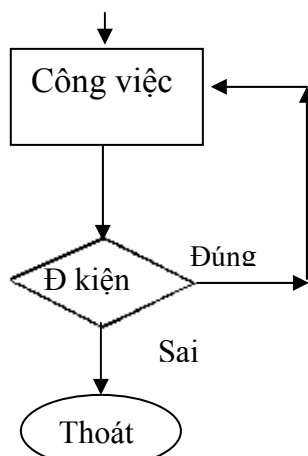
Cú pháp:

```

do
    <Công việc>
while (<Biểu thức điều kiện>)

```

Lưu đồ:



Giải thích:

- <Công việc>: được thể hiện bằng 1 câu lệnh hay 1 khối lệnh.
- Trước tiên công việc được thực hiện trước, sau đó mới kiểm tra *Biểu thức điều kiện*.
- Nếu điều kiện sai thì thoát khỏi lệnh do ...while.
- Nếu điều kiện còn đúng thì thực hiện công việc rồi quay lại kiểm tra điều kiện tiếp.

Lưu ý:

- Lệnh do...while thực hiện công việc ít nhất 1 lần.
- Vòng lặp dừng lại khi điều kiện sai.
- Khối lệnh thực hiện công việc có thể rỗng, có thể làm thay đổi điều kiện.

Ví dụ 1: Viết đoạn chương trình in dãy số nguyên từ 1 đến 10.

```
#include <stdio.h>
#include<conio.h>
int main ()
{   int i;
    clrscr();
    printf("\n Day so tu 1 den 10 :");
    i=1;
    do
        printf("%d ",i++);
    while (i<=10);
    getch();
    return 0;
}
```

Kết quả chương trình như sau:

```
Day so tu 1 den 10 :1 2 3 4 5 6 7 8 9 10
```

Ví dụ 2: Viết chương trình nhập vào một số nguyên n. Tính tổng của các số nguyên từ 1 đến n.

```
#include <stdio.h>
#include<conio.h>
int main ()
{   unsigned int n,i,tong;
    clrscr();
    printf("\n Nhap vao so nguyen duong n:");
    scanf("%d",&n);
    tong=0;
    i=1;
    do
    {
        tong+=i;
        i++;
    } while (i<=n);
    printf("\n Tong tu 1 den %d =%d ",n,tong);
    getch();
    return 0;
}
```

Nếu chúng ta nhập vào số 9 thì kết quả như sau:

```
Nhap vao so nguyen duong n:9
Tong tu 1 den 9 =45 _
```

Ví dụ 3: Viết chương trình in ra trên màn hình một ma trận có n dòng m cột như sau (n, m>=1):

```

1   2   3   4   5   6   7
2   3   4   5   6   7   8
3   4   5   6   7   8   9
...

```

```

#include <stdio.h>
#include <conio.h>
int main ()
{   unsigned int dong, cot, n, m;
    clrscr();
    printf("\n Nhap vao so dong va so cot :");
    scanf("%d%d", &n, &m);
    dong=0;
    do
    {
        printf("\n");
        cot=1;
        do
        {
            printf("%d\t", dong+cot);
            cot++;
        } while (cot<=m);
        dong++;
    } while (dong<n);
    getch();
    return 0;
}

```

Kết quả khi nhập 3 dòng 6 cột như sau

```

Nhập vao so dong va so cot :3 6
1       2       3       4       5       6
2       3       4       5       6       7
3       4       5       6       7       8

```

IV.4. So sánh các vòng lặp

Vòng lặp for, while:

- Kiểm tra điều kiện trước thực hiện công việc sau nên đoạn lệnh thực hiện công việc có thể không được thực hiện .

- Vòng lặp kết thúc khi nào điều kiện sai.

Vòng lặp do...while:

- Thực hiện công việc trước kiểm tra điều kiện sau nên đoạn lệnh thực hiện công việc được thực hiện ít nhất 1 lần.

- Vòng lặp kết thúc khi nào điều kiện sai.

V. CÁC CÂU LỆNH ĐẶC BIỆT

V.1. Lệnh break

Cú pháp: **break**

Dùng để thoát khỏi vòng lặp. Khi gặp câu lệnh này trong vòng lặp, chương trình sẽ thoát ra khỏi vòng lặp và chỉ đến câu lệnh liền sau nó. Nếu nhiều vòng lặp --> break sẽ thoát ra khỏi vòng lặp gần nhất. Ngoài ra, break còn được dùng trong cấu trúc lựa chọn switch.

IV.2. Lệnh continue

Cú pháp: continue

- Khi gặp lệnh này trong các vòng lặp, chương trình sẽ bỏ qua phần còn lại trong vòng lặp và tiếp tục thực hiện lần lặp tiếp theo.

- Đối với lệnh for, *biểu thức 3* sẽ được tính trị và quay lại bước 2.

- Đối với lệnh while, do while; *biểu thức điều kiện* sẽ được tính và xét xem có thể tiếp tục thực hiện <Công việc> nữa hay không? (dựa vào kết quả của *biểu thức điều kiện*).

VI. BÀI TẬP

VI.1 Mục đích yêu cầu

Làm quen và nắm vững các lệnh có cấu trúc của C, biết cách chọn lựa trong trường hợp nào sẽ sử dụng cấu trúc nào. Thực hiện các chương trình trong phần nội dung bằng cách kết hợp các lệnh lặp, các lệnh rẽ nhánh và các lệnh đơn.

VI.2 Nội dung

1. Viết chương trình nhập 3 số từ bàn phím, tìm số lớn nhất trong 3 số đó, in kết quả lên màn hình.

2. Viết chương trình tính chu vi, diện tích của tam giác với yêu cầu sau khi nhập 3 số a, b, c phải kiểm tra lại xem a, b, c có tạo thành một tam giác không? Nếu có thì tính chu vi và diện tích. Nếu không thì in ra câu " Không tạo thành tam giác".

3. Viết chương trình giải phương trình bậc nhất $ax+b=0$ với a, b nhập từ bàn phím.

4. Viết chương trình giải phương trình bậc hai $ax^2+bx + c = 0$ với a, b, c nhập từ bàn phím.

5. Viết chương trình nhập từ bàn phím 2 số a, b và một ký tự ch.

Nếu: ch là "+" thì thực hiện phép tính $a + b$ và in kết quả lên màn hình.

ch là "-" thì thực hiện phép tính $a - b$ và in kết quả lên màn hình.

ch là "*" thì thực hiện phép tính $a * b$ và in kết quả lên màn hình.

ch là "/" thì thực hiện phép tính a / b và in kết quả lên màn hình.

6. Viết chương trình nhập vào 2 số là tháng và năm của một năm. Xét xem tháng đó có bao nhiêu ngày? Biết rằng:

Nếu tháng là 4, 6, 9, 11 thì số ngày là 30.

Nếu tháng là 1, 3, 5, 7, 8, 10, 12 thì số ngày là 31.

Nếu tháng là 2 và năm nhuận thì số ngày 29, ngược lại thì số ngày là 28.

7. Có hai phương thức gửi tiền tiết kiệm: gửi không kỳ hạn lãi suất 2.4%/tháng, mỗi tháng tính lãi một lần, gửi có kỳ hạn 3 tháng lãi suất 4%/tháng, 3 tháng tính lãi một lần.

Viết chương trình tính tổng cộng số tiền cả vốn lẫn lời sau một thời gian gửi nhập từ bàn phím.

8. Một số nguyên dương chia hết cho 3 nếu tổng các chữ số của nó chia hết cho 3. Viết chương trình nhập vào một số có 3 chữ số, kiểm tra số đó có chia hết cho 3 dùng tính chất trên. (if)

9. Trò chơi "Oẳn tù tì": trò chơi có 2 người chơi mỗi người sẽ dùng tay để biểu thị một trong 3 công cụ sau: Kéo, Bao và Búa.

Nguyên tắc: Kéo thắng bao.
Bao thắng búa.
Búa thắng kéo.

Viết chương trình mô phỏng trò chơi này cho hai người chơi và người chơi với máy. (switch)

10. Viết chương trình tính tiền điện gồm các khoản sau:

Tiền thuê bao điện kế : 1000 đồng / tháng.

Định mức sử dụng điện cho mỗi hộ là 50 Kw

Phần định mức tính giá 450 đồng /Kwh

Nếu phần vượt định mức ≤ 50 Kw tính giá phạt cho phần này là 700 đồng/Kwh .

Nếu phần vượt định mức lớn 50 Kw và nhỏ hơn 100Kw tính giá phạt cho phần này là 910 đồng/Kwh

Nếu phần vượt định mức lớn hơn hay bằng 100 Kw tính giá phạt cho phần này là 1200 đồng/Kwh .

Với : chỉ số điện kế cũ và chỉ số điện kế mới nhập vào từ bàn phím. In ra màn hình số tiền trả trong định mức, vượt định mức và tổng của chúng. (if)

11. Viết chương trình nhập vào giờ, phút, giây dạng (hh:mm:ss), từ bàn phím. Cộng thêm một số giây vào và in ra kết quả dưới dạng (hh:mm:ss).

12. Viết chương trình nhập vào ngày tháng năm của một ngày, kiểm tra nó có hợp lệ không.

13. Kiểm tra một ký tự nhập vào thuộc tập hợp nào trong các tập ký tự sau:

Các ký tự chữ hoa: 'A' ... 'Z'

Các ký tự chữ thường: 'a' ... 'z'

Các ký tự chữ số : '0' ... '9'

Các ký tự khác.

14. Hệ thập lục phân dùng 16 ký số bao gồm các ký tự 0 .. 9 và A, B, C, D, E ,F.

Các ký số A, B, C, D, E, F có giá trị tương ứng trong hệ thập phân như sau:

A	10
B	11
C	12
D	13
E	14
F	15

Hãy viết chương trình cho nhập vào ký tự biểu diễn một ký số của hệ thập lục phân và cho biết giá trị thập phân tương ứng. Trường hợp ký tự nhập vào không thuộc các ký số trên, đưa ra thông báo lỗi :

"Hệ thập lục phân không dùng ký số này"

15. Viết chương trình nhập vào ngày tháng năm của ngày hôm nay, in ra ngày tháng năm của ngày mai.

16. Viết chương trình tính các tổng sau:

a) $S=1 + 2 + \dots + n$

b) $S=1/2 + 2/3 + \dots + n/(n+1)$

c) $S= - 1 + 2 - 3 + 4 - \dots + (-1)^n n$

17. Viết chương trình nhập vào một dãy n số, tìm số lớn nhất của dãy và xác định vị trí của số lớn nhất trong dãy.

18. Fibonacci là một dãy số được định nghĩa như sau:

$$F_n = \begin{cases} 1, & n=1 \\ 2, & n=2 \\ F_{n-1} + F_{n-2}, & n > 2 \end{cases}$$

Viết chương trình in ra màn hình dãy Fibonacci có n số hạng, n nhập từ bàn phím khi cho chạy chương trình.

19. Viết chương trình đếm số chữ số của một số nguyên n.

20. Tìm số nguyên dương k nhỏ nhất sao cho $2^k > n$ với n là một số nguyên dương nhập từ bàn phím.

21. Viết chương trình in ra số đảo ngược của một số nguyên n, với n nhập từ bàn phím.

22. Tính giá trị trung bình của một dãy số thực, kết thúc dãy với -1.

23. Viết chương trình mô phỏng phép chia nguyên DIV 2 số nguyên a và b như sau: để chia nguyên a và b ta tính trị a-b, sau đó lấy hiệu tìm được lại trừ cho b... tiếp tục cho đến khi hiệu của nó nhỏ hơn b. Số lần thực hiện được các phép trừ ở trên sẽ bằng trị của phép chia nguyên.

24. Tìm số nguyên dương N nhỏ nhất sao cho

$$1+1/2+ \dots+1/N > S, \text{ với } S \text{ nhập từ bàn phím.}$$

25. Viết chương trình tính $P=2*4*6* \dots *(2n)$, n nhập từ bàn phím.

26. Viết chương trình tìm UCLN và BCNN của hai số a và b theo thuật toán sau (Ký hiệu UCLN của a, b là (a,b) còn BCNN là [a,b])

- Nếu a chia hết cho b thì $(a,b) = b$

- Nếu $a = b*q + r$ thì $(a,b) = (b,r)$

- $[a,b] = a*b/(b,r)$

27. Viết chương trình nhập vào một số nguyên dương n, in ra màn hình các số nguyên tố $p \leq n$. Số nguyên p gọi là số nguyên tố nếu p chỉ chia hết cho một và chia hết cho bản thân nó.

28. Viết chương trình tính gần đúng căn bậc hai của một số dương a theo phương pháp Newton : Trước hết cho $x_0=(1 + a)/2$ sau đó là công thức truy hồi: $x_{n+1}=(x_n + a/x_n)/2$

$$\text{Nếu: } \left| \frac{x_{n+1} - x_n}{x_n} \right| < \epsilon \text{ thì căn bậc hai của } a \text{ bằng } x_{n+1}$$

Trong đó ϵ là một hằng số cho trước làm độ chính xác.

29. Viết chương trình tính gần đúng căn bậc n của một số dương a theo phương pháp Newton : Trước hết cho $x_0 = a/n$ sau đó là công thức truy hồi:

$$x_{k+1} = \left| \frac{(n-1)x_k^n + a}{nx_k^{n-1}} \right|$$

Nếu $|a - x_n^n| < \epsilon$ thì x_n là căn bậc n của a . Trong đó ϵ là một hằng số cho trước làm độ chính xác. Nếu $a < 0$ và n chẵn thì không tồn tại căn.

Chương 5

CHƯƠNG TRÌNH CON

Học xong chương này, sinh viên sẽ nắm được các vấn đề sau:

- *Khái niệm về hàm (function) trong C.*
- *Cách xây dựng và cách sử dụng hàm trong C.*

I. KHÁI NIỆM VỀ HÀM TRONG C

Trong những chương trình lớn, có thể có những đoạn chương trình viết lặp đi lặp lại nhiều lần, để tránh rườm rà và mất thời gian khi viết chương trình; người ta thường phân chia chương trình thành nhiều module, mỗi module giải quyết một công việc nào đó. Các module như vậy gọi là các chương trình con.

Một tiện lợi khác của việc sử dụng chương trình con là ta có thể dễ dàng kiểm tra xác định tính đúng đắn của nó trước khi ráp nối vào chương trình chính và do đó việc xác định sai sót để tiến hành hiệu đính trong chương trình chính sẽ thuận lợi hơn.

Trong C, chương trình con được gọi là hàm. Hàm trong C có thể trả về kết quả thông qua tên hàm hay có thể không trả về kết quả.

Hàm có hai loại: hàm chuẩn và hàm tự định nghĩa. Trong chương này, ta chú trọng đến cách định nghĩa hàm và cách sử dụng các hàm đó.

Một hàm khi được định nghĩa thì có thể sử dụng bất cứ đâu trong chương trình. Trong C, một chương trình bắt đầu thực thi bằng hàm main.

Ví dụ 1: Ta có hàm max để tìm số lớn giữa 2 số nguyên a, b như sau:

```
int max(int a, int b)
{
    return (a>b) ? a:b;
}
```

Ví dụ 2: Ta có chương trình chính (hàm main) dùng để nhập vào 2 số nguyên a,b và in ra màn hình số lớn trong 2 số

```
#include <stdio.h>
#include <conio.h>
int max(int a, int b)
{
    return (a>b) ? a:b;
}

int main()
{
    int a, b, c;
    printf("\n Nhập vào 3 số a, b, c ");
    scanf("%d%d%d", &a, &b, &c);
    printf("\n Số lớn là %d", max(a, max(b, c)));
    getch();
    return 0;
}
```


I.1. Hàm thư viện

Hàm thư viện là những hàm đã được định nghĩa sẵn trong một thư viện nào đó, muốn sử dụng các hàm thư viện thì phải khai báo thư viện trước khi sử dụng bằng lệnh `#include <tên thư viện.h>`

Một số thư viện:

alloc.h	assert.h	bcd.h	bios.h	complex.h
conio.h	ctype.h	dir.h	dirent.h	dos.h
errno.h	fcntl.h	float.h	fstream.h	grneric.h
graphics.h	io.h	iomanip.h	iostream.h	limits.h
locale.h	malloc.h	math.h	mem.h	process.h
setjmp.h	share.h	signal.h	stdarg.h	stddef.h
stdio.h	stdiostr.h	stdlib.h	stream.h	string.h
strstrea.h	sys\stat.h	sys\timeb.h	sys\types.h	time.h
values.h				

Ý nghĩa của một số thư viện thường dùng:

1. stdio.h : Thư viện chứa các hàm vào/ ra chuẩn (standard input/output). Gồm các hàm **printf()**, **scanf()**, **getc()**, **putc()**, **gets()**, **puts()**, **fflush()**, **fopen()**, **fclose()**, **fread()**, **fwrite()**, **getchar()**, **putchar()**, **getw()**, **putw()**...

2. conio.h : Thư viện chứa các hàm vào ra trong chế độ DOS (DOS console). Gồm các hàm **clrscr()**, **getch()**, **getche()**, **getpass()**, **cgets()**, **cputs()**, **putch()**, **clreol()**,...

3. math.h: Thư viện chứa các hàm tính toán gồm các hàm **abs()**, **sqrt()**, **log()**, **log10()**, **sin()**, **cos()**, **tan()**, **acos()**, **asin()**, **atan()**, **pow()**, **exp()**,...

4. alloc.h: Thư viện chứa các hàm liên quan đến việc quản lý bộ nhớ. Gồm các hàm **calloc()**, **realloc()**, **malloc()**, **free()**, **farmalloc()**, **farcalloc()**, **farfree()**, ...

5. io.h: Thư viện chứa các hàm vào ra cấp thấp. Gồm các hàm **open()**, **_open()**, **read()**, **_read()**, **close()**, **_close()**, **creat()**, **_creat()**, **creatnew()**, **eof()**, **filelength()**, **lock()**,...

6. graphics.h: Thư viện chứa các hàm liên quan đến đồ họa. Gồm **initgraph()**, **line()**, **circle()**, **putpixel()**, **getpixel()**, **setcolor()**, ...

...

Muốn sử dụng các hàm thư viện thì ta phải xem cú pháp của các hàm và sử dụng theo đúng cú pháp (xem trong phần trợ giúp của Turbo C).

I.2. Hàm người dùng

Hàm người dùng là những hàm do người lập trình tự tạo ra nhằm đáp ứng nhu cầu xử lý của mình.

II. XÂY DỰNG MỘT HÀM

II.1 Định nghĩa hàm

Cấu trúc của một hàm tự thiết kế:

```
<kiểu kết quả> Tên hàm ([<kiểu t số> <tham số>][,<kiểu t số><tham số>][...])
{
    [Khai báo biến cục bộ và các câu lệnh thực hiện hàm]
    [return [<Biểu thức>];]
}
```

Giải thích:

- *Kiểu kết quả*: là kiểu dữ liệu của kết quả trả về, có thể là : int, byte, char, float, void... Một hàm có thể có hoặc không có kết quả trả về. Trong trường hợp *hàm không có kết quả trả về ta nên sử dụng kiểu kết quả là void*.

- *Kiểu t số*: là kiểu dữ liệu của tham số.

- *Tham số*: là tham số truyền dữ liệu vào cho hàm, một hàm có thể có hoặc không có tham số. *Tham số này gọi là tham số hình thức, khi gọi hàm chúng ta phải truyền cho nó các tham số thực tế*. Nếu có nhiều tham số, mỗi tham số phân cách nhau dấu phẩy (,).

- Bên trong thân hàm (phần giới hạn bởi cặp dấu {}) là các khai báo cùng các câu lệnh xử lý. Các khai báo bên trong hàm được gọi là các khai báo cục bộ trong hàm và các khai báo này chỉ tồn tại bên trong hàm mà thôi.

- Khi định nghĩa hàm, ta thường sử dụng câu lệnh return để trả về kết quả thông qua tên hàm.

Lệnh return dùng để thoát khỏi một hàm và có thể trả về một giá trị nào đó.

Cú pháp:

```
return ; /*không trả về giá trị*/
return <biểu thức>; /*Trả về giá trị của biểu thức*/
return (<biểu thức>); /*Trả về giá trị của biểu thức*/
```

Nếu hàm có kết quả trả về, ta bắt buộc phải sử dụng câu lệnh return để trả về kết quả cho hàm.

Ví dụ 1: Viết hàm tìm số lớn giữa 2 số nguyên a và b

```
int max(int a, int b)
{
    return (a>b) ? a:b;
}
```

Ví dụ 2: Viết hàm tìm ước chung lớn nhất giữa 2 số nguyên a, b. Cách tìm: đầu tiên ta giả sử UCLN của hai số là số nhỏ nhất trong hai số đó. Nếu điều đó không đúng thì ta giảm đi một đơn vị và cứ giảm như vậy cho tới khi nào tìm thấy UCLN

```
int ucln(int a, int b)
{
    int u;
    if (a<b)
```

```
        u=a;
    else
        u=b;
    while ((a%u !=0) || (b%u!=0))
        u--;
    return u;
}
```

II.2 Sử dụng hàm

Một hàm khi định nghĩa thì chúng vẫn chưa được thực thi trừ khi ta có một lời gọi đến hàm đó.

Cú pháp gọi hàm: <Tên hàm>([Danh sách các tham số])

Ví dụ: Viết chương trình cho phép tìm ước số chung lớn nhất của hai số tự nhiên.

```
#include<stdio.h>
unsigned int ucln(unsigned int a, unsigned int b)
{
    unsigned int u;
    if (a<b)
        u=a;
    else
        u=b;
    while ((a%u !=0) || (b%u!=0))
        u--;
    return u;
}
int main()
{
    unsigned int a, b, UC;
    printf("Nhap a,b: ");scanf("%d%d",&a,&b);
    UC = ucln(a,b);
    printf("Uoc chung lon nhat la: ", UC);
    return 0;
}
```

Lưu ý: Việc gọi hàm là một phép toán, không phải là một phát biểu.

II.3 Nguyên tắc hoạt động của 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.

- Khi gặp lệnh return hoặc dấu } cuối cùng trong thân hàm, chương trình sẽ thoát khỏi hàm để trở về chương trình gọi nó và thực hiện tiếp tục những câu lệnh của chương trình này.

III. TRUYỀN THAM SỐ CHO HÀM

Mặc nhiên, việc truyền tham số cho hàm trong C là truyền theo giá trị; nghĩa là các giá trị thực (tham số thực) không bị thay đổi giá trị khi truyền cho các tham số hình thức

Ví dụ 1: Giả sử ta muốn in ra nhiều dòng, mỗi dòng 50 ký tự nào đó. Để đơn giản ta viết một hàm, nhiệm vụ của hàm này là in ra trên một dòng 50 ký tự nào đó. Hàm này có tên là InKT.

```
#include <stdio.h>
#include <conio.h>
void InKT(char ch)
{
    int i;
    for(i=1;i<=50;i++) printf("%c",ch);
    printf("\n");
}
int main()
{
    char c = 'A';
    InKT('*'); /* In ra 50 dau * */
    InKT('+');
    InKT(c);
    return 0;
}
```

Lưu ý:

- Trong hàm InKT ở trên, biến ch gọi là tham số hình thức được truyền bằng giá trị (gọi là tham trị của hàm). Các tham trị của hàm coi như là một biến cục bộ trong hàm và chúng được sử dụng như là dữ liệu đầu vào của hàm.

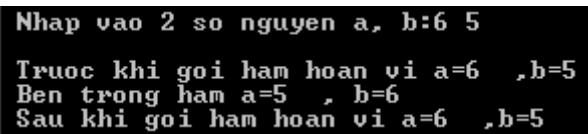
- Khi chương trình con được gọi để thi hành, tham trị được cấp ô nhớ và nhận giá trị là bản sao giá trị của tham số thực. Do đó, mặc dù tham trị cũng là biến, nhưng việc thay đổi giá trị của chúng không có ý nghĩa gì đối với bên ngoài hàm, không ảnh hưởng đến chương trình chính, nghĩa là không làm ảnh hưởng đến tham số thực tương ứng.

Ví dụ 2: Ta xét chương trình sau đây:

```
#include <stdio.h>
#include <conio.h>
int hoanvi(int a, int b)
{
    int t;
    t=a; /*Đoạn này hoán vị giá trị của 2 biến a, b*/
    a=b;
    b=t;
    printf("\nBen trong ham a=%d , b=%d",a,b);
    return 0;
}
```

```
int main()
{
    int a, b;
    clrscr();
    printf("\n Nhập vào 2 số nguyên a, b:");
    scanf("%d%d",&a,&b);
    printf("\n Trước khi gọi hàm hoán vị a=%d ,b=%d",a,b);
    hoanvi(a,b);
    printf("\n Sau khi gọi hàm hoán vị a=%d ,b=%d",a,b);
    getch();
    return 0;
}
```

Kết quả thực hiện chương trình:



```
Nhap vào 2 số nguyên a, b:6 5
Trước khi gọi hàm hoán vị a=6 ,b=5
Bên trong hàm a=5 , b=6
Sau khi gọi hàm hoán vị a=6 ,b=5
```

Giải thích:

- Nhập vào 2 số 6 và 5 (a=6, b=5)
- Trước khi gọi hàm hoán vị thì a=6, b=5
- Bên trong hàm hoán vị a=5, b=6
- Khi ra khỏi hàm hoán vị thì a=6, b=5

* Lưu ý

Trong đoạn chương trình trên, nếu ta muốn sau khi kết thúc chương trình con giá trị của a, b thay đổi thì ta phải đặt tham số hình thức là các con trỏ, còn tham số thực tế là địa chỉ của các biến.

Lúc này mọi sự thay đổi trên vùng nhớ được quản lý bởi con trỏ là các tham số hình thức của hàm thì sẽ ảnh hưởng đến vùng nhớ đang được quản lý bởi tham số thực tế tương ứng (cần để ý rằng vùng nhớ này chính là các biến ta cần thay đổi giá trị).

Người ta thường áp dụng cách này đối với các dữ liệu đầu ra của hàm.

Ví dụ: Xét chương trình sau đây:

```
#include <stdio.h>
#include <conio.h>
long hoanvi(long *a, long *b)
/* Khai báo tham số hình thức *a, *b là các con trỏ kiểu long */
{
    long t;
    t=*a;      /*gán nội dung của x cho t*/
    *a=*b;     /*Gán nội dung của b cho a*/
    *b=t;      /*Gán nội dung của t cho b*/
    printf("\n Bên trong hàm a=%ld , b=%ld",*a,*b);
    /*In ra nội dung của a, b*/
    return 0;
}

int main()
{
    long a, b;
    clrscr();
```

```

printf("\n Nhập vào 2 số nguyên a, b:");
scanf("%ld%ld",&a,&b);
printf("\n Trước khi gọi hàm hoan vị a=%ld ,b=%ld",a,b);
hoanvi(&a,&b); /* Phải là địa chỉ của a và b */
printf("\n Sau khi gọi hàm hoan vị a=%ld ,b=%ld",a,b);
getch();
return 0;
}

```

Kết quả thực hiện chương trình:

```

Nhập vào 2 số nguyên a, b: 5 6
Trước khi gọi hàm hoan vị a=5 ,b=6
Bên trong hàm a=6 , b=5
Sau khi gọi hàm hoan vị a=6 ,b=5

```

Giải thích:

- Nhập vào 2 số 5, 6 (a=5, b=6)
- Trước khi gọi hàm hoanvi thì a=5, b=6
- Trong hàm hoanvi (khi đã hoán vị) thì a=6, b=5
- Khi ra khỏi hàm hoán vị thì a=6, b=6

Lưu ý: Kiểu con trỏ và các phép toán trên biến kiểu con trỏ sẽ nói trong phần sau.

IV. HÀM ĐỆ QUY

IV.1. Định nghĩa

Một hàm được gọi là đệ quy nếu bên trong thân hàm có lệnh gọi đến chính nó.

Ví dụ: Người ta định nghĩa giai thừa của một số nguyên dương n như sau:

$$n! = 1 * 2 * 3 * \dots * (n-1) * n = (n-1)! * n \quad (\text{với } 0! = 1)$$

Như vậy, để tính n! ta thấy nếu n=0 thì n!=1 ngược lại thì n!=n * (n-1)!

Với định nghĩa trên thì hàm đệ quy tính n! được viết:

```

#include <stdio.h>
#include <conio.h>
/*Hàm tính n! bằng đệ quy*/
unsigned int giaiithua_dequy(int n)
{
    if (n==0)
        return 1;
    else
        return n*giaiithua_dequy(n-1);
}
/*Hàm tính n! không đệ quy*/
unsigned int giaiithua_khongdequy(int n)
{
    unsigned int kq,i;
    kq=1;
    for (i=2;i<=n;i++)
        kq=kq*i;
    return kq;
}

```

```
int main()
{
    int n;
    clrscr();
    printf("\n Nhập số n cần tính giai thừa ");
    scanf("%d", &n);
    printf("\nGoi ham de quy: %d != %u", n, giai thua_de quy(n));
    printf("\nGoi ham khong de quy: %d != %u",
           n, giai thua_khong de quy(n));
    getch();
    return 0;
}
```

IV.2. Đặc điểm cần lưu ý khi viết hàm đệ quy

- Hàm đệ quy phải có 2 phần:

- Phần dừng hay phải có trường hợp nguyên tố. Trong ví dụ ở trên thì trường hợp $n=0$ là trường hợp nguyên tố.

- Phần đệ quy: là phần có gọi lại hàm đang được định nghĩa. Trong ví dụ trên thì phần đệ quy là $n > 0$ thì $n! = n * (n-1)!$

- Sử dụng hàm đệ quy trong chương trình sẽ làm chương trình dễ đọc, dễ hiểu và vấn đề được nêu bật rõ ràng hơn. Tuy nhiên trong đa số trường hợp thì hàm đệ quy tốn bộ nhớ nhiều hơn và tốc độ thực hiện chương trình chậm hơn không đệ quy.

- Tùy từng bài có cụ thể mà người lập trình quyết định có nên dùng đệ quy hay không (có những trường hợp không dùng đệ quy thì không giải quyết được bài toán).

V. BÀI TẬP

V.1 Mục đích yêu cầu

Mục đích của việc sử dụng hàm là làm cho chương trình viết ra được sáng sủa, ngắn gọn. Vì thế sinh viên phải nắm vững cách định nghĩa các hàm và cách dùng chúng. Kết hợp các phần đã học trong các chương trước để viết các chương trình con.

V.2 Nội dung

1. Viết hàm tìm số lớn nhất trong hai số. Áp dụng tìm số lớn nhất trong ba số a, b, c với a, b, c nhập từ bàn phím.
2. Viết hàm tìm UCLN của hai số a và b. Áp dụng: nhập vào tử và mẫu số của một phân số, kiểm tra xem phân số đó đã tối giản hay chưa.
3. Viết hàm in n ký tự c trên một dòng. Viết chương trình cho nhập 5 số nguyên cho biết số lượng hàng bán được của mặt hàng A ở 5 cửa hàng khác nhau. Dùng hàm trên vẽ biểu đồ so sánh 5 giá trị đó, mỗi trị dùng một ký tự riêng.
4. Viết một hàm tính tổng các chữ số của một số nguyên. Viết chương trình nhập vào một số nguyên, dùng hàm trên kiểm tra xem số đó có chia hết cho 3 không. Một số chia hết cho 3 khi tổng các chữ số của nó chia hết cho 3.

5. Tam giác Pascal là một bảng số, trong đó hàng thứ 0 bằng 1, mỗi một số hạng của hàng thứ $n+1$ là một tổ hợp chập k của n ($C_n^k = \frac{n!}{k!(n-k)!}$)

Tam giác Pascal có dạng sau:

```

1                ( hàng 0 )
1 1              ( hàng 1 )
1 2 1           ( hàng 2 )
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1 ( hàng 6 )
    
```

.....
 Viết chương trình in lên màn hình tam giác Pascal có n hàng (n nhập vào khi chạy chương trình) bằng cách tạo hai hàm tính giai thừa và tính tổ hợp.

6. Yêu cầu như câu 5 nhưng dựa vào tính chất sau của tổ hợp: $C_n^k = C_{n-1}^{k-1} + C_{n-1}^k$ để hình thành thuật toán là: tạo một hàm tổ hợp có hai biến n, k mang tính đệ quy như sau:

$$\text{ToHop}(n,k) = \begin{cases} 1 & \text{nếu } k=0 \text{ hoặc } k=n \\ \text{ToHop}(n-1,k-1) + \text{ToHop}(n-1,k) & \text{nếu } 1 < k < n \end{cases}$$

7. Viết chương trình tính các tổng sau:

- a) $S = 1 + x + x^2 + x^3 + \dots + x^n$
- b) $S = 1 - x + x^2 - x^3 + \dots (-1)^n x^n$
- c) $S = 1 + x/1! + x^2/2! + x^3/3! + \dots + x^n/n!$

Trong đó n là một số nguyên dương và x là một số bất kỳ được nhập từ bàn phím khi chạy chương trình.

8. Viết chương trình in dãy Fibonacci đã nêu trong bảng phương pháp dùng một hàm Fibonacci F có tính đệ quy.

$$F_n = \begin{cases} 1, & n \text{ chẵn} = 1 \\ 2, & n \text{ chẵn} = 2 \\ F_{n-1} + F_{n-2} \end{cases}$$

9. Bài toán tháp Hà Nội: Có một cái tháp gồm n tầng, tầng trên nhỏ hơn tầng dưới (hình vẽ). Hãy tìm cách chuyển cái tháp này từ vị trí thứ nhất sang vị trí thứ hai thông qua vị trí trung gian thứ ba. Biết rằng chỉ được chuyển mỗi lần một tầng và không được để tầng lớn trên tầng nhỏ.



10. Viết chương trình phân tích một số nguyên dương ra thừa số nguyên tố.

Chương VI

Kiểu mảng

Học xong chương này, sinh viên sẽ nắm được các vấn đề sau:

- Khái niệm về kiểu dữ liệu mảng cũng như ứng dụng của nó.
- Cách khai báo biến kiểu mảng và các phép toán trên các phần tử của mảng.

I. GIỚI THIỆU KIỂU DỮ LIỆU “KIỂU MẢNG” TRONG C

Mảng là một tập hợp các phần tử cố định có cùng một kiểu, gọi là kiểu phần tử. Kiểu phần tử có thể là có các kiểu bất kỳ: ký tự, số, chuỗi ký tự...; cũng có khi ta sử dụng kiểu mảng để làm kiểu phần tử cho một mảng (trong trường hợp này ta gọi là mảng của mảng hay mảng nhiều chiều).

Ta có thể chia mảng làm 2 loại: mảng 1 chiều và mảng nhiều chiều.

Mảng là kiểu dữ liệu được sử dụng rất thường xuyên. Chẳng hạn người ta cần quản lý một danh sách họ và tên của khoảng 100 sinh viên trong một lớp. Nhận thấy rằng mỗi họ và tên để lưu trữ ta cần 1 biến kiểu chuỗi, như vậy 100 họ và tên thì cần khai báo 100 biến kiểu chuỗi. Nếu khai báo như thế này thì đoạn khai báo cũng như các thao tác trên các họ tên sẽ rất dài dòng và rắc rối. Vì thế, kiểu dữ liệu mảng giúp ích ta trong trường hợp này; chỉ cần khai báo 1 biến, biến này có thể coi như là tương đương với 100 biến chuỗi ký tự; đó là 1 mảng mà các phần tử của nó là chuỗi ký tự. Hay như để lưu trữ các từ khóa của ngôn ngữ lập trình C, ta cũng dùng đến một mảng để lưu trữ chúng.

II. MẢNG 1 CHIỀU

Nếu xét dưới góc độ toán học, mảng 1 chiều giống như một vector. Mỗi phần tử của mảng một chiều có giá trị *không phải là một mảng khác*.

II.1. Khai báo

II.1.1. Khai báo mảng với số phần tử xác định (khai báo tường minh)

Cú pháp: <Kiểu> <Tên mảng> [<số phần tử>]

Ý nghĩa:

- **Tên mảng:** đây là một cái tên đặt đúng theo quy tắc đặt tên của danh biểu. Tên này cũng mang ý nghĩa là tên biến mảng.

- **Số phần tử:** là một hằng số nguyên, cho biết số lượng phần tử tối đa trong mảng là bao nhiêu (hay nói khác đi kích thước của mảng là gì).

- **Kiểu:** mỗi phần tử của mảng có dữ liệu thuộc kiểu gì.

- Ở đây, ta khai báo một biến mảng gồm có **số phần tử** phần tử, phần tử thứ nhất là **tên mảng** [0], phần tử cuối cùng là **tên mảng**[**số phần tử** -1]

Ví dụ:

```
int a[10]; /* Khai báo mảng tên a, phần tử thứ nhất là a[0], phần tử cuối cùng là a[9].*/
```

Ta có thể coi mảng a là một dãy liên tiếp các phần tử trong bộ nhớ như sau:

Vị trí	0	1	2	3	4	5	6	7	8	9
Tên phần tử	a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]

Hình 1: Hình ảnh mảng a trong bộ nhớ

II.1.2. Khai báo mảng với số phần tử không xác định (khai báo không tường minh)

Cú pháp: <Kiểu> <Tên mảng> <[]>

Khi khai báo, không cho biết rõ số phần tử của mảng, kiểu khai báo này thường được áp dụng trong các trường hợp: vừa khai báo vừa gán giá trị, khai báo mảng là tham số hình thức của hàm.

a. Vừa khai báo vừa gán giá trị

Cú pháp:

<Kiểu> <Tên mảng> [= {Các giá trị cách nhau bởi dấu phẩy}]

Nếu vừa khai báo vừa gán giá trị thì mặc nhiên C sẽ hiểu số phần tử của mảng là số giá trị mà chúng ta gán cho mảng trong cặp dấu {}. Chúng ta có thể sử dụng hàm `sizeof()` để lấy số phần tử của mảng như sau:

Số phần tử = `sizeof(tên mảng) / sizeof(kiểu)`

b. Khai báo mảng là tham số hình thức của hàm, trong trường hợp này ta không cần chỉ định số phần tử của mảng là bao nhiêu.

II.2 Truy xuất từng phần tử của mảng

Mỗi phần tử của mảng được truy xuất thông qua **Tên biến mảng** theo sau là **chỉ số** nằm trong cặp **dấu ngoặc vuông []**. Chẳng hạn `a[0]` là phần tử đầu tiên của mảng a được khai báo ở trên. Chỉ số của phần tử mảng là một biểu thức mà giá trị là kiểu số nguyên.

Với cách truy xuất theo kiểu này, **Tên biến mảng[Chỉ số]** có thể coi như là một biến có kiểu dữ liệu là **kiểu** được chỉ ra trong khai báo biến mảng.

Ví dụ 1:

```
int a[10];
```

Trong khai báo này, việc truy xuất các phần tử được chỉ ra trong hình 1. Chẳng hạn phần tử thứ 2 (có vị trí 1) là `a[1]`...

Ví dụ 2: Vừa khai báo vừa gán giá trị cho 1 mảng 1 chiều các số nguyên. In mảng số nguyên này lên màn hình.

Giả sử ta đã biết số phần tử của mảng là n; việc hiển thị 1 giá trị số nguyên lên màn hình ta cần sử dụng hàm `printf()` với định dạng `%d`, tổng quát hóa lên nếu muốn hiển thị lên màn hình giá trị của n số nguyên, ta cần gọi hàm `printf()` đúng n lần. Như vậy trong trường hợp này ta sử dụng 1 vòng lặp để in ra giá trị các phần tử.

Ta có đoạn chương trình sau:

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int n, i, j, tam;
    int dayso[] = {66, 65, 69, 68, 67, 70};
    clrscr();
```

```
n=sizeof(dayso)/sizeof(int); /*Lấy số phần tử*/
printf("\n Noi dung cua mang ");
for (i=0;i<n;i++)
    printf("%d ",dayso[i]);
return 0;
}
```

Ví dụ 3: Đổi một số nguyên dương thập phân thành số nhị phân. Việc chuyển đổi này được thực hiện bằng cách lấy số đó chia liên tiếp cho 2 cho tới khi bằng 0 và lấy các số dư theo chiều ngược lại để tạo thành số nhị phân. Ta sẽ dùng mảng một chiều để lưu lại các số dư đó. Chương trình cụ thể như sau:

```
#include<conio.h>
#include<stdio.h>

int main()
{
    unsigned int N;
    unsigned int Du;
    unsigned int NhiPhan[20],K=0,i;
    printf("Nhap vao so nguyen N= ");scanf("%d",&N);
    do
    {
        Du=N % 2;
        NhiPhan[K]=Du; /* Lưu số dư vào mảng ở vị trí K*/
        K++; /* Tăng K lên để lần kế lưu vào vị trí kế*/
        N = N/2;
    } while (N>0);

    printf("Dang nhi phan la: ");
    for(i=K-1;i>=0;i--)
        printf("%d",NhiPhan[i]);
    getch();
    return 0;
}
```

Ví dụ 4: Nhập vào một dãy n số và sắp xếp các số theo thứ tự tăng. Đây là một bài toán có ứng dụng rộng rãi trong nhiều lĩnh vực. Có rất nhiều giải thuật sắp xếp. Một trong số đó được mô tả như sau:

Đầu tiên đưa phần tử thứ nhất so sánh với các phần tử còn lại, nếu nó lớn hơn một phần tử đang so sánh thì đổi chỗ hai phần tử cho nhau. Sau đó tiếp tục so sánh phần tử thứ hai với các phần tử từ thứ ba trở đi ... cứ tiếp tục như vậy cho đến phần tử thứ n-1.

Chương trình sẽ được chia thành các hàm Nhập (Nhập các số), SapXep (Sắp xếp) và InMang (In các số); các tham số hình thức của các hàm này là 1 mảng không chỉ định rõ số phần tử tối đa, nhưng ta cần có *thêm số phần tử thực tế được sử dụng của mảng là bao nhiêu*, đây là một giá trị nguyên.

```
#include<conio.h>
#include<stdio.h>
voidNhap(int a[],int N)
{
    int i;
    for(i=0; i< N; i++)
    {
        printf("Phan tu thu %d: ",i);scanf("%d",&a[i]);
    }
}
```

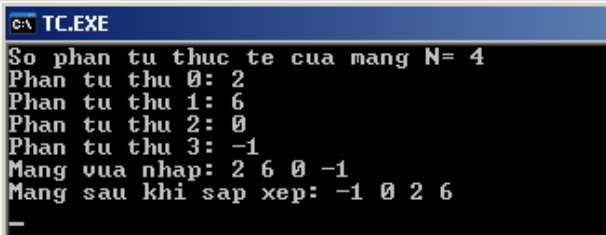
```

void InMang(int a[], int N)
{
    int i;
    for (i=0; i<N;i++)
        printf("%d ",a[i]);
    printf("\n");
}
void SapXep(int a[], int N)
{
    int t,i;
    for(i=0;i<N-1;i++)
        for(int j=i+1;j<N;j++)
            if (a[i]>a[j])
            {
                t=a[i];
                a[i]=a[j];
                a[j]=t;
            }
}

int main()
{
    int b[20], N;
    printf("So phan tu thuc te cua mang N= ");
    scanf("%d",&N);
   Nhap(b,N);
    printf("Mang vua nhap: ");
    InMang(b,N);
    SapXep(b,N); /* Gõi hàm sắp xếp*/
    printf("Mang sau khi sap xep: ");
    InMang(b,N);
    getch();
    return 0;
}

```

Kết quả chạy chương trình có thể là:



```

c:\ TC.EXE
So phan tu thuc te cua mang N= 4
Phan tu thu 0: 2
Phan tu thu 1: 6
Phan tu thu 2: 0
Phan tu thu 3: -1
Mang vua nhap: 2 6 0 -1
Mang sau khi sap xep: -1 0 2 6

```

III. MẢNG NHIỀU CHIỀU

Mảng nhiều chiều là mảng có từ 2 chiều trở lên. Điều đó có nghĩa là mỗi phần tử của mảng là một mảng khác.

Người ta thường sử dụng mảng nhiều chiều để lưu các ma trận, các tọa độ 2 chiều, 3 chiều...

Phần dưới đây là các vấn đề liên quan đến mảng 2 chiều; các mảng 3, 4,... chiều thì tương tự (chỉ cần tổng quát hóa lên).

III.1 Khai báo

III.1.1. Khai báo mảng 2 chiều tường minh

Cú pháp:

<Kiểu> <Tên mảng><[Số phần tử chiều 1]><[Số phần tử chiều 2]>

Ví dụ: Người ta cần lưu trữ thông tin của một ma trận gồm các số thực. Lúc này ta có thể khai báo một mảng 2 chiều như sau:

```
float m[8][9]; /* Khai báo mảng 2 chiều có 8*9 phần tử là số thực*/
```

Trong trường hợp này, ta đã khai báo cho một ma trận có tối đa là 8 dòng, mỗi dòng có tối đa là 9 cột. Hình ảnh của ma trận này được cho trong hình 2:

Dòng\Cột	0	1	2	3	4	5	6	7	8
0	m[0][0]	m[0][1]	m[0][2]	m[0][3]	m[0][4]	m[0][5]	m[0][6]	m[0][7]	m[0][8]
1	m[1][0]	m[1][1]	m[1][2]	m[1][3]	m[1][4]	m[1][5]	m[1][6]	m[1][7]	m[1][8]
2	m[2][0]	m[2][1]	m[2][2]	m[2][3]	m[2][4]	m[2][5]	m[2][6]	m[2][7]	m[2][8]
3	m[3][0]	m[3][1]	m[3][2]	m[3][3]	m[3][4]	m[3][5]	m[3][6]	m[3][7]	m[3][8]
4	m[4][0]	m[4][1]	m[4][2]	m[4][3]	m[4][4]	m[4][5]	m[4][6]	m[4][7]	m[4][8]
5	m[5][0]	m[5][1]	m[5][2]	m[5][3]	m[5][4]	m[5][5]	m[5][6]	m[5][7]	m[5][8]
6	m[6][0]	m[6][1]	m[6][2]	m[6][3]	m[6][4]	m[6][5]	m[6][6]	m[6][7]	m[6][8]
7	m[7][0]	m[7][1]	m[7][2]	m[7][3]	m[7][4]	m[7][5]	m[7][6]	m[7][7]	m[7][8]

Hình 2: Ma trận được mô tả là 1 mảng 2 chiều

III.1.2. Khai báo mảng 2 chiều không tường minh

Để khai báo mảng 2 chiều không tường minh, ta vẫn phải chỉ ra số phần tử của chiều thứ hai (chiều cuối cùng).

Cú pháp: <Kiểu> <Tên mảng> <[]><[Số phần tử chiều 2]>

Cách khai báo này cũng được áp dụng trong trường hợp vừa khai báo, vừa gán trị hay đặt mảng 2 chiều là tham số hình thức của hàm.

III.2 Truy xuất từng phần tử của mảng 2 chiều

Ta có thể truy xuất một phần tử của mảng hai chiều bằng cách viết ra **tên mảng** theo sau là hai chỉ số đặt trong hai cặp dấu ngoặc vuông. Chẳng hạn ta viết m[2][3].

Với cách truy xuất theo cách này, **Tên mảng**[**Chỉ số 1**][**Chỉ số 2**] có thể coi là 1 biến có kiểu được chỉ ra trong khai báo biến mảng.

Ví dụ 1: Viết chương trình cho phép nhập 2 ma trận a, b có m dòng n cột, thực hiện phép toán cộng hai ma trận a,b và in ma trận kết quả lên màn hình.

Trong ví dụ này, ta sẽ sử dụng hàm để làm ngắn gọn hơn chương trình của ta. Ta sẽ viết các hàm: nhập 1 ma trận từ bàn phím, hiển thị ma trận lên màn hình, cộng 2 ma trận.

```
#include<conio.h>
#include<stdio.h>
void Nhap(int a[][10],int M,int N)
{
    int i,j;
    for(i=0;i<M;i++)
```

```

        for(j=0; j<N; j++){
            printf("Phan tu o dong %d cot %d: ",i,j);
            scanf("%d",&a[i][j]);
        }
    }

void InMaTran(int a[][10], int M, int N)
{
    int i,j;
    for(i=0;i<M;i++){
        for(j=0; j< N; j++)
            printf("%d ",a[i][j]);
        printf("\n");
    }
}

/* Cong 2 ma tran A & B ket qua la ma tran C*/
void CongMaTran(int a[][10],int b[][10],int M,int N,int c[][10]){
    int i,j;
    for(i=0;i<M;i++)
        for(j=0; j<N; j++)
            c[i][j]=a[i][j]+b[i][j];
}

int main()
{
    int a[10][10], b[10][10], M, N;
    int c[10][10];/* Ma tran tong*/
    printf("So dong M= "); scanf("%d",&M);
    printf("So cot M= "); scanf("%d",&N);
    printf("Nhap ma tran A\n");
    Nhap(a,M,N);
    printf("Nhap ma tran B\n");
    Nhap(b,M,N);
    printf("Ma tran A: \n");
    InMaTran(a,M,N);
    printf("Ma tran B: \n");
    InMaTran(b,M,N);

    CongMaTran(a,b,M,N,c);
    printf("Ma tran tong C:\n");
    InMaTran(c,M,N);
    getch();
    return 0;
}

```

Ví dụ 2: Nhập vào một ma trận 2 chiều gồm các số thực, in ra tổng của các phần tử trên đường chéo chính của ma trận này.

Ta nhận thấy rằng giả sử ma trận a có M dòng, N cột thì các phần tử của đường chéo chính là các phần tử có dạng: $a[i][i]$ với $i \in [0 \dots \min(M,N)-1]$.

```

#include<conio.h>
#include<stdio.h>

int main()
{
    float a[10][10], T=0;
    int M, N, i, j, Min;
    clrscr();

```

```
printf("Ma tran co bao nhieu dong? ");scanf("%d",&M);
printf("Ma tran co bao nhieu cot? ");scanf("%d",&N);

for(i=0;i<M;i++)
    for(j=0; j<N; j++)
    {
        printf("Phan tu o dong %d cot %d: ",i,j);
        scanf("%f",&a[i][j]);
    }

printf("Ma tran vua nhap: \n");
for(i=0;i<M;i++)
{
    for(j=0; j< N; j++)
        printf("%.2f    ",a[i][j]);
    printf("\n");
}

Min=(M>N) ? N: M; /* Tìm giá trị nhỏ nhất của M & N*/
for(i=0;i<Min;i++)
    T=T+a[i][i];

printf("Tong cac phan tu o duong cheo chinh la: %f",T);
getch();
return 0;
}
```

IV. BÀI TẬP

IV.1 Mục đích yêu cầu

Làm quen với kiểu dữ liệu có cấu trúc trong C, kiểu mảng. Thực hiện các bài tập trong phần nội dung bằng cách kết hợp kiểu dữ liệu mảng, các kiểu dữ liệu đã học và các phần đã học trong các bài tập trước.

IV.2 Nội dung

- Viết chương trình nhập vào một dãy n số thực $a[0], a[1], \dots, a[n-1]$, sắp xếp dãy số theo thứ tự từ lớn đến nhỏ. In dãy số sau khi sắp xếp.
- Viết chương trình sắp xếp một mảng theo thứ tự tăng dần sau khi đã loại bỏ các phần tử trùng nhau.
- Viết chương trình nhập vào một mảng, hãy xuất ra màn hình:
 - Phần tử lớn nhất của mảng.
 - Phần tử nhỏ nhất của mảng.
 - Tính tổng của các phần tử trong mảng .
- Viết chương trình nhập vào một dãy các số theo thứ tự tăng, nếu nhập sai quy cách thì yêu cầu nhập lại. In dãy số sau khi đã nhập xong. Nhập thêm một số mới và chèn số đó vào dãy đã có sao cho dãy vẫn đảm bảo thứ tự tăng. In lại dãy số để kiểm tra.
- Viết chương trình nhập vào một ma trận (mảng hai chiều) các số nguyên, gồm m hàng, n cột. In ma trận đó lên màn hình. Nhập một số nguyên khác vào và xét xem có phần tử nào của ma trận trùng với số này không? Ở vị trí nào? Có bao nhiêu phần tử?

6. Viết chương trình để chuyển đổi vị trí từ dòng thành cột của một ma trận (ma trận chuyển vị) vuông 4 hàng 4 cột. Sau đó viết cho ma trận tổng quát cấp $m \times n$.

Ví dụ:

1	2	3	4		1	2	9	1		
2	5	5	8				2	5	4	5
9	4	2	0				3	5	2	8
1	5	8	6				4	8	0	6

7. Viết chương trình nhập vào một mảng số tự nhiên. Hãy xuất ra màn hình:

- Dòng 1 : gồm các số lẻ, tổng cộng có bao nhiêu số lẻ.
- Dòng 2 : gồm các số chẵn, tổng cộng có bao nhiêu số chẵn.
- Dòng 3 : gồm các số nguyên tố.
- Dòng 4 : gồm các số không phải là số nguyên tố.

8. Viết chương trình tính tổng bình phương của các số âm trong một mảng các số nguyên.

9. Viết chương trình thực hiện việc đảo một mảng một chiều.

Ví dụ : 1 2 3 4 5 7 9 10 đảo thành 10 9 7 5 4 3 2 1 .

10. Viết chương trình nhập vào hai ma trận A và B có cấp m, n. In hai ma trận lên màn hình. Tổng hai ma trận A và B là ma trận C được tính bởi công thức:

$$c_{ij} = a_{ij} + b_{ij} \quad (i=0,1,2,\dots,m-1; j=0,1,2,\dots,n-1)$$

Tính ma trận tổng C và in kết quả lên màn hình.

11. Viết chương trình nhập vào hai ma trận A có cấp m, k và B có cấp k, n. In hai ma trận lên màn hình. Tích hai ma trận A và B là ma trận C được tính bởi công thức:

$$c_{ij} = a_{i1} * b_{1j} + a_{i2} * b_{2j} + a_{i3} * b_{3j} + \dots + a_{ik} * b_{kj} \quad (i=0,1,2,\dots,m-1; j=0,1,2,\dots,n-1)$$

Tính ma trận tích C và in kết quả lên màn hình.

12. Xét ma trận A vuông cấp n, các phần tử $a[i, i]$ ($i = 1 \dots n$) được gọi là đường chéo chính của ma trận vuông A. Ma trận vuông A được gọi là ma trận tam giác nếu tất cả các phần tử dưới đường chéo chính đều bằng 0. Định thức của ma trận tam giác bằng tích các phần tử trên đường chéo chính.

Ta có thể chuyển một ma trận vuông bất kỳ về ma trận tam giác bằng thuật toán:

- Xét cột i ($i = 0, 1 \dots n-2$)
- Trong cột i xét các phần tử $a[k, i]$ ($k = i+1 \dots n-1$)
 - + Nếu $a[k, i] = 0$ thì tăng k lên xét phần tử khác
 - + Nếu $a[k, i] \neq 0$ thì làm như sau:
 - Nhân toàn bộ hàng k với $-a[i, i]/a[k, i]$
 - Lấy hàng i cộng vào hàng k sau khi thực hiện phép nhân trên.
 - Đổi chỗ hai hàng i và k cho nhau
 - Nhân toàn bộ hàng k với -1 sau khi đã đổi chỗ với hàng i
 - Tăng k lên xét phần tử khác.

Viết chương trình tính định thức cấp n thông qua các bước nhập ma trận, in ma trận, đưa ma trận về dạng tam giác, in ma trận tam giác, in kết quả tính định thức.

13. Viết chương trình thực hiện việc trộn hai dãy có thứ tự thành một dãy có thứ tự. Yêu cầu không được trộn chung rồi mới sắp thứ tự. Khi trộn phải tận dụng được tính chất đã sắp của hai dãy con.

Chương VII

KIỂU CON TRỎ

Học xong chương này, sinh viên sẽ nắm được các vấn đề sau:

- Khái niệm về kiểu dữ liệu “con trỏ”.
- Cách khai báo và cách sử dụng biến kiểu con trỏ.
- Mối quan hệ giữa mảng và con trỏ.

I. GIỚI THIỆU KIỂU DỮ LIỆU CON TRỎ

Các biến chúng ta đã biết và sử dụng trước đây đều là biến có kích thước và kiểu dữ liệu xác định. Người ta gọi các biến kiểu này là biến tĩnh. Khi khai báo biến tĩnh, một lượng ô nhớ cho các biến này sẽ được cấp phát mà không cần biết trong quá trình thực thi chương trình có sử dụng hết lượng ô nhớ này hay không. Mặt khác, các biến tĩnh dạng này sẽ tồn tại trong suốt thời gian thực thi chương trình dù có những biến mà chương trình chỉ sử dụng 1 lần rồi bỏ.

Một số hạn chế có thể gặp phải khi sử dụng các biến tĩnh:

- Cấp phát ô nhớ dư, gây ra lãng phí ô nhớ.
- Cấp phát ô nhớ thiếu, chương trình thực thi bị lỗi.

Để tránh những hạn chế trên, ngôn ngữ C cung cấp cho ta một loại biến đặc biệt gọi là biến động với các đặc điểm sau:

- Chỉ phát sinh trong quá trình thực hiện chương trình chứ không phát sinh lúc bắt đầu chương trình.
- Khi chạy chương trình, kích thước của biến, vùng nhớ và địa chỉ vùng nhớ được cấp phát cho biến có thể thay đổi.
- Sau khi sử dụng xong có thể giải phóng để tiết kiệm chỗ trong bộ nhớ.

Tuy nhiên các biến động không có địa chỉ nhất định nên ta không thể truy cập đến chúng được. Vì thế, ngôn ngữ C lại cung cấp cho ta một loại biến đặc biệt nữa để khắc phục tình trạng này, đó là biến con trỏ (pointer) với các đặc điểm:

- Biến con trỏ không chứa dữ liệu mà chỉ chứa địa chỉ của dữ liệu hay chứa địa chỉ của ô nhớ chứa dữ liệu.
- Kích thước của biến con trỏ không phụ thuộc vào kiểu dữ liệu, luôn có kích thước cố định là 2 byte.

II. KHAI BÁO VÀ SỬ DỤNG BIẾN CON TRỎ

II.1. Khai báo biến con trỏ

Cú pháp: <Kiểu> * <Tên con trỏ>

Ý nghĩa: Khai báo một biến có tên là *Tên con trỏ* dùng để chứa địa chỉ của các biến có kiểu *Kiểu*.

Ví dụ 1: Khai báo 2 biến a,b có kiểu int và 2 biến pa, pb là 2 biến con trỏ kiểu int.
int a, b, *pa, *pb;

Ví dụ 2: Khai báo biến f kiểu float và biến pf là con trỏ float
float f, *pf;

Ghi chú: Nếu chưa muốn khai báo kiểu dữ liệu mà con trỏ ptr đang chỉ đến, ta sử dụng:

```
void *ptr;
```

Sau đó, nếu ta muốn con trỏ ptr chỉ đến kiểu dữ liệu gì cũng được. Tác dụng của khai báo này là chỉ dành ra 2 bytes trong bộ nhớ để cấp phát cho biến con trỏ ptr.

II.2. Các thao tác trên con trỏ

II.2.1 Gán địa chỉ của biến cho biến con trỏ

Toán tử & dùng để định vị con trỏ đến địa chỉ của một biến đang làm việc.

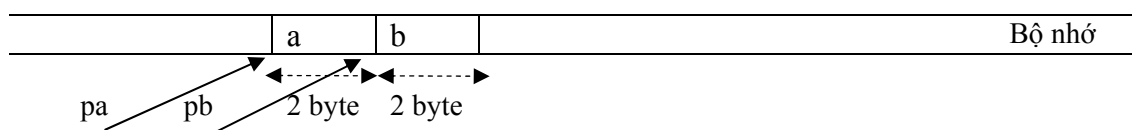
Cú pháp: <Tên biến con trỏ>=&<Tên biến>

Giải thích: Ta gán địa chỉ của biến *Tên biến* cho con trỏ *Tên biến con trỏ*.

Ví dụ: Gán địa chỉ của biến a cho con trỏ pa, gán địa chỉ của biến b cho con trỏ pb.

```
pa=&a; pb=&b;
```

Lúc này, hình ảnh của các biến trong bộ nhớ được mô tả:



Lưu ý:

Khi gán địa chỉ của biến tĩnh cho con trỏ cần phải lưu ý kiểu dữ liệu của chúng. Ví dụ sau đây không đúng do không tương thích kiểu:

```
int Bien_Nguyen;
float *Con_Tro_Thuc;
...
Con_Tro_Thuc=&Bien_Nguyen;
```

Phép gán ở đây là sai vì Con_Tro_Thuc là một con trỏ kiểu float (nó chỉ có thể chứa được địa chỉ của biến kiểu float); trong khi đó, Bien_Nguyen có kiểu int.

II.2.2 Nội dung của ô nhớ con trỏ chỉ tới

Để truy cập đến nội dung của ô nhớ mà con trỏ chỉ tới, ta sử dụng cú pháp:

***<Tên biến con trỏ>**

Với cách truy cập này thì ***<Tên biến con trỏ>** có thể coi là một biến có kiểu được mô tả trong phần khai báo biến con trỏ.

Ví dụ: Ví dụ sau đây cho phép khai báo, gán địa chỉ cũng như lấy nội dung vùng nhớ của biến con trỏ:

```
int x=100;
int *ptr;
ptr=&x;
int y= *ptr;
```

Lưu ý: Khi gán địa chỉ của một biến cho một biến con trỏ, mọi sự thay đổi trên nội dung ô nhớ con trỏ chỉ tới sẽ làm giá trị của biến thay đổi theo (thực chất nội dung ô nhớ và biến chỉ là một).

Ví dụ: Đoạn chương trình sau thấy rõ sự thay đổi này :

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int a,b,*pa,*pb;
    a=2;
    b=3;
```

```

clrscr();
printf("\nGia tri cua bien a=%d \nGia tri cua bien b=%d ",a,b);
pa=&a;
pb=&b;
printf("\nNoi dung cua o nho con tro pa tro toi=%d",*pa);
printf("\nNoi dung cua o nho con tro pb tro toi=%d ",*pb);
*pa=20; /* Thay doi gia tri cua *pa*/
*pb=20; /* Thay doi gia tri cua *pb*/
printf("\nGia tri moi cua bien a=%d \n
      Gia tri moi cua bien b=%d ",a,b); /* a, b thay doi theo*/
getch();
return 0;
}

```

Kết quả thực hiện chương trình:

```

c:\ TC.EXE
Gia tri cua bien a=2
Gia tri cua bien b=3
Noi dung cua o nho con tro pa tro toi=2
Noi dung cua o nho con tro pb tro toi=3
Gia tri moi cua bien a=20
Gia tri moi cua bien b=20

```

II.2.3 Cấp phát vùng nhớ cho biến con trỏ

Trước khi sử dụng biến con trỏ, ta nên cấp phát vùng nhớ cho biến con trỏ này quản lý địa chỉ. Việc cấp phát được thực hiện nhờ các hàm malloc(), calloc() trong thư viện alloc.h.

Cú pháp các hàm:

void *malloc(size_t size): Cấp phát vùng nhớ có kích thước là size.

void *calloc(size_t nitems, size_t size): Cấp phát vùng nhớ có kích thước là nitems*size.

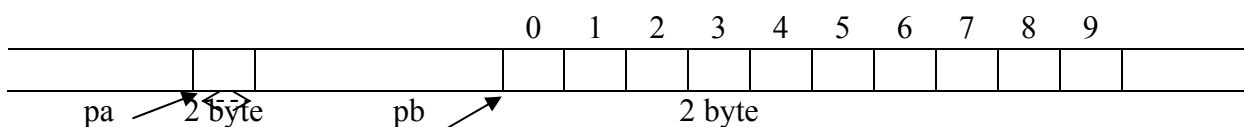
Ví dụ: Giả sử ta có khai báo:

```

int a, *pa, *pb;
pa = (int*)malloc(sizeof(int)); /* Cấp phát vùng nhớ có kích thước bằng
với kích thước của một số nguyên */
pb= (int*)calloc(10, sizeof(int)); /* Cấp phát vùng nhớ có thể chứa được
10 số nguyên*/

```

Lúc này hình ảnh trong bộ nhớ như sau:



Lưu ý: Khi sử dụng hàm malloc() hay calloc(), ta phải ép kiểu vì nguyên mẫu các hàm này trả về con trỏ kiểu void.

II.2.4 Cấp phát lại vùng nhớ cho biến con trỏ

Trong quá trình thao tác trên biến con trỏ, nếu ta cần cấp phát thêm vùng nhớ có kích thước lớn hơn vùng nhớ đã cấp phát, ta sử dụng hàm realloc().

Cú pháp: void *realloc(void *block, size_t size)

Ý nghĩa:

- Cấp phát lại 1 vùng nhớ cho con trỏ *block* quản lý, vùng nhớ này có kích thước mới là *size*; khi cấp phát lại thì nội dung vùng nhớ trước đó vẫn tồn tại.

- Kết quả trả về của hàm là địa chỉ đầu tiên của vùng nhớ mới. Địa chỉ này có thể khác với địa chỉ được chỉ ra khi cấp phát ban đầu.

Ví dụ: Trong ví dụ trên ta có thể cấp phát lại vùng nhớ do con trỏ pa quản lý như sau:

```
int a, *pa;
pa=(int*)malloc(sizeof(int)); /*Cấp phát vùng nhớ có kích thước 2 byte*/
pa = realloc(pa, 6); /* Cấp phát lại vùng nhớ có kích thước 6 byte*/
```

II.2.5 Giải phóng vùng nhớ cho biến con trỏ

Một vùng nhớ đã cấp phát cho biến con trỏ, khi không còn sử dụng nữa, ta sẽ thu hồi lại vùng nhớ này nhờ hàm free().

Cú pháp: void free(void *block)

Ý nghĩa: Giải phóng vùng nhớ được quản lý bởi con trỏ block.

Ví dụ: Ở ví dụ trên, sau khi thực hiện xong, ta giải phóng vùng nhớ cho 2 biến con trỏ pa & pb:

```
free(pa);
free(pb);
```

II.2.6 Một số phép toán trên con trỏ

a. *Phép gán con trỏ:* Hai con trỏ cùng kiểu có thể gán cho nhau.

Ví dụ:

```
int a, *p, *a ; float *f;
a = 5 ; p = &a ; q = p ; /* đúng */
f = p ; /* sai do khác kiểu */
```

Ta cũng có thể ép kiểu con trỏ theo cú pháp:

(<Kiểu kết quả>*)<Tên con trỏ>

Chẳng hạn, ví dụ trên được viết lại:

```
int a, *p, *a ; float *f;
a = 5 ; p = &a ; q = p ; /* đúng */
f = (float*)p; /* Đúng nhờ ép kiểu*/
```

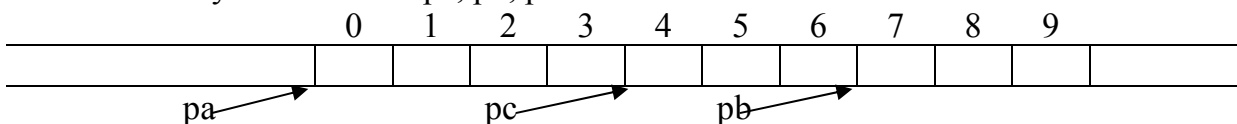
b. *Cộng, trừ con trỏ với một số nguyên*

Ta có thể cộng (+), trừ (-) 1 con trỏ với 1 số nguyên N nào đó; kết quả trả về là 1 con trỏ. Con trỏ này chỉ đến vùng nhớ cách vùng nhớ của con trỏ hiện tại N phần tử.

Ví dụ: Cho đoạn chương trình sau:

```
int *pa;
pa = (int*) malloc(20); /* Cấp phát vùng nhớ 20 byte=10 số nguyên*/
int *pb, *pc;
pb = pa + 7;
pc = pb - 3;
```

Lúc này hình ảnh của pa, pb, pc như sau:



c. *Con trỏ NULL:* là con trỏ không chứa địa chỉ nào cả. Ta có thể gán giá trị NULL cho 1 con trỏ có kiểu bất kỳ.

d. *Lưu ý:*

- Ta không thể cộng 2 con trỏ với nhau.

- Phép trừ 2 con trỏ cùng kiểu sẽ trả về 1 giá trị nguyên (int). Đây chính là khoảng cách (số phần tử) giữa 2 con trỏ đó. *Chẳng hạn*, trong ví dụ trên $pc-pa=4$.

III. CON TRỎ VÀ MẢNG

III.1 Con trỏ và mảng 1 chiều

Giữa mảng và con trỏ có một sự liên hệ rất chặt chẽ. Những phần tử của mảng có thể được xác định bằng chỉ số trong mảng, bên cạnh đó chúng cũng có thể được xác lập qua biến con trỏ.

III.1.1 Truy cập các phần tử mảng theo dạng con trỏ

Ta có các quy tắc sau:

&<Tên mảng>[0] tương đương với **<Tên mảng>**

&<Tên mảng> [<Vị trí>] tương đương với **<Tên mảng> + <Vị trí>**

<Tên mảng>[<Vị trí>] tương đương với ***(<Tên mảng> + <Vị trí>)**

Ví dụ: Cho 1 mảng 1 chiều các số nguyên a có 5 phần tử, truy cập các phần tử theo kiểu mảng và theo kiểu con trỏ.

```
#include <stdio.h>
#include <conio.h>
/* Nhập mảng bình thường*/
void NhapMang(int a[], int N){
    int i;
    for(i=0;i<N;i++){
        {
            printf("Phan tu thu %d: ",i);scanf("%d",&a[i]);
        }
    }
    /* Nhập mảng theo dạng con trỏ*/
void NhapContro(int a[], int N)
{
    int i;
    for(i=0;i<N;i++){
        printf("Phan tu thu %d: ",i);scanf("%d",a+i);
    }
}

int main()
{
    int a[20],N,i;
    clrscr();
    printf("So phan tu N= ");scanf("%d",&N);
    NhapMang(a,N); /* NhapContro(a,N)*/
    printf("Truy cap theo kieu mang: ");
    for(i=0;i<N;i++)
        printf("%d ",a[i]);
    printf("\nTruy cap theo kieu con tro: ");
    for(i=0;i<N;i++)
        printf("%d ",*(a+i));
    getch();
    return 0;
}
```

Kết quả thực thi của chương trình:

```

c:\ TC.EXE
So phan tu N= 4
Phan tu thu 0: 2
Phan tu thu 1: 3
Phan tu thu 2: 4
Phan tu thu 3: 5
Truy cap theo kieu mang: 2 3 4 5
Truy cap theo kieu con tro: 2 3 4 5 _
    
```

III.1.2 Truy xuất từng phần tử đang được quản lý bởi con trỏ theo dạng mảng

<Tên biến>[<Vị trí>] tương đương với ***(<Tên biến> + <Vị trí>)**

&<Tên biến>[<Vị trí>] tương đương với **(<Tên biến> + <Vị trí>)**

Trong đó <Tên biến> là biến con trỏ, <Vị trí> là 1 biểu thức số nguyên.

Ví dụ: Giả sử có khai báo:

```

#include <stdio.h>
#include <alloc.h>
#include <conio.h>

int main() {
    int *a;
    int i;
    clrscr();
    a=(int*)malloc(sizeof(int)*10);
    for(i=0;i<10;i++)
        a[i] = 2*i;
    printf("Truy cap theo kieu mang: ");
    for(i=0;i<10;i++)
        printf("%d ",a[i]);
    printf("\nTruy cap theo kieu con tro: ");
    for(i=0;i<10;i++)
        printf("%d ",*(a+i));
    getch();
    return 0;
}
    
```

Kết quả chương trình:

```

c:\ TC.EXE
Truy cap theo kieu mang: 0 2 4 6 8 10 12 14 16 18
Truy cap theo kieu con tro: 0 2 4 6 8 10 12 14 16 18
    
```

Với khai báo ở trên, hình ảnh của con trỏ a trong bộ nhớ:

	0	1	2	3	4	5	6	7	8	9
a	0	2	4	6	8	10	12	14	16	18

2 byte

III.1.3 Con trỏ chỉ đến phần tử mảng

Giả sử con trỏ ptr chỉ đến phần tử a[i] nào đó của mảng a thì:

ptr + j chỉ đến phần tử thứ j sau a[i], tức a[i+j]

ptr - j chỉ đến phần tử đứng trước a[i], tức a[i-j]

Ví dụ: Giả sử có 1 mảng mang_int, cho con trỏ contro_int chỉ đến phần tử thứ 5 trong mảng. In ra các phần tử của contro_int & mang_int.

```

#include <stdio.h>
    
```

```

#include <conio.h>
#include <alloc.h>
int main()
{
    int i,mang_int[10];
    int *contro_int;
    clrscr();
    for(i=0;i<=9;i++)
        mang_int[i]=i*2;
    contro_int=&mang_int[5];
    printf("\nNoi dung cua mang_int ban dau=");
    for (i=0;i<=9;i++)
        printf("%d ",mang_int[i]);
    printf("\nNoi dung cua contro_int ban dau =");
    for (i=0;i<5;i++)
        printf("%d ",contro_int[i]);

    for(i=0;i<5;i++)
        contro_int[i]++;
    printf("\n-----");
    printf("\nNoi dung cua mang_int sau khi tang 1=");
    for (i=0;i<=9;i++)
        printf("%d ",mang_int[i]);
    printf("\nNoi dung cua contro_int sau khi tang 1=");
    for (i=0;i<5;i++)
        printf("%d ",contro_int[i]);
    if (contro_int!=NULL)
        free(contro_int);
    getch();
    return 0;
}

```

Kết quả chương trình

```

C:\ TC.EXE
Noi dung cua mang_int ban dau=0 2 4 6 8 10 12 14 16 18
Noi dung cua contro_int ban dau =10 12 14 16 18
-----
Noi dung cua mang_int sau khi tang 1=0 2 4 6 8 11 13 15 17 19
Noi dung cua contro_int sau khi tang 1=11 13 15 17 19

```

III.2 Con trỏ và mảng nhiều chiều

Ta có thể sử dụng con trỏ thay cho mảng nhiều chiều như sau:

Giả sử ta có mảng 2 chiều và biến con trỏ như sau:

```
int a[n][m];
```

```
int *contro_int;
```

Thực hiện phép gán `contro_int=a;`

Khi đó phần tử `a[0][0]` được quản lý bởi `contro_int;`

`a[0][1]` được quản lý bởi `contro_int+1;`

`a[0][2]` được quản lý bởi `contro_int+2;`

...

`a[1][0]` được quản lý bởi `contro_int+m;`

`a[1][1]` được quản lý bởi `contro_int+m+1;`

...

`a[n][m]` được quản lý bởi `contro_int+n*m;`

Tương tự như thế đối với mảng nhiều hơn 2 chiều.

Ví dụ: Sự tương đương giữa mảng 2 chiều và con trỏ.

```
#include <stdio.h>
#include <conio.h>
#include <alloc.h>
int main()
{
    int i,j;
    int mang_int[4][5]={1,2,3,4,5,6,7,8,9,10,11,12,13,14,
                        15,16,17,18,19,20};

    int *contro_int;
    clrscr();
    contro_int=(int*)mang_int;
    printf("\nNoi dung cua mang_int ban dau=");
    for (i=0;i<4;i++)
    {
        printf("\n");
        for (j=0;j<5;j++)
            printf("%d\t",mang_int[i][j]);
    }
    printf("\n-----");
    printf("\nNoi dung cua contro_int ban dau \n");
    for (i=0;i<20;i++)
        printf("%d ",contro_int[i]);

    for(i=0;i<20;i++)
        contro_int[i]++;
    printf("\n-----");
    printf("\nNoi dung cua mang_int sau khi tang 1=");
    for (i=0;i<4;i++)
    {
        printf("\n");
        for (j=0;j<5;j++)
            printf("%d\t",mang_int[i][j]);
    }

    printf("\nNoi dung cua contro_int sau khi tang 1=\n");
    for (i=0;i<20;i++)
        printf("%d ",contro_int[i]);
    if (contro_int!=NULL)
        free(contro_int);
    getch();
    return 0;
}
```

Kết quả thực hiện chương trình như sau:

```
Noi dung cua mang_int ban dau=
1  2  3  4  5
6  7  8  9  10
11 12 13 14 15
16 17 18 19 20
-----
Noi dung cua contro_int ban dau
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
-----
Noi dung cua mang_int sau khi tang 1=
2  3  4  5  6
7  8  9  10 11
12 13 14 15 16
17 18 19 20 21
Noi dung cua contro_int sau khi tang 1=
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
```

IV. CON TRỎ VÀ THAM SỐ HÌNH THỨC CỦA HÀM

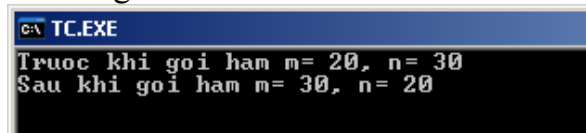
Khi tham số hình thức của hàm là một con trỏ thì theo nguyên tắc gọi hàm ta dùng tham số thực tế là 1 con trỏ có kiểu giống với kiểu của tham số hình thức. Nếu lúc thực thi hàm ta có sự thay đổi trên nội dung vùng nhớ được chỉ bởi con trỏ tham số hình thức thì lúc đó nội dung vùng nhớ được chỉ bởi tham số thực tế cũng sẽ bị thay đổi theo.

Ví dụ : Xét hàm hoán vị được viết như sau :

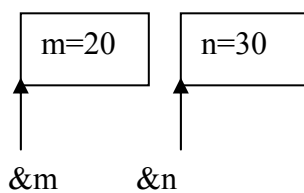
```
#include<stdio.h>
#include<conio.h>
void HoanVi(int *a, int *b)
{
    int c=*a;
    *a=*b;
    *b=c;
}

int main()
{
    int m=20,n=30;
    clrscr();
    printf("Truoc khi goi ham m= %d, n= %d\n",m,n);
    HoanVi (&m,&n);
    printf("Sau khi goi ham m= %d, n= %d",m,n);
    getch();
    return 0;
}
```

Kết quả thực thi chương trình:

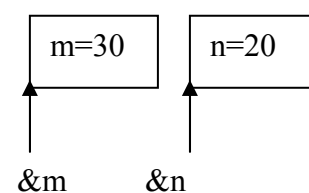
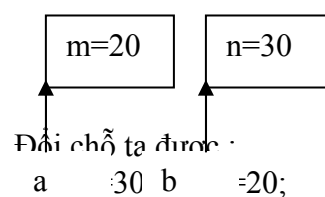


Trước khi gọi hàm



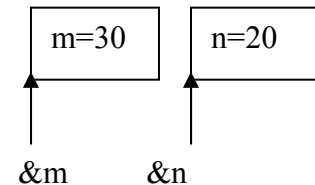
Khi gọi hàm

a=&m; b= &n;
Lúc này : *a=m; *b=n;



Sau khi gọi hàm:

Con trỏ a, b bị giải phóng
m, n đã thay đổi:



V. BÀI TẬP

V.1 Mục tiêu

Tiếp cận với một kiểu dữ liệu rất mạnh trong C là kiểu con trỏ. Từ đó, sinh viên có thể xây dựng các ứng dụng bằng cách sử dụng cấp phát động thông qua biến con trỏ.

V.2 Nội dung

Thực hiện các bài tập ở chương trước (chương VI : Kiểu mảng) bằng cách sử dụng con trỏ.

Chương VIII: CHUỖI KÝ TỰ

Học xong chương này, sinh viên sẽ nắm được các vấn đề sau:

- *Khái niệm về chuỗi ký tự.*
- *Một số hàm xử lý chuỗi và ứng dụng của chúng trong thực tế.*

I. KHÁI NIỆM

Chuỗi ký tự là một dãy gồm các ký tự hoặc một mảng các ký tự được kết thúc bằng ký tự ‘\0’ (còn được gọi là ký tự NULL trong bảng mã Ascii).

Các hằng chuỗi ký tự được đặt trong cặp dấu nháy kép “”.

II. KHAI BÁO

II.1 Khai báo theo mảng

Cú pháp: `char <Biến> [Chiều dài tối đa]`

Ví dụ: Trong chương trình, ta có khai báo:

```
char Ten[12];
```

Trong khai báo này, bộ nhớ sẽ cung cấp 12+1 bytes để lưu trữ nội dung của chuỗi ký tự Ten; byte cuối cùng lưu trữ ký tự ‘\0’ để chấm dứt chuỗi.

Ghi chú:

- Chiều dài tối đa của biến chuỗi là một hằng nguyên nằm trong khoảng từ 1 đến 255 bytes.

- Chiều dài tối đa không nên khai báo thừa để tránh lãng phí bộ nhớ, nhưng cũng không nên khai báo thiếu.

II.2 Khai báo theo con trỏ

Cú pháp: `char *<Biến>`

Ví dụ: Trong chương trình, ta có khai báo:

```
char *Ten;
```

Trong khai báo này, bộ nhớ sẽ dành 2 byte để lưu trữ địa chỉ của biến con trỏ Ten đang chỉ đến, chưa cung cấp nơi để lưu trữ dữ liệu. Muốn có chỗ để lưu trữ dữ liệu, ta phải gọi đến hàm *malloc()* hoặc *calloc()* có trong “alloc.h”, sau đó mới gán dữ liệu cho biến.

II.3 Vừa khai báo vừa gán giá trị

Cú pháp: `char <Biến>[]=<”Hằng chuỗi”>`

Ví dụ:

```
#include<stdio.h>
```

```
#include<conio.h>
```

```
int main()
```

```
{
```

```
    char Chuoi[]="Mau nang hay la mau mat em" ;
```

```

printf("Vua khai bao vua gan tri : %s", Chuoi) ;
getch();
return 0;
}

```

* **Ghi chú:** Chuỗi được khai báo là một mảng các ký tự nên các thao tác trên mảng có thể áp dụng đối với chuỗi ký tự.

III. CÁC THAO TÁC TRÊN CHUỖI KÝ TỰ

III.1. Nhập xuất chuỗi

III.1.1 Nhập chuỗi từ bàn phím

Để nhập một chuỗi ký tự từ bàn phím, ta sử dụng hàm gets()

Cú pháp: gets(<Biến chuỗi>)

Ví dụ: char Ten[20];
gets(Ten);

Ta cũng có thể sử dụng hàm scanf() để nhập dữ liệu cho biến chuỗi, tuy nhiên lúc này ta chỉ có thể nhập được một chuỗi không có dấu khoảng trắng.

Ngoài ra, hàm cgets() (trong conio.h) cũng được sử dụng để nhập chuỗi.

III.1.2 Xuất chuỗi lên màn hình

Để xuất một chuỗi (biểu thức chuỗi) lên màn hình, ta sử dụng hàm puts().

Cú pháp: puts(<Biểu thức chuỗi>)

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("Nhap chuoi: "); gets(Ten);
printf("Chuoi vua nhap: "); puts(Ten);
getch();
return 0;
}

```

Ngoài ra, ta có thể sử dụng hàm printf(), cputs() (trong conio.h) để hiển thị chuỗi lên màn hình.

III.2 Một số hàm xử lý chuỗi (trong string.h)

III.2.1 Cộng chuỗi - Hàm strcat()

Cú pháp: char *strcat(char *des, const char *source)

Hàm này có tác dụng ghép chuỗi nguồn vào chuỗi đích.

Ví dụ: Nhập vào họ lót và tên của một người, sau đó in cả họ và tên của họ lên màn hình.

```

#include<conio.h>
#include<stdio.h>
#include<string.h>

```

```
int main()
{
    char HoLot[30], Ten[12];
    printf("Nhap Ho Lot: ");gets(HoLot);
    printf("Nhap Ten: ");gets(Ten);
    strcat(HoLot,Ten); /* Ghep Ten vao HoLot*/
    printf("Ho ten la: ");puts(HoLot);
    getch();
    return 0;
}
```

III.2.2 Xác định độ dài chuỗi - Hàm strlen()

Cú pháp: `int strlen(const char* s)`

Ví dụ: Sử dụng hàm strlen xác định độ dài một chuỗi nhập từ bàn phím.

```
#include<conio.h>
#include<stdio.h>
#include<string.h>

int main() {
    char Chuoi[255];
    int Dodai;
    printf("Nhap chuoi: ");gets(Chuoi);
    Dodai = strlen(Chuoi)
    printf("Chuoi vua nhap: ");puts(Chuoi);
    printf("Co do dai %d",Dodai);
    getch();
    return 0;
}
```

III.2.3 Đổi một ký tự thường thành ký tự hoa - Hàm toupper()

Hàm toupper() (trong ctype.h) được dùng để chuyển đổi một ký tự thường thành ký tự hoa.

Cú pháp: `char toupper(char c)`

III.2.4 Đổi chuỗi chữ thường thành chuỗi chữ hoa, hàmstrupr()

Hàmstrupr() được dùng để chuyển đổi chuỗi chữ thường thành chuỗi chữ hoa, kết quả trả về của hàm là một con trỏ chỉ đến địa chỉ chuỗi được chuyển đổi.

Cú pháp: `char *strupr(char *s)`

Ví dụ: Viết chương trình nhập vào một chuỗi ký tự từ bàn phím. Sau đó sử dụng hàmstrupr() để chuyển đổi chúng thành chuỗi chữ hoa.

```
#include<conio.h>
#include<stdio.h>
#include<string.h>

int main()
{
    char Chuoi[255],*s;
    printf("Nhap chuoi: ");gets(Chuoi);
    s=strupr(Chuoi) ;
    printf("Chuoi chu hoa: ");puts(s);
    getch();
    return 0;
}
```

III.2.5 Đổi chuỗi chữ hoa thành chuỗi chữ thường, hàm `strlwr()`

Muốn chuyển đổi chuỗi chữ hoa thành chuỗi toàn chữ thường, ta sử dụng hàm `strlwr()`, các tham số của hàm tương tự như hàm `strupr()`

Cú pháp: `char *strlwr(char *s)`

III.2.6 Sao chép chuỗi, hàm `strcpy()`

Hàm này được dùng để sao chép toàn bộ nội dung của chuỗi nguồn vào chuỗi đích.

Cú pháp: `char *strcpy(char *Des, const char *Source)`

Ví dụ: Viết chương trình cho phép chép toàn bộ chuỗi nguồn vào chuỗi đích.

```
#include<conio.h>
#include<stdio.h>
#include<string.h>

int main()
{
    char Chuoi[255],s[255];
    printf("Nhap chuoi: ");gets(Chuoi);
    strcpy(s,Chuoi);
    printf("Chuoi dich: ");puts(s);
    getch();
    return 0;
}
```

III.2.7 Sao chép một phần chuỗi, hàm `strncpy()`

Hàm này cho phép chép n ký tự đầu tiên của chuỗi nguồn sang chuỗi đích.

Cú pháp: `char *strncpy(char *Des, const char *Source, size_t n)`

III.2.8 Trích một phần chuỗi, hàm `strchr()`

Để trích một chuỗi con của một chuỗi ký tự bắt đầu từ một ký tự được chỉ định trong chuỗi cho đến hết chuỗi, ta sử dụng hàm `strchr()`.

Cú pháp : `char *strchr(const char *str, int c)`

Ghi chú:

- Nếu ký tự đã chỉ định không có trong chuỗi, kết quả trả về là NULL.
- Kết quả trả về của hàm là một con trỏ, con trỏ này chỉ đến ký tự c được tìm thấy đầu tiên trong chuỗi str.

III.2.9 Tìm kiếm nội dung chuỗi, hàm `strstr()`

Hàm `strstr()` được sử dụng để tìm kiếm sự xuất hiện đầu tiên của chuỗi s2 trong chuỗi s1.

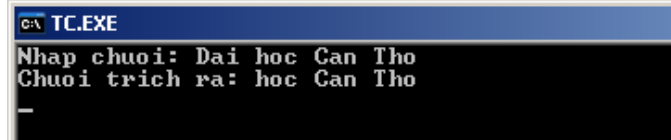
Cú pháp: `char *strstr(const char *s1, const char *s2)`

Kết quả trả về của hàm là một con trỏ chỉ đến phần tử đầu tiên của chuỗi s1 có chứa chuỗi s2 hoặc giá trị NULL nếu chuỗi s2 không có trong chuỗi s1.

Ví dụ: Viết chương trình sử dụng hàm `strstr()` để lấy ra một phần của chuỗi gốc bắt đầu từ chuỗi "hoc".

```
#include<conio.h>
#include<stdio.h>
#include<string.h>
```

```
int main()
{
    char Chuoi[255], *s;
    printf("Nhap chuoi: "); gets(Chuoi);
    s=strstr(Chuoi, "hoc");
    printf("Chuoi trích ra: "); puts(s);
    getch();
    return 0;
}
```



III.2.10 So sánh chuỗi, hàm strcmp()

Để so sánh hai chuỗi theo từng ký tự trong bảng mã Ascii, ta có thể sử dụng hàm strcmp().

Cú pháp: `int strcmp(const char *s1, const char *s2)`

Hai chuỗi s1 và s2 được so sánh với nhau, kết quả trả về là một số nguyên (số này có được bằng cách lấy ký tự của s1 trừ ký tự của s2 tại vị trí đầu tiên xảy ra sự khác nhau).

- Nếu kết quả là số âm, chuỗi s1 nhỏ hơn chuỗi s2.
- Nếu kết quả là 0, hai chuỗi bằng nhau.
- Nếu kết quả là số dương, chuỗi s1 lớn hơn chuỗi s2.

III.2.11 So sánh chuỗi, hàm stricmp()

Hàm này thực hiện việc so sánh trong n ký tự đầu tiên của 2 chuỗi s1 và s2, giữa chữ thường và chữ hoa không phân biệt.

Cú pháp: `int stricmp(const char *s1, const char *s2)`

Kết quả trả về tương tự như kết quả trả về của hàm strcmp()

III.2.12 Khởi tạo chuỗi, hàm memset()

Hàm này được sử dụng để đặt n ký tự đầu tiên của chuỗi là ký tự c.

Cú pháp: `memset(char *Des, int c, size_t n)`

III.2.13 Đổi từ chuỗi ra số, hàm atoi(), atof(), atol() (trong stdlib.h)

Để chuyển đổi chuỗi ra số, ta sử dụng các hàm trên.

Cú pháp : `int atoi(const char *s)` : chuyển chuỗi thành số nguyên

`long atol(const char *s)` : chuyển chuỗi thành số nguyên dài

`float atof(const char *s)` : chuyển chuỗi thành số thực

Nếu chuyển đổi không thành công, kết quả trả về của các hàm là 0.

Ngoài ra, thư viện *string.h* còn hỗ trợ các hàm xử lý chuỗi khác, ta có thể đọc thêm trong phần trợ giúp.

IV. BÀI TẬP

IV.1 Mục đích yêu cầu

Đi sâu vào kiểu dữ liệu chuỗi và các phép toán trên chuỗi.

IV.2 Nội dung

1. Viết chương trình nhập một chuỗi ký tự từ bàn phím, xuất ra màn hình mã Ascii của từng ký tự có trong chuỗi.
2. Viết chương trình nhập một chuỗi ký tự từ bàn phím, xuất ra màn hình chuỗi đảo ngược của chuỗi đó. Ví dụ đảo của “abcd egh” là “hge dcba”.
3. Viết chương trình nhập một chuỗi ký tự và kiểm tra xem chuỗi đó có đối xứng không.

Ví dụ : Chuỗi ABCDEDCBA là chuỗi đối xứng.

4. Nhập vào một chuỗi bất kỳ, hãy đếm số lần xuất hiện của mỗi loại ký tự.
5. Viết chương trình nhập vào một chuỗi.

- In ra màn hình từ bên trái nhất và phần còn lại của chuỗi. Ví dụ: “Nguyễn Văn Minh” in ra thành:

Nguyễn
Văn Minh

- In ra màn hình từ bên phải nhất và phần còn lại của chuỗi. Ví dụ: “Nguyễn Văn Minh” in ra thành:

Minh
Nguyễn Văn

6. Viết chương trình nhập vào một chuỗi rồi xuất chuỗi đó ra màn hình dưới dạng mỗi từ một dòng.

Ví dụ: “Nguyễn Văn Minh”

In ra :

Nguyễn
Văn
Minh

7. Viết chương trình nhập vào một chuỗi, in ra chuỗi đảo ngược của nó theo từng từ.

Ví dụ : chuỗi “Nguyễn Văn Minh” đảo thành “Minh Văn Nguyễn”

8. Viết chương trình đổi số tiền từ số thành chữ.
9. Viết chương trình nhập vào họ và tên của một người, cắt bỏ các khoảng trống không cần thiết (nếu có), tách tên ra khỏi họ và tên, in tên lên màn hình. Chú ý đến trường hợp cả họ và tên chỉ có một từ.
10. Viết chương trình nhập vào họ và tên của một người, cắt bỏ các khoảng trắng bên phải, trái và các khoảng trắng không có nghĩa trong chuỗi. In ra màn hình toàn bộ họ tên người đó dưới dạng chữ hoa, chữ thường.
11. Viết chương trình nhập vào một danh sách họ và tên của n người theo kiểu chữ thường, đổi các chữ cái đầu của họ, tên và chữ lót của mỗi người thành chữ hoa. In kết quả lên màn hình.

12. Viết chương trình nhập vào một danh sách họ và tên của n người, tách tên từng người ra khỏi họ và tên rồi sắp xếp danh sách tên theo thứ tự từ điển. In danh sách họ và tên sau khi đã sắp xếp.

Chương IX: KIỂU CẤU TRÚC

Học xong chương này, sinh viên sẽ nắm được các vấn đề sau:

- *Khái niệm về kiểu cấu trúc.*
- *Cách sử dụng kiểu cấu trúc.*
- *Con trỏ cấu trúc.*

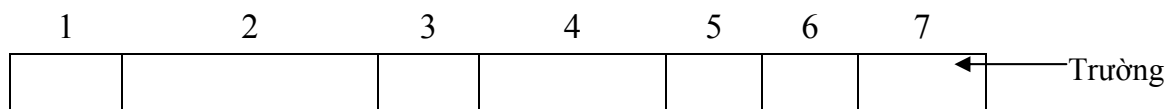
I. KIỂU CẤU TRÚC TRONG C

I.1 Khái niệm

Kiểu cấu trúc (Structure) là kiểu dữ liệu bao gồm nhiều thành phần có kiểu khác nhau, mỗi thành phần được gọi là một trường (field)

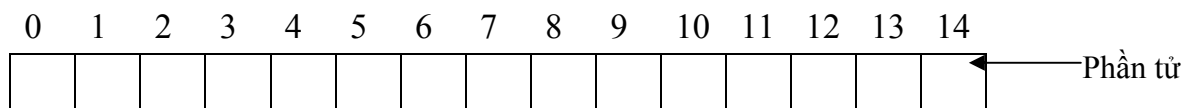
Sự khác biệt giữa kiểu cấu trúc và kiểu mảng là: các phần tử của mảng là cùng kiểu còn các phần tử của kiểu cấu trúc có thể có kiểu khác nhau.

Hình ảnh của kiểu cấu trúc được minh họa:



Đây là cấu trúc có 7 trường

Còn kiểu mảng có dạng:



Đây là mảng có 15 phần tử

I.2 Định nghĩa kiểu cấu trúc

Cách 1:

```
struct <Tên cấu trúc>
{
    <Kiểu> <Trường 1>;
    <Kiểu> <Trường 2>;
    .....
    <Kiểu> <Trường n>;
};
```

Cách 2: Sử dụng từ khóa typedef để định nghĩa kiểu:

```
typedef struct
{
    <Kiểu> <Trường 1>;
    <Kiểu> <Trường 2>;
    .....
```

```

    <Kiểu> <Trường n>;
} <Tên cấu trúc>;

```

Trong đó:

- <Tên cấu trúc>: là một tên được đặt theo quy tắc đặt tên của danh biểu; tên này mang ý nghĩa sẽ là tên kiểu cấu trúc.

- <Kiểu> <Trường i> (i=1..n): mỗi trường trong cấu trúc có dữ liệu thuộc kiểu gì (tên của trường phải là một tên được đặt theo quy tắc đặt tên của danh biểu).

Ví dụ 1: Để quản lý ngày, tháng, năm của một ngày trong năm ta có thể khai báo kiểu cấu trúc gồm 3 thông tin: ngày, tháng, năm.

<pre> struct NgayThang { unsigned char Ngay; unsigned char Thang; unsigned int Nam; }; </pre>	<pre> typedef struct { unsigned char Ngay; unsigned char Thang; unsigned int Nam; } NgayThang; </pre>
---	---

Ví dụ 2: Mỗi sinh viên cần được quản lý bởi các thông tin: mã số sinh viên, họ tên, ngày tháng năm sinh, giới tính, địa chỉ thường trú. Lúc này ta có thể khai báo một struct gồm các thông tin trên.

<pre> struct SinhVien { char MSSV[10]; char HoTen[40]; struct NgayThang NgaySinh; int Phai; char DiaChi[40]; }; </pre>	<pre> typedef struct { char MSSV[10]; char HoTen[40]; NgayThang NgaySinh; int Phai; char DiaChi[40]; } SinhVien; </pre>
--	---

I.3 Khai báo biến cấu trúc

Việc khai báo biến cấu trúc cũng tương tự như khai báo biến thuộc kiểu dữ liệu chuẩn.

Cú pháp:

- Đối với cấu trúc được định nghĩa theo cách 1:

```
struct <Tên cấu trúc> <Biến 1> [, <Biến 2>...];
```
- Đối với các cấu trúc được định nghĩa theo cách 2:

```
<Tên cấu trúc> <Biến 1> [, <Biến 2>...];
```

Ví dụ: Khai báo biến NgaySinh có kiểu cấu trúc NgayThang; biến SV có kiểu cấu trúc SinhVien.

<pre> struct NgayThang NgaySinh; struct SinhVien SV; </pre>	<pre> NgayThang NgaySinh; SinhVien SV; </pre>
---	---

II. CÁC THAO TÁC TRÊN BIẾN KIỂU CẤU TRÚC

II.1 Truy xuất đến từng trường của biến cấu trúc

Cú pháp: <Biến cấu trúc>.<Tên trường>

Khi sử dụng cách truy xuất theo kiểu này, các thao tác trên <Biến cấu trúc>.<Tên trường> giống như các thao tác trên các biến của kiểu dữ liệu của <Tên trường>.

Ví dụ : Viết chương trình cho phép đọc dữ liệu từ bàn phím cho biến mẫu tin SinhVien và in biến mẫu tin đó lên màn hình:

```
#include<conio.h>
#include<stdio.h>
#include<string.h>

typedef struct
{
    unsigned char Ngay;
    unsigned char Thang;
    unsigned int Nam;
} NgayThang;
typedef struct
{
    char MSSV[10];
    char HoTen[40];
    NgayThang NgaySinh;
    int Phai;
    char DiaChi[40];
} SinhVien;

/* Hàm in lên màn hình 1 mẫu tin SinhVien*/
void InSV(SinhVien s)
{
    printf("MSSV: | Ho va ten | Ngay Sinh | Dia chi\n");
    printf("%s | %s | %d-%d-%d | %s\n",s.MSSV,s.HoTen,
        s.NgaySinh.Ngay,s.NgaySinh.Thang,s.NgaySinh.Nam,s.DiaChi);
}

int main()
{
    SinhVien SV, s;
    printf("Nhap MSSV: ");gets(SV.MSSV);
    printf("Nhap Ho va ten: ");gets(SV.HoTen);
    printf("Sinh ngay: ");scanf("%d",&SV.NgaySinh.Ngay);
    printf("Thang: ");scanf("%d",&SV.NgaySinh.Thang);
    printf("Nam: ");scanf("%d",&SV.NgaySinh.Nam);
    printf("Gioi tinh (0: Nu), (1: Nam): ");scanf("%d",&SV.Phai);
    fflush();
    printf("Dia chi: ");gets(SV.DiaChi);
    InSV(SV);

    s=SV; /* Gán trị cho mẫu tin s*/
    InSV(s);
    getch();
    return 0;
}
```

```

c:\ TC.EXE
Nhap MSSU: 1040393
Nhap Ho va ten: Lam Nhat Tien
Sinh ngay: 29
Thang: 8
Nam: 1986
Gioi tinh <0: Nu>, <1: Nam>: 1
Dia chi: 1 Ly Tu Trong
MSSU:      |      Ho va ten      |      Ngay Sinh      |      Dia chi
1040393   |      Lam Nhat Tien   |      29-8-1986      |      1 Ly Tu Trong
MSSU:      |      Ho va ten      |      Ngay Sinh      |      Dia chi
1040393   |      Lam Nhat Tien   |      29-8-1986      |      1 Ly Tu Trong

```

Lưu ý:

- Các biến cấu trúc có thể gán cho nhau. Thực chất đây là thao tác trên toàn bộ cấu trúc không phải trên một trường riêng rẽ nào. Chương trình trên dòng `s=SV` là một ví dụ.

- Với các biến kiểu cấu trúc ta không thể thực hiện được các thao tác sau đây:

- Sử dụng các hàm xuất nhập trên biến cấu trúc.
- Các phép toán quan hệ, các phép toán số học và logic.

Ví dụ: Nhập vào hai số phức và tính tổng của chúng. Ta biết rằng số phức là một cặp (a,b) trong đó a, b là các số thực, a gọi là phần thực, b là phần ảo. (Đôi khi người ta cũng viết số phức dưới dạng $a + ib$ trong đó i là một đơn vị ảo có tính chất $i^2=-1$). Gọi số phức $c1=(a1, b1)$ và $c2=(a2,b2)$ khi đó tổng của hai số phức $c1$ và $c2$ là một số phức $c3$ mà $c3=(a1+a2, b1+b2)$. Với hiểu biết như vậy ta có thể xem mỗi số phức là một cấu trúc có hai trường, một trường biểu diễn cho phần thực, một trường biểu diễn cho phần ảo. Việc tính tổng của hai số phức được tính bằng cách lấy phần thực cộng với phần thực và phần ảo cộng với phần ảo.

```

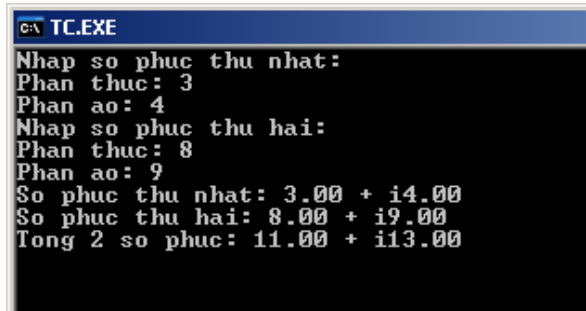
#include<conio.h>
#include<stdio.h>
#include<string.h>
typedef struct
{
    float Thuc;
    float Ao;
} SoPhuc;
/* Hàm in số phức lên màn hình*/
void InSoPhuc(SoPhuc p)
{
    printf("%.2f + i%.2f\n",p.Thuc,p.Ao);
}

int main()
{
    SoPhuc p1,p2,p;
    clrscr();
    printf("Nhap so phuc thu nhat:\n");
    printf("Phan thuc: ");scanf("%f",&p1.Thuc);
    printf("Phan ao: ");scanf("%f",&p1.Ao);
    printf("Nhap so phuc thu hai:\n");
    printf("Phan thuc: ");scanf("%f",&p2.Thuc);
    printf("Phan ao: ");scanf("%f",&p2.Ao);
    printf("So phuc thu nhat: ");
    InSoPhuc(p1);
    printf("So phuc thu hai: ");

```

```
InSoPhuc(p2);  
p.Thuc = p1.Thuc+p2.Thuc;  
p.Ao = p1.Ao + p2.Ao;  
printf("Tong 2 so phuc: ");  
InSoPhuc(p);  
getch();  
return 0;  
}
```

Kết quả thực hiện chương trình:



```
c:\ TC.EXE  
Nhap so phuc thu nhat:  
Phan thuc: 3  
Phan ao: 4  
Nhap so phuc thu hai:  
Phan thuc: 8  
Phan ao: 9  
So phuc thu nhat: 3.00 + i4.00  
So phuc thu hai: 8.00 + i9.00  
Tong 2 so phuc: 11.00 + i13.00
```

II.2 Khởi tạo cấu trúc

Việc khởi tạo cấu trúc có thể được thực hiện trong lúc khai báo biến cấu trúc. Các trường của cấu trúc được khởi tạo được đặt giữa 2 dấu { và }, chúng được phân cách nhau bởi dấu phẩy (,).

Ví dụ: Khởi tạo biến cấu trúc NgaySinh:

```
struct NgayThang NgaySinh = {29, 8, 1986};
```

III. CON TRỎ CẤU TRÚC

III.1 Khai báo

Việc khai báo một biến con trỏ kiểu cấu trúc cũng tương tự như khi khai báo một biến con trỏ khác, nghĩa là đặt thêm dấu * vào phía trước tên biến.

Cú pháp: struct <Tên cấu trúc> * <Tên biến con trỏ>;

Ví dụ: Ta có thể khai báo một con trỏ cấu trúc kiểu NgayThang như sau:

```
struct NgayThang *p;  
/* NgayThang *p; // Nếu có định nghĩa kiểu */
```

III.2 Sử dụng các con trỏ kiểu cấu trúc

Khi khai báo biến con trỏ cấu trúc, biến con trỏ chưa có địa chỉ cụ thể. Lúc này nó chỉ mới được cấp phát 2 byte để lưu giữ địa chỉ và được ghi nhận là con trỏ chỉ đến 1 cấu trúc, nhưng chưa chỉ đến 1 đối tượng cụ thể. Muốn thao tác trên con trỏ cấu trúc hợp lệ, cũng tương tự như các con trỏ khác, ta phải:

- Cấp phát một vùng nhớ cho nó (sử dụng hàm malloc() hay calloc)
- Hoặc, cho nó quản lý địa chỉ của một biến cấu trúc nào đó.

Ví dụ: Sau khi khởi tạo giá trị của cấu trúc:

```
struct NgayThang Ngay = {29, 8, 1986};
```

```
p = &Ngay;
```

lúc này biến con trỏ p đã chứa địa chỉ của Ngay.

III.3 Truy cập các thành phần của cấu trúc đang được quản lý bởi con trỏ

Để truy cập đến từng trường của 1 cấu trúc thông qua con trỏ của nó, ta sử dụng toán tử dấu mũ tên (->: dấu - và dấu >).

Ngoài ra, ta vẫn có thể sử dụng đến phép toán * để truy cập vùng dữ liệu đang được quản lý bởi con trỏ cấu trúc để lấy thông tin cần thiết.

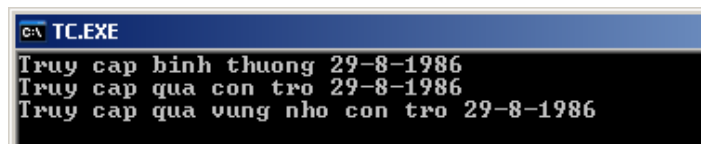
Ví dụ: Sử dụng con trỏ cấu trúc.

```
#include<conio.h>
#include<stdio.h>

typedef struct
{
    unsigned char Ngay;
    unsigned char Thang;
    unsigned int Nam;
} NgayThang;

int main()
{
    NgayThang Ng={29,8,1986};
    NgayThang *p;
    clrscr();
    p=&Ng;
    printf("Truy cap binh thuong %d-%d-%d\n",
        Ng.Ngay,Ng.Thang,Ng.Nam);
    printf("Truy cap qua con tro %d-%d-%d\n",
        p->Ngay,p->Thang,p->Nam);
    printf("Truy cap qua vung nho con tro %d-%d-%d\n",
        (*p).Ngay,(*p).Thang,(*p).Nam);
    getch();
    return 0;
}
```

Kết quả:



```
C:\ TC.EXE
Truy cap binh thuong 29-8-1986
Truy cap qua con tro 29-8-1986
Truy cap qua vung nho con tro 29-8-1986
```

IV. BÀI TẬP

IV.1 Mục đích yêu cầu

Làm quen và biết cách sử dụng kiểu dữ liệu cấu trúc kết hợp với các kiểu dữ liệu đã học. Phân biệt kiểu dữ liệu mảng và kiểu cấu trúc. Thực hiện các bài tập trong phần nội dung.

IV.2 Nội dung

1. Hãy định nghĩa kiểu:

```
struct Hosoi{\n    char HoTen[40];\n    float Diem;\n    char Loai[10];\n};
```

Viết chương trình nhập vào họ tên, điểm của n học sinh. Xếp loại văn hóa theo cách sau:

Điểm	Xếp loại
9, 10	Giỏi
7, 8	Khá
5, 6	Trung bình
dưới 5	Không đạt

In danh sách lên màn hình theo dạng sau:

XEP LOAI VAN HOA		
HO VA TEN	DIEM	XEPLOAI
Nguyen Van A	7	Kha
Ho Thi B	5	Trung binh
Dang Kim C	4	Khong dat

2. Xem một phân số là một cấu trúc có hai trường là tử số và mẫu số. Hãy viết chương trình thực hiện các phép toán cộng, trừ, nhân, chia hai phân số. (Các kết quả phải tối giản).

3. Tạo một danh sách cán bộ công nhân viên, mỗi người người xem như một cấu trúc bao gồm các trường Ho, Ten, Luong, Tuoi, Dchi. Nhập một số người vào danh sách, sắp xếp tên theo thứ tự từ điển, in danh sách đã sắp xếp theo mẫu sau:

DANH SACH CAN BO CONG NHAN VIEN

STT	HO VA TEN	LUONG	TUOI	DIACHI
1	Nguyen Van Tho	333.00	26	Can
2	Dang Kim B	290.00		
23	Vinh Long			

Chương 10

Kiểu Tập Tin

Học xong chương này, sinh viên sẽ nắm rõ các vấn đề sau:

- Một số khái niệm về tập tin?
- Các bước thao tác với tập tin.
- Một số hàm truy xuất tập tin văn bản.
- Một số hàm truy xuất tập tin nhị phân.

I. MỘT SỐ KHÁI NIỆM VỀ TẬP TIN

Đối với các kiểu dữ liệu ta đã biết như kiểu số, kiểu mảng, kiểu cấu trúc thì dữ liệu được tổ chức trong bộ nhớ trong (RAM) của máy tính nên khi kết thúc việc thực hiện chương trình thì dữ liệu cũng bị mất; khi cần chúng ta bắt buộc phải nhập lại từ bàn phím. Điều đó vừa mất thời gian vừa không giải quyết được các bài toán với số liệu lớn. Để giải quyết vấn đề, người ta đưa ra kiểu tập tin (file) cho phép lưu trữ dữ liệu ở bộ nhớ ngoài (đĩa). Khi kết thúc chương trình thì dữ liệu vẫn còn do đó chúng ta có thể sử dụng nhiều lần. Một đặc điểm khác của kiểu tập tin là kích thước lớn với số lượng các phần tử không hạn chế (chỉ bị hạn chế bởi dung lượng của bộ nhớ ngoài).

Có 3 loại dữ liệu kiểu tập tin:

- Tập tin văn bản (Text File): là loại tập tin dùng để ghi các ký tự lên đĩa, các ký tự này được lưu trữ dưới dạng mã Ascii. Điểm đặc biệt là dữ liệu của tập tin được lưu trữ thành các dòng, mỗi dòng được kết thúc bằng ký tự xuống dòng (new line), ký hiệu '\n'; ký tự này là sự kết hợp của 2 ký tự CR (Carriage Return - Về đầu dòng, mã Ascii là 13) và LF (Line Feed - Xuống dòng, mã Ascii là 10). Mỗi tập tin được kết thúc bởi ký tự EOF (End Of File) có mã Ascii là 26 (xác định bởi tổ hợp phím Ctrl + Z).

Tập tin văn bản chỉ có thể truy xuất theo kiểu tuần tự.

- Tập tin định kiểu (Typed File): là loại tập tin bao gồm nhiều phần tử có cùng kiểu: char, int, long, cấu trúc... và được lưu trữ trên đĩa dưới dạng một chuỗi các byte liên tục.

- Tập tin không định kiểu (Untyped File): là loại tập tin mà dữ liệu của chúng gồm các cấu trúc dữ liệu mà người ta không quan tâm đến nội dung hoặc kiểu của nó, chỉ lưu ý đến các yếu tố vật lý của tập tin như độ lớn và các yếu tố tác động lên tập tin mà thôi.

Biến tập tin: là một biến thuộc kiểu dữ liệu tập tin dùng để đại diện cho một tập tin. Dữ liệu chứa trong một tập tin được truy xuất qua các thao tác với thông số là biến tập tin đại diện cho tập tin đó.

Con trỏ tập tin: Khi một tập tin được mở ra để làm việc, tại mỗi thời điểm, sẽ có một vị trí của tập tin mà tại đó việc đọc/ghi thông tin sẽ xảy ra. Người ta hình dung có một con trỏ đang chỉ đến vị trí đó và đặt tên nó là con trỏ tập tin.

Sau khi đọc/ghi xong dữ liệu, con trỏ sẽ chuyển dịch thêm một phần tử về phía cuối tập tin. Sau phần tử dữ liệu cuối cùng của tập tin là dấu kết thúc tập tin EOF (End Of File).

II. CÁC THAO TÁC TRÊN TẬP TIN

Muốn thao tác trên tập tin, ta phải lần lượt làm theo các bước:

- Khai báo biến tập tin.
- Mở tập tin bằng hàm fopen().
- Thực hiện các thao tác xử lý dữ liệu của tập tin bằng các hàm đọc/ghi dữ liệu.
- Đóng tập tin bằng hàm fclose().

Ở đây, ta thao tác với tập tin nhờ các hàm được định nghĩa trong thư viện stdio.h.

II.1. Khai báo biến tập tin

Cú pháp: FILE <Danh sách các biến con trỏ>

Các biến trong danh sách phải là các con trỏ và được phân cách bởi dấu phẩy(,).

Ví dụ: FILE *f1,*f2;

II.2. Mở tập tin

Cú pháp: FILE *fopen(char *Path, const char *Mode)

Trong đó:

- Path: chuỗi chỉ đường dẫn đến tập tin trên đĩa.

- Type: chuỗi xác định cách thức mà tập tin sẽ mở. Các giá trị có thể của *Mode*:

Chế độ	Ý nghĩa
r	Mở tập tin văn bản để đọc
w	Tạo ra tập tin văn bản mới để ghi
a	Nối vào tập tin văn bản
rb	Mở tập tin nhị phân để đọc
wb	Tạo ra tập tin nhị phân để ghi
ab	Nối vào tập tin nhị phân
r+	Mở một tập tin văn bản để đọc/ghi
w+	Tạo ra tập tin văn bản để đọc ghi
a+	Nối vào hay tạo mới tập tin văn bản để đọc/ghi
r+b	Mở ra tập tin nhị phân để đọc/ghi
w+b	Tạo ra tập tin nhị phân để đọc/ghi
a+b	Nối vào hay tạo mới tập tin nhị phân

- Hàm fopen trả về một con trỏ tập tin. Chương trình của ta không thể thay đổi giá trị của con trỏ này. Nếu có một lỗi xuất hiện trong khi mở tập tin thì hàm này trả về con trỏ NULL.

Ví dụ: Mở một tập tin tên TEST.txt để ghi.

```
FILE *f;  
f = fopen("TEST.txt", "w");  
if (f!=NULL)  
{  
    /* Các câu lệnh để thao tác với tập tin*/  
}
```

```
/* Đóng tập tin*/
```

```
}
```

Trong ví dụ trên, ta có sử dụng câu lệnh kiểm tra điều kiện để xác định mở tập tin có thành công hay không?.

Nếu mở tập tin để ghi, nếu tập tin đã tồn tại rồi thì tập tin sẽ bị xóa và một tập tin mới được tạo ra. Nếu ta muốn ghi nối dữ liệu, ta phải sử dụng chế độ “a”. Khi mở với chế độ đọc, tập tin phải tồn tại rồi, nếu không một lỗi sẽ xuất hiện.

II.3. Đóng tập tin

Hàm `fclose()` được dùng để đóng tập tin được mở bởi hàm `fopen()`. Hàm này sẽ ghi dữ liệu còn lại trong vùng đệm vào tập tin và đóng lại tập tin.

Cú pháp: `int fclose(FILE *f)`

Trong đó `f` là con trỏ tập tin được mở bởi hàm `fopen()`. Giá trị trả về của hàm là 0 báo rằng việc đóng tập tin thành công. Hàm trả về EOF nếu có xuất hiện lỗi.

Ngoài ra, ta còn có thể sử dụng hàm `fcloseall()` để đóng tất cả các tập tin lại.

Cú pháp: `int fcloseall()`

Kết quả trả về của hàm là tổng số các tập tin được đóng lại. Nếu không thành công, kết quả trả về là EOF.

II.4. Kiểm tra đến cuối tập tin hay chưa?

Cú pháp: `int feof(FILE *f)`

Ý nghĩa: Kiểm tra xem đã chạm tới cuối tập tin hay chưa và trả về EOF nếu cuối tập tin được chạm tới, ngược lại trả về 0.

II.5 Di chuyển con trỏ tập tin về đầu tập tin - Hàm `rewind()`

Khi ta đang thao tác một tập tin đang mở, con trỏ tập tin luôn di chuyển về phía cuối tập tin. Muốn cho con trỏ quay về đầu tập tin như khi mở nó, ta sử dụng hàm `rewind()`.

Cú pháp: `void rewind(FILE *f)`

III. TRUY CẬP TẬP TIN VĂN BẢN

III.1. Ghi dữ liệu lên tập tin văn bản

III.1.1 Hàm `putc()`

Hàm này được dùng để ghi một ký tự lên một tập tin văn bản đang được mở để làm việc.

Cú pháp: `int putc(int c, FILE *f)`

Trong đó, tham số `c` chứa mã Ascii của một ký tự nào đó. Mã này được ghi lên tập tin liên kết với con trỏ `f`. Hàm này trả về EOF nếu gặp lỗi.

III.1.2 Hàm `fputs()`

Hàm này dùng để ghi một chuỗi ký tự chứa trong vùng đệm lên tập tin văn bản.

Cú pháp: `int fputs(const char *buffer, FILE *f)`

Trong đó, buffer là con trỏ có kiểu char chỉ đến vị trí đầu tiên của chuỗi ký tự được ghi vào. Hàm này trả về giá trị 0 nếu buffer chứa chuỗi rỗng và trả về EOF nếu gặp lỗi.

III.1.3 Hàm fprintf()

Hàm này dùng để ghi dữ liệu có định dạng lên tập tin văn bản.

Cú pháp: `fprintf(FILE *f, const char *format, varexpr)`

Trong đó: `format`: chuỗi định dạng (giống với các định dạng của hàm `printf()`), `varexpr`: danh sách các biểu thức, mỗi biểu thức cách nhau dấu phẩy (,).

Định dạng	Ý nghĩa
<code>%d</code>	Ghi số nguyên
<code>%[.số chữ số thập phân] f</code>	Ghi số thực có <số chữ số thập phân> theo quy tắc làm tròn số.
<code>%o</code>	Ghi số nguyên hệ bát phân
<code>%x</code>	Ghi số nguyên hệ thập lục phân
<code>%c</code>	Ghi một ký tự
<code>%s</code>	Ghi chuỗi ký tự
<code>%e</code> hoặc <code>%E</code> hoặc <code>%g</code> hoặc <code>%G</code>	Ghi số thực dạng khoa học (nhân 10 mũ x)

Ví dụ: Viết chương trình ghi chuỗi ký tự lên tập tin văn bản `D:\\Baihat.txt`

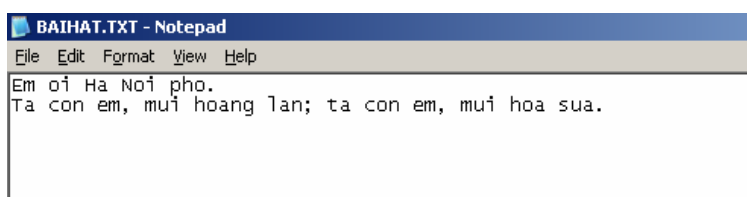
```
#include<stdio.h>
#include<conio.h>

int main()
{
    FILE *f;
    clrscr();
    f=fopen("D:\\Baihat.txt","r+");

    if (f!=NULL)
    {
        fputs("Em oi Ha Noi pho.\n",f);
        fputs("Ta con em, mui hoang lan; ta con em, mui hoa sua.",f);
        fclose(f);
    }
    getch();

    return 0;
}
```

Nội dung tập tin `Baihat.txt` khi được mở bằng trình soạn thảo văn bản Notepad.



III.2. Đọc dữ liệu từ tập tin văn bản

III.2.1 Hàm getc()

Hàm này dùng để đọc dữ liệu từ tập tin văn bản đang được mở để làm việc.

Cú pháp: `int getc(FILE *f)`

Hàm này trả về mã Ascii của một ký tự nào đó (kể cả EOF) trong tập tin liên kết với con trỏ f.

III.2.2 Hàm fgetc()

Cú pháp: `char *fgetc(char *buffer, int n, FILE *f)`

Hàm này được dùng để đọc một chuỗi ký tự từ tập tin văn bản đang được mở ra và liên kết với con trỏ f cho đến khi đọc đủ n ký tự hoặc gặp ký tự xuống dòng '\n' (ký tự này cũng được đưa vào chuỗi kết quả) hay gặp ký tự kết thúc EOF (ký tự này không được đưa vào chuỗi kết quả).

Trong đó:

- buffer (vùng đệm): con trỏ có kiểu char chỉ đến cùng nhớ đủ lớn chứa các ký tự nhận được.

- n: giá trị nguyên chỉ độ dài lớn nhất của chuỗi ký tự nhận được.

- f: con trỏ liên kết với một tập tin nào đó.

- Ký tự NULL ('\0') tự động được thêm vào cuối chuỗi kết quả lưu trong vùng đệm.

- Hàm trả về địa chỉ đầu tiên của vùng đệm khi không gặp lỗi và chưa gặp ký tự kết thúc EOF. Ngược lại, hàm trả về giá trị NULL.

III.2.3 Hàm fscanf()

Hàm này dùng để đọc dữ liệu từ tập tin văn bản vào danh sách các biến theo định dạng.

Cú pháp: `fscanf(FILE *f, const char *format, varlist)`

Trong đó: format: chuỗi định dạng (giống hàm scanf()); varlist: danh sách các biến mỗi biến cách nhau dấu phẩy (,).

Ví dụ: Viết chương trình chép tập tin D:\Baihat.txt ở trên sang tập tin D:\Baica.txt.

```
#include<stdio.h>
#include<conio.h>

int main()
{
    FILE *f1,*f2;
    clrscr();
    f1=fopen("D:\\Baihat.txt","rt");
    f2=fopen("D:\\Baica.txt","wt");
    if (f1!=NULL && f2!=NULL)
    {
        int ch=fgetc(f1);
        while (! feof(f1))
        {
            fputc(ch,f2);
            ch=fgetc(f1);
        }
    }
}
```

```
        fcloseall();  
    }  
    getch();  
    return 0;  
}
```

IV. TRUY CẬP TẬP TIN NHỊ PHÂN

IV.1 Ghi dữ liệu lên tập tin nhị phân - Hàm fwrite()

Cú pháp: `size_t fwrite(const void *ptr, size_t size, size_t n, FILE *f)`

Trong đó:

- ptr: con trỏ chỉ đến vùng nhớ chứa thông tin cần ghi lên tập tin.
- n: số phần tử sẽ ghi lên tập tin.
- size: kích thước của mỗi phần tử.
- f: con trỏ tập tin đã được mở.
- Giá trị trả về của hàm này là số phần tử được ghi lên tập tin. Giá trị này bằng n trừ khi xuất hiện lỗi.

IV.2 Đọc dữ liệu từ tập tin nhị phân - Hàm fread()

Cú pháp: `size_t fread(const void *ptr, size_t size, size_t n, FILE *f)`

Trong đó:

- ptr: con trỏ chỉ đến vùng nhớ sẽ nhận dữ liệu từ tập tin.
- n: số phần tử được đọc từ tập tin.
- size: kích thước của mỗi phần tử.
- f: con trỏ tập tin đã được mở.
- Giá trị trả về của hàm này là số phần tử đã đọc được từ tập tin. Giá trị này bằng n hay nhỏ hơn n nếu đã chạm đến cuối tập tin hoặc có lỗi xuất hiện..

IV.3 Di chuyển con trỏ tập tin - Hàm fseek()

Việc ghi hay đọc dữ liệu từ tập tin sẽ làm cho con trỏ tập tin dịch chuyển một số byte, đây chính là kích thước của kiểu dữ liệu của mỗi phần tử của tập tin.

Khi đóng tập tin rồi mở lại nó, con trỏ luôn ở vị trí ngay đầu tập tin. Nhưng nếu ta sử dụng kiểu mở tập tin là “a” để ghi nối dữ liệu, con trỏ tập tin sẽ di chuyển đến vị trí cuối cùng của tập tin này.

Ta cũng có thể điều khiển việc di chuyển con trỏ tập tin đến vị trí chỉ định bằng hàm fseek().

Cú pháp: `int fseek(FILE *f, long offset, int whence)`

Trong đó:

- f: con trỏ tập tin đang thao tác.
- offset: số byte cần dịch chuyển con trỏ tập tin kể từ vị trí trước đó. Phần tử đầu tiên là vị trí 0.
- whence: vị trí bắt đầu để tính offset, ta có thể chọn điểm xuất phát là:

0	SEEK_SET	Vị trí đầu tập tin
---	----------	--------------------

1	SEEK_CUR	Vị trí hiện tại của con trỏ tập tin
2	SEEK_END	Vị trí cuối tập tin

- Kết quả trả về của hàm là 0 nếu việc di chuyển thành công. Nếu không thành công, 1 giá trị khác 0 (đó là 1 mã lỗi) được trả về.

IV.4 Ví dụ

Ví dụ 1: Viết chương trình ghi lên tập tin CacSo.Dat 3 giá trị số (thực, nguyên, nguyên dài). Sau đó đọc các số từ tập tin vừa ghi và hiển thị lên màn hình.

```
#include<stdio.h>
#include<conio.h>

int main()
{
    FILE *f;
    clrscr();
    f=fopen("D:\\CacSo.txt", "wb");
    if (f!=NULL)
    {
        double d=3.14;
        int i=101;
        long l=54321;
        fwrite(&d, sizeof(double), 1, f);
        fwrite(&i, sizeof(int), 1, f);
        fwrite(&l, sizeof(long), 1, f);
        /* Doc tu tap tin*/
        rewind(f);
        fread(&d, sizeof(double), 1, f);
        fread(&i, sizeof(int), 1, f);
        fread(&l, sizeof(long), 1, f);
        printf("Cac ket qua la: %f %d %ld", d, i, l);
        fclose(f);
    }
    getch();
    return 0;
}
```

Ví dụ 2: Mỗi sinh viên cần quản lý ít nhất 2 thông tin: mã sinh viên và họ tên. Viết chương trình cho phép lựa chọn các chức năng: nhập danh sách sinh viên từ bàn phím rồi ghi lên tập tin SinhVien.dat, đọc dữ liệu từ tập tin SinhVien.dat rồi hiển thị danh sách lên màn hình, tìm kiếm họ tên của một sinh viên nào đó dựa vào mã sinh viên nhập từ bàn phím.

Ta nhận thấy rằng mỗi phần tử của tập tin SinhVien.Dat là một cấu trúc có 2 trường: mã và họ tên. Do đó, ta cần khai báo cấu trúc này và sử dụng các hàm đọc/ghi tập tin nhị phân với kích thước mỗi phần tử của tập tin là chính kích thước cấu trúc đó.

```
#include<stdio.h>
#include<conio.h>
#include<string.h>

typedef struct
{
    char Ma[10];
    char HoTen[40];
}
```



```
    } SinhVien;

void WriteFile(char *FileName)
{
    FILE *f;
    int n,i;
    SinhVien sv;
    f=fopen(FileName,"ab");
    printf("Nhap bao nhieu sinh vien? ");scanf("%d",&n);
    fflush(stdin);
    for(i=1;i<=n;i++)
    {
        printf("Sinh vien thu %i\n",i);
        printf(" - MSSV: ");gets(sv.Ma);
        printf(" - Ho ten: ");gets(sv.HoTen);
        fwrite(&sv,sizeof(sv),1,f);
        fflush(stdin);
    }
    fclose(f);
    printf("Bam phim bat ky de tiep tuc");
    getch();
}

void ReadFile(char *FileName)
{
    FILE *f;
    SinhVien sv;
    f=fopen(FileName,"rb");
    printf("      MSSV      |      Ho va ten\n");
    fread(&sv,sizeof(sv),1,f);
    while (!feof(f))
    {
        printf("   %s   |   %s\n",sv.Ma,sv.HoTen);
        fread(&sv,sizeof(sv),1,f);
    }
    fclose(f);
    printf("Bam phim bat ky de tiep tuc!!!");
    getch();
}

void Search(char *FileName)
{
    char MSSV[10];
    FILE *f;
    int Found=0;
    SinhVien sv;

    fflush(stdin);
    printf("Ma so sinh vien can tim: ");gets(MSSV);
    f=fopen(FileName,"rb");
    while (!feof(f) && Found==0)
    {
        fread(&sv,sizeof(sv),1,f);
        if (strcmp(sv.Ma,MSSV)==0) Found=1;
    }
    fclose(f);
    if (Found == 1)
        printf("Tim thay SV co ma %s. Ho ten la: %s",sv.Ma,sv.HoTen);
}
```

```

else
    printf("Tim khong thay sinh vien co ma %s",MSSV);

    printf("\nBam phim bat ky de tiep tuc!!!");
    getch();
}

int main()
{
    int c;
    for (;;)
    {
        clrscr();
        printf("1. Nhap DSSV\n");
        printf("2. In DSSV\n");
        printf("3. Tim kiem\n");
        printf("4. Thoat\n");
        printf("Ban chon 1, 2, 3, 4: "); scanf("%d",&c);
        if(c==1)
            WriteFile("d:\\SinhVien.Dat");
        else if (c==2)
            ReadFile("d:\\SinhVien.Dat");
        else if (c==3)
            Search("d:\\SinhVien.Dat");
        else break;
    }
    return 0;
}

```

Ngoài ra thư viện `stdio.h` còn định nghĩa một số hàm khác cho phép thao tác với tập tin, sinh viên có thể tham khảo trong phần trợ giúp.

V. BÀI TẬP

V.1 Mục đích yêu cầu

Nắm vững cách sử dụng kiểu dữ liệu tập tin. Phân biệt nó với tất cả các kiểu dữ liệu có cấu trúc đã học. Làm quen và biết cách thao tác trên tập tin. Vận dụng các kiến thức đã học viết các chương trình trong phần nội dung.

V.2 Nội dung

- Viết chương trình quản lý một tập tin văn bản theo các yêu cầu:
 - Nhập từ bàn phím nội dung một văn bản sau đó ghi vào đĩa.
 - Đọc từ đĩa nội dung văn bản vừa nhập và in lên màn hình.
 - Đọc từ đĩa nội dung văn bản vừa nhập, in nội dung đó lên màn hình và cho phép nối thêm thông tin vào cuối tập tin đó.
- Viết chương trình cho phép thống kê số lần xuất hiện của các ký tự là chữ ('A'..'Z','a'..'z') trong một tập tin văn bản.
- Viết chương trình đếm số từ và số dòng trong một tập tin văn bản.
- Viết chương trình nhập từ bàn phím và ghi vào 1 tập tin tên là DMHH.DAT với mỗi phần tử của tập tin là 1 cấu trúc bao gồm các trường: Ma (mã hàng: char[5]), Ten (Tên

hàng: char[20]). Kết thúc việc nhập bằng cách gõ ENTER vào Ma. Ta sẽ dùng tập tin này để giải mã hàng hóa cho tập tin DSHH.DAT sẽ đề cập trong bài 5.

5. Viết chương trình cho phép nhập từ bàn phím và ghi vào 1 tập tin tên DSHH.Dat với mỗi phần tử của tập tin là một cấu trúc bao gồm các trường : mh (mã hàng: char[5]), sl (số lượng : int), dg (đơn giá: float), st (Số tiền: float) theo yêu cầu:

- Mỗi lần nhập một cấu trúc

- Trước tiên nhập mã hàng (mh), đưa mh so sánh với Ma trong tập tin DMHH.DAT đã được tạo ra bởi bài tập 1, nếu mh=ma thì in tên hàng ngay bên cạnh mã hàng.

- Nhập số lượng (sl).

- Nhập đơn giá (dg).

- Tính số tiền = số lượng * đơn giá.

Kết thúc việc nhập bằng cách đánh ENTER vào mã hàng. Sau khi nhập xong yêu cầu in toàn bộ danh sách hàng hóa có sự giải mã về tên hàng theo mẫu sau:

STT	MA HÀNG	TEN HÀNG	SO LG	DON GIA	SO TIEN
1	a0101	Duong cat trang	25	10000.00	250000.00
2	b0101	Sua co gai Ha Lan	10	40000.00	400000.00

GIÁO TRÌNH LẬP TRÌNH C CĂN BẢN

MỤC LỤC

BÀI 1 :	NGÔN NGỮ LẬP TRÌNH & PHƯƠNG PHÁP LẬP TRÌNH.....	8
1.1	Mục tiêu	8
1.2	Lý thuyết.....	8
1.2.1	Ngôn ngữ lập trình (Programming Language).....	8
1.2.1.1	Thuật giải (Algorithm).....	8
1.2.1.2	Chương trình (Program)	8
1.2.1.3	Ngôn ngữ lập trình (Programming language)	9
1.2.2	Các bước lập trình	9
1.2.3	Kỹ thuật lập trình	9
1.2.3.1	I-P-O Cycle (Input-Process-Output Cycle) (Quy trình nhập-xử lý-xuất)	9
1.2.3.2	Sử dụng lưu đồ (Flowchart).....	10
BÀI 2 :	LÀM QUEN LẬP TRÌNH C QUA CÁC VÍ DỤ ĐƠN GIẢN.....	13
2.1	Mục tiêu	13
2.2	Nội dung.....	13
2.2.1	Khởi động và thoát BorlandC	13
2.2.1.1	Khởi động.....	13
2.2.1.2	Thoát.....	14
2.2.2	Các ví dụ đơn giản	14
2.2.2.1	Ví dụ 1	14
2.2.2.2	Ví dụ 2	16
2.2.2.3	Ví dụ 3	17
2.2.2.4	Ví dụ 4	17
BÀI 3 :	CÁC THÀNH PHẦN TRONG NGÔN NGỮ C.....	20
3.1	Mục tiêu	20
3.2	Nội dung.....	20
3.2.1	Từ khóa.....	20
3.2.2	Tên	20
3.2.3	Kiểu dữ liệu	20
3.2.4	Ghi chú	21
3.2.5	Khai báo biến	21
3.2.5.1	Tên biến.....	21
3.2.5.2	Khai báo biến	21
3.2.5.3	Vừa khai báo vừa khởi gán.....	22
3.2.5.4	Phạm vi của biến.....	22
BÀI 4 :	NHẬP/ XUẤT DỮ LIỆU	23
4.1	Mục tiêu	23
4.2	Nội dung.....	23

4.2.1	Hàm printf.....	23
4.2.2	Hàm scanf.....	26
4.3	Bài tập.....	27
BÀI 5 : CẤU TRÚC RỄ NHÁNH CÓ ĐIỀU KIỆN		28
5.1	Mục tiêu	28
5.2	Nội dung.....	28
5.2.1	Lệnh và khối lệnh.....	28
5.2.1.1	Lệnh.....	28
5.2.1.2	Khối lệnh	28
5.2.2	Lệnh if.....	28
5.2.2.1	Dạng 1 (if thiếu)	28
5.2.2.2	Dạng 2 (if đủ).....	32
5.2.2.3	Cấu trúc else if	35
5.2.2.4	Cấu trúc if lồng	39
5.2.3	Lệnh switch.....	43
5.2.3.1	Cấu trúc switch...case (switch thiếu)	43
5.2.3.2	Cấu trúc switch...case...default (switch đủ).....	46
5.2.3.3	Cấu trúc switch lồng	48
5.3	Bài tập.....	49
5.3.1	Sử dụng lệnh if	49
5.3.2	Sử dụng lệnh switch	50
5.4	Bài tập làm thêm.....	51
BÀI 6 : CẤU TRÚC VÒNG LẶP		52
6.1	Mục tiêu	52
6.2	Nội dung.....	52
6.2.1	Lệnh for	52
6.2.2	Lệnh break	57
6.2.3	Lệnh continue.....	57
6.2.4	Lệnh while	57
6.2.5	Lệnh do...while	59
6.2.6	Vòng lặp lồng nhau	61
6.2.7	So sánh sự khác nhau của các vòng lặp.....	62
6.3	Bài tập.....	62
BÀI 7 : HÀM		65
7.1	Mục tiêu	65
7.2	Nội dung.....	65
7.2.1	Các ví dụ về hàm	65
7.2.2	Tham số dạng tham biến và tham trị.....	68

7.2.3	Sử dụng biến toàn cục.....	69
7.2.4	Dùng dẫn hướng #define.....	71
7.3	Bài tập	71
BÀI 8 :	MẢNG VÀ CHUỖI.....	72
8.1	Mục tiêu	72
8.2	Nội dung	72
8.2.1	Mảng.....	72
8.2.1.1	Cách khai báo mảng	72
8.2.1.2	Tham chiếu đến từng phần tử mảng	72
8.2.1.3	Nhập dữ liệu cho mảng	73
8.2.1.4	Đọc dữ liệu từ mảng.....	73
8.2.1.5	Sử dụng biến kiểu khác.....	74
8.2.1.6	Kỹ thuật Sentinel	74
8.2.1.7	Khởi tạo mảng.....	75
8.2.1.8	Khởi tạo mảng không bao hàm kích thước.....	76
8.2.1.9	Mảng nhiều chiều.....	76
8.2.1.10	Tham chiếu đến từng phần tử mảng 2 chiều	76
8.2.1.11	Nhập dữ liệu cho mảng 2 chiều	77
8.2.1.12	Đọc dữ liệu từ mảng 2 chiều	77
8.2.1.13	Sử dụng biến kiểu khác trong mảng 2 chiều.....	78
8.2.1.14	Khởi tạo mảng 2 chiều	78
8.2.1.15	Dùng mảng 1 chiều làm tham số cho hàm	79
8.2.1.16	Dùng mảng 2 chiều làm tham số cho hàm	82
8.2.2	Chuỗi.....	84
8.2.2.1	Cách khai báo chuỗi.....	84
8.2.2.2	Hàm nhập (gets), xuất (puts) chuỗi	85
8.2.2.3	Khởi tạo chuỗi.....	86
8.2.2.4	Mảng chuỗi.....	86
8.3	Bài tập	87
BÀI 9 :	CON TRỎ.....	89
9.1	Mục tiêu	89
9.2	Nội dung	89
9.2.1	Con trỏ?	89
9.2.2	Khái báo biến con trỏ.....	89
9.2.3	Truyền địa chỉ sang hàm.....	90
9.2.4	Con trỏ và mảng.....	91
9.2.5	Con trỏ trỏ đến mảng trong hàm	91
9.2.6	Con trỏ và chuỗi	92
9.2.7	Khởi tạo mảng con trỏ trỏ đến chuỗi	93
9.2.8	Xử lý con trỏ trỏ đến chuỗi.....	94
9.2.9	Con trỏ trỏ đến con trỏ	96
9.3	Bài tập	97

BÀI 10 : CÁC KIỂU DỮ LIỆU TỰ TẠO	98
10.1 Mục tiêu	98
10.2 Nội dung	98
10.2.1 Structure.....	98
10.2.1.1 Khai báo kiểu structure	98
10.2.1.2 Cách khai báo biến có kiểu structure.....	98
10.2.1.3 Tham chiếu các phần tử trong structure	98
10.2.1.4 Khởi tạo structure.....	100
10.2.1.5 Structure lồng nhau	101
10.2.1.6 Truyền structure sang hàm	103
10.2.2 Enum.....	105
10.2.2.1 Định nghĩa kiểu enum.....	105
10.2.2.2 Cách khai báo biến có kiểu enum.....	105
10.2.2.3 Sử dụng enum trong chương trình.....	106
10.3 Bài tập.....	107
BÀI 11 : TẬP TIN.....	109
11.1 Mục tiêu	109
11.2 Nội dung	109
11.2.1 Ví dụ ghi, đọc số nguyên	109
11.2.2 Ghi, đọc mảng	110
11.2.3 Ghi, đọc structure	111
11.2.4 Các mode khác để mở tập tin	112
11.2.5 Một số hàm thao tác trên file khác	112
11.3 Bài tập.....	112
BÀI 12 : ĐỆ QUY.....	113
12.1 Mục tiêu	113
12.2 Nội dung	113
12.3 Bài tập.....	116
BÀI 13 : TRÌNH SOẠN THẢO CỦA BORLAND C	117
13.1 Mở tập tin soạn thảo mới	117
13.2 Lưu tập tin	117
13.2.1 Nếu là tập tin soạn thảo mới chưa lưu	117
13.2.2 Nếu là tập tin đã lưu ít nhất 1 lần hoặc được mở bằng lệnh Open:	117
13.3 Mở tập tin.....	118
13.4 Các phím, tổ hợp phím thường dùng.....	118
13.4.1 Các phím di chuyển con trỏ	118

13.4.2 Các phím thao tác trên khối	119
13.4.3 Các thao tác xóa	119
13.4.4 Các thao tác copy, di chuyển	119
13.4.5 Các thao tác khác	119
13.5 Ghi một khối ra đĩa.....	120
13.6 Chèn nội dung file từ đĩa vào vị trí con trỏ.....	120
13.7 Tìm kiếm văn bản trong nội dung soạn thảo.....	120
13.8 Tìm và thay thế văn bản trong nội dung soạn thảo.....	120
13.9 Sửa lỗi cú pháp.....	121
13.10 Chạy từng bước	121
13.11 Sử dụng Help (Giúp đỡ)	121
BÀI 14 : CÁC HỆ ĐẾM.....	123
14.1 Khái niệm.....	123
14.2 Quy tắc	123
14.3 Chuyển đổi giữa các hệ.....	124
14.3.1 Chuyển đổi giữa hệ 2 và hệ 10	124
14.3.2 Chuyển đổi giữa hệ 8 và hệ 10	125
14.3.3 Chuyển đổi giữa hệ 16 và hệ 10	125
14.3.4 Chuyển đổi giữa hệ 2 và hệ 16	126
BÀI 15 : BIỂU THỨC VÀ PHÉP TOÁN.....	127
15.1 Biểu thức	127
15.2 Phép toán.....	127
15.2.1 Phép toán số học.....	127
15.2.2 Phép quan hệ	127
15.2.3 Phép toán luận lý.....	128
15.2.4 Phép toán trên bit (bitwise)	128
15.2.5 Các phép toán khác.....	129
15.2.6 Độ ưu tiên của các phép toán.....	129
15.3 Bài tập	129
BÀI 16 : MỘT SỐ HÀM CHUẨN THƯỜNG DÙNG.....	131
16.1 Các hàm chuyển đổi dữ liệu	131
16.1.1 atof	131
16.1.2 atoi	131
16.1.3 itoa	131
16.1.4 tolower	131

16.1.5 toupper	131
16.2 Các hàm xử lý chuỗi ký tự.....	132
16.2.1 strcat	132
16.2.2 strcpy	132
16.2.3 strcmp	132
16.2.4 strcmpi	132
16.2.5 strlen	132
16.2.6 strdup	132
16.2.7 strlen	133
16.3 Các hàm toán học	133
16.3.1 abs	133
16.3.2 labs	133
16.3.3 rand.....	133
16.3.4 random	133
16.3.5 pow.....	133
16.3.6 sqrt.....	133
16.4 Các hàm xử lý file	134
16.4.1 rewind	134
16.4.2 ftell	134
16.4.3 fseek	134



Bài 1 :**NGÔN NGỮ LẬP TRÌNH & PHƯƠNG PHÁP LẬP TRÌNH****1.1 Mục tiêu**

Sau khi hoàn tất bài này học viên sẽ hiểu và vận dụng các kiến thức kỹ năng cơ bản sau:

- Ý nghĩa, các bước lập trình.
- Xác định dữ liệu vào, ra.
- Phân tích các bài toán đơn giản.
- Khái niệm so sánh, lặp.
- Thể hiện bài toán bằng lưu đồ.

1.2 Lý thuyết**1.2.1 Ngôn ngữ lập trình (Programming Language)**

Phần này chúng ta sẽ tìm hiểu một số khái niệm căn bản về thuật toán, chương trình, ngôn ngữ lập trình. Thuật ngữ "thuật giải" và "thuật toán" dĩ nhiên có sự khác nhau song trong nhiều trường hợp chúng có cùng nghĩa.

1.2.1.1 Thuật giải (Algorithm)

Là một dãy các thao tác xác định trên một đối tượng, sao cho sau khi thực hiện một số hữu hạn các bước thì đạt được mục tiêu. Theo R.A.Kowalski thì bản chất của thuật giải:

Thuật giải = Logic + Điều khiển

* **Logic**: Đây là phần khá quan trọng, nó trả lời câu hỏi "Thuật giải làm gì, giải quyết vấn đề gì?", những yếu tố trong bài toán có quan hệ với nhau như thế nào v.v... Ở đây bao gồm những kiến thức chuyên môn mà bạn phải biết để có thể tiến hành giải bài toán.

Ví dụ 1: Để giải một bài toán tính diện tích hình cầu, mà bạn không còn nhớ công thức tính hình cầu thì bạn không thể viết chương trình cho máy để giải bài toán này được.

* **Điều khiển**: Thành phần này trả lời câu hỏi: giải thuật phải làm như thế nào?. Chính là cách thức tiến hành áp dụng thành phần logic để giải quyết vấn đề.

1.2.1.2 Chương trình (Program)

Là một tập hợp các mô tả, các phát biểu, nằm trong một hệ thống qui ước về ý nghĩa và thứ tự thực hiện, nhằm điều khiển máy tính làm việc. Theo Niklaus Wirth thì:

Chương trình = Thuật toán + Cấu trúc dữ liệu

Các thuật toán và chương trình đều có cấu trúc dựa trên **3 cấu trúc điều khiển cơ bản**:

* **Tuần tự** (Sequential): Các bước thực hiện tuần tự một cách chính xác từ trên xuống, mỗi bước chỉ thực hiện đúng một lần.

* **Chọn lọc** (Selection): Chọn 1 trong 2 hay nhiều thao tác để thực hiện.

* **Lặp lại** (Repetition): Một hay nhiều bước được thực hiện lặp lại một số lần.

Muốn trở thành lập trình viên chuyên nghiệp bạn hãy làm đúng trình tự để có thói quen tốt và thuận lợi sau này trên nhiều mặt của một người làm máy tính. Bạn hãy làm theo các bước sau:

Tìm, xây dựng thuật giải (trên giấy) ết chương trình trên máy
 ch chương trình ạy và thử chương trình

1.2.1.3 Ngôn ngữ lập trình (Programming language)

Ngôn ngữ lập trình là hệ thống các ký hiệu tuân theo các qui ước về ngữ pháp và ngữ nghĩa, dùng để xây dựng thành các chương trình cho máy tính.

Một chương trình được viết bằng một ngôn ngữ lập trình cụ thể (ví dụ Pascal, C...) gọi là chương trình nguồn, chương trình dịch làm nhiệm vụ dịch chương trình nguồn thành chương trình thực thi được trên máy tính.

1.2.2 Các bước lập trình

Bước 1: Phân tích vấn đề và xác định các đặc điểm. (xác định I-P-O)

Bước 2: Lập ra giải pháp. (đưa ra thuật giải)

Bước 3: Cài đặt. (viết chương trình)

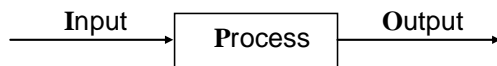
Bước 4: Chạy thử chương trình. (dịch chương trình)

Bước 5: Kiểm chứng và hoàn thiện chương trình. (thử nghiệm bằng nhiều số liệu và đánh giá)

1.2.3 Kỹ thuật lập trình

1.2.3.1 I-P-O Cycle (Input-Process-Output Cycle) (Quy trình nhập-xử lý-xuất)

Quy trình xử lý cơ bản của máy tính gồm I-P-O.



Ví dụ 2: Xác định Input, Process, Output của việc làm 1 ly nước chanh nóng

Input : ly, đường, chanh, nước nóng, muỗng.

Process : - cho hỗn hợp đường, chanh, nước nóng vào ly.
- dùng muỗng khuấy đều.

Output : ly chanh nóng đã sẵn sàng để dùng.

Ví dụ 3: Xác định Input, Process, Output của chương trình tính tiền lương công nhân tháng 10/2002 biết rằng lương = lương căn bản * ngày công

Input : lương căn bản, ngày công

Process : nhân lương căn bản với ngày công

Output : lương

Ví dụ 4: Xác định Input, Process, Output của chương trình giải phương trình bậc nhất $ax + b = 0$

Input : hệ số a, b

Process : chia -b cho a

Output : nghiệm x

Ví dụ 5: Xác định Input, Process, Output của chương trình tìm số lớn nhất của 2 số a và b.

Input : a, b

Process : Nếu $a > b$ thì *Output* = a lớn nhất
Ngược lại *Output* = b lớn nhất

Bài tập

Xác định Input, Process, Output của các chương trình sau:

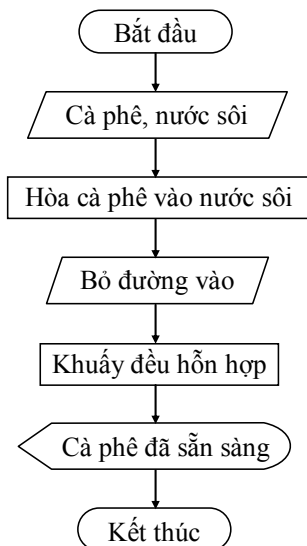
1. Đổi từ tiền VND sang tiền USD.
2. Tính điểm trung bình của học sinh gồm các môn Toán, Lý, Hóa.
3. Giải phương trình bậc 2: $ax^2 + bx + c = 0$
4. Đổi từ độ sang radian và đổi từ radian sang độ
(công thức $\text{radian} = a/180$, với a -radian, a: độ)
5. Kiểm tra 2 số a, b giống nhau hay khác nhau.

1.2.3.2 Sử dụng lưu đồ (Flowchart)

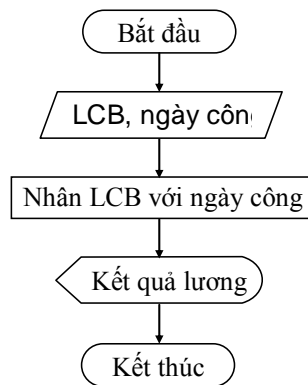
Để dễ hơn về quy trình xử lý, các nhà lập trình đưa ra dạng lưu đồ để minh họa từng bước quá trình xử lý một vấn đề (bài toán).

Hình dạng (symbol)	Hành động (Activity)
	Dữ liệu vào (Input)
	Xử lý (Process)
	Dữ liệu ra (Output)
	Quyết định (Decision), sử dụng điều kiện
	Luồng xử lý (Flow lines)
	Gọi CT con, hàm... (Procedure, Function...)
	Bắt đầu, kết thúc (Begin, End)
	Điểm ghép nối (Connector)

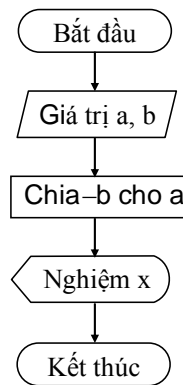
Ví dụ 6: Chuẩn bị cà phê



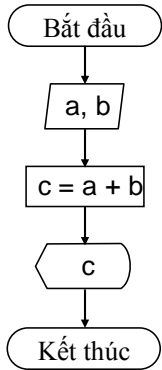
Ví dụ 7: Mô tả ví dụ 3



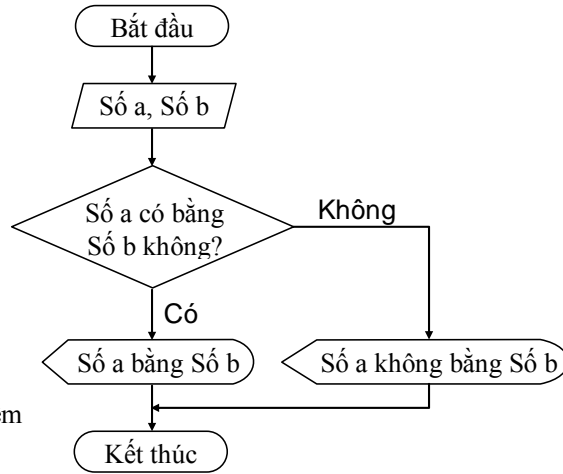
Ví dụ 8: Mô tả ví dụ 4



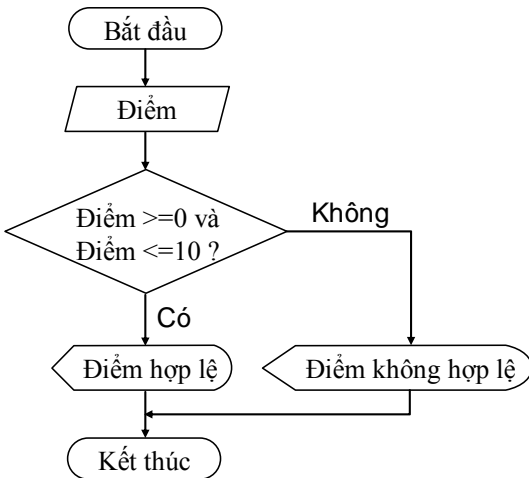
Ví dụ 9: Cộng 2 số



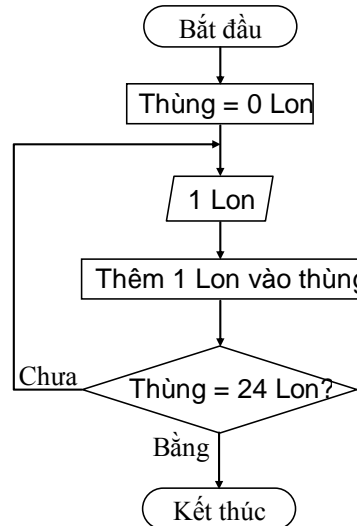
Ví dụ 10: so sánh 2 số



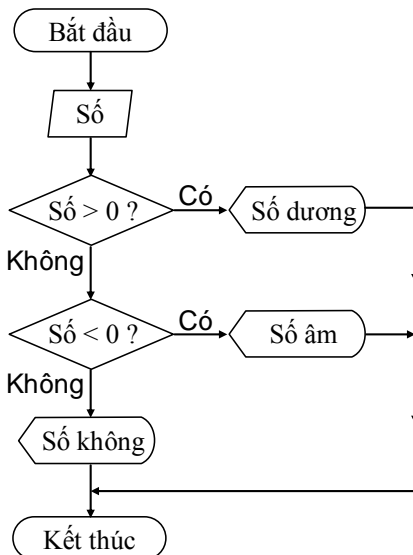
Ví dụ 11: Kiểm tra tính hợp lệ của điểm



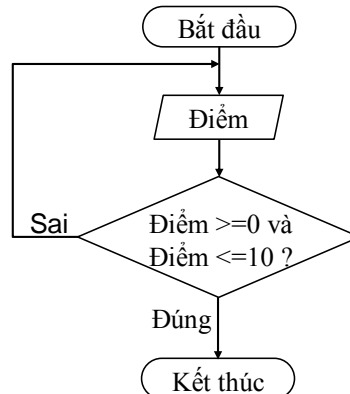
Ví dụ 12: Xếp lon vào thùng



Ví dụ 13: Kiểm tra loại số



Ví dụ 14: Kiểm tra tính hợp lệ của điểm



 **Bài tập**

Vẽ lưu đồ cho các chương trình sau:

1. Đổi từ tiền VND sang tiền USD.
2. Tính điểm trung bình của học sinh gồm các môn Toán, Lý, Hóa.
3. Giải phương trình bậc 2: $ax^2 + bx + c = 0$
4. Đổi từ độ sang radian và đổi từ radian sang độ
(công thức $\text{radian} = a/180$, với a radian, a : độ)
5. Kiểm tra 2 số a, b giống nhau hay khác nhau.



Bài 2 :**LÀM QUEN LẬP TRÌNH C QUA CÁC VÍ DỤ ĐƠN GIẢN****2.1 Mục tiêu**

Sau khi hoàn tất bài này học viên sẽ hiểu và vận dụng các kiến thức kỹ năng cơ bản sau:

- Ngôn ngữ C.
- Một số thao tác cơ bản của trình soạn thảo C.
- Cách lập trình trên C.
- Tiếp cận một số lệnh đơn giản thông qua các ví dụ.
- Nắm bắt được một số kỹ năng đơn giản.

2.2 Nội dung**2.2.1 Khởi động và thoát BorlandC****2.2.1.1 Khởi động**

■ Nhập lệnh tại dấu nhắc DOS: **gõ BC [Enter]** (nếu đường dẫn đã được cài đặt bằng lệnh **path** trong đó có chứa đường dẫn đến thư mục chứa tập tin BC.EXE). Nếu đường dẫn chưa được cài đặt ta tìm xem thư mục BORLANDC nằm ở ổ đĩa nào. Sau đó ta gõ lệnh sau:

<ổ đĩa>:\BORLANDC\BIN\BC [Enter]

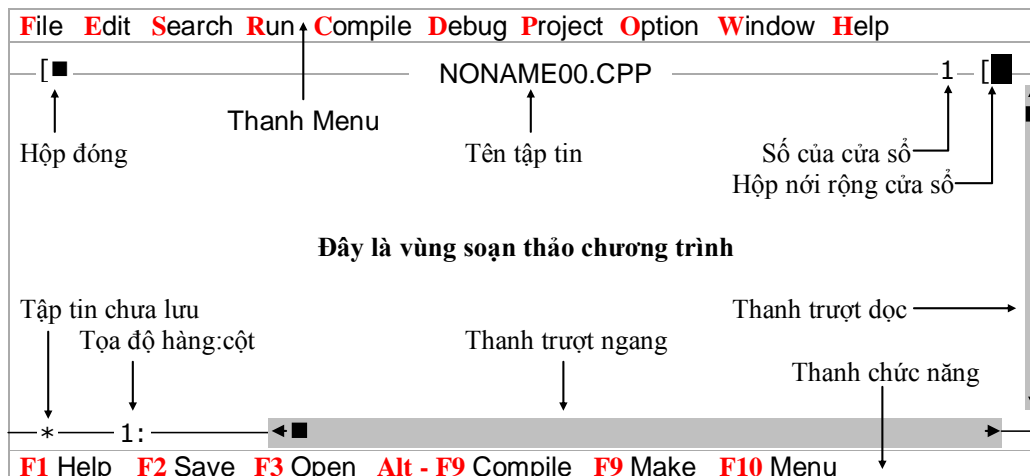
Nếu bạn muốn vừa khởi động BC vừa soạn thảo chương trình với một tập tin có tên do chúng ta đặt, thì gõ lệnh: **BC [đường dẫn]<tên file cần soạn thảo>**, nếu tên file cần soạn thảo đã có thì được nạp lên, nếu chưa có sẽ được tạo mới.

■ Khởi động tại Windows: Bạn vào menu Start, chọn Run, bạn gõ vào hộp Open 1 trong các dòng lệnh như nhập tại DOS. Hoặc bạn vào Window Explorer, chọn ổ đĩa chứa thư mục BORLANDC, vào thư mục BORLANDC, vào thư mục BIN, khởi động tập tin BC.EXE.

Ví dụ: Bạn gõ D:\BORLANDC\BIN\BC E:\BAITAP_BC\VIDU1.CPP

Câu lệnh trên có nghĩa khởi động BC và nạp tập tin VIDU1.CPP chứa trong thư mục BAITAP_BC trong ổ đĩa E. Nếu tập tin này không có sẽ được tạo mới.

Màn hình sau khi khởi động thành công




2.2.1.2 Thoát

- Ấn phím **F10** (kích hoạt Menu), chọn menu **File**, chọn **Quit**;
- Hoặc ấn tổ hợp phím **Alt - X**.

2.2.2 Các ví dụ đơn giản

2.2.2.1 Ví dụ 1

Dòng	File	Edit	Search	Run	Compile	Debug	Project	Option	Window	Help
1	/* Chương trình in ra câu bài học C đầu tiên */									
2	#include <stdio.h>									
3										
4	void main(void)									
5	{									
6	printf("Bài học C đầu tiên.");									
7	}									
	F1 Help	Alt-F8 Next Msg	Alt-F7 Prev Msg	Alt - F9 Compile	F9 Make	F10 Menu				

 **Kết quả in ra màn hình**

```
Bai hoc C dau tien. _
```

■ Dòng thứ 1: bắt đầu bằng **/*** và kết thúc bằng ***/** cho biết hàng này là hàng diễn giải (chú thích). Khi dịch và chạy chương trình, dòng này không được dịch và cũng không thi hành lệnh gì cả. Mục đích của việc ghi chú này giúp chương trình rõ ràng hơn. Sau này bạn đọc lại chương trình biết chương trình làm gì.

■ Dòng thứ 2: chứa phát biểu tiền xử lý **#include <stdio.h>**. Vì trong chương trình này ta sử dụng hàm thư viện của C là **printf**, do đó bạn cần phải có khai báo của hàm thư viện này để bảo cho trình biên dịch C biết. **Nếu không khai báo chương trình sẽ báo lỗi.**

■ Dòng thứ 3: hàng trắng viết ra với ý đồ làm cho bảng chương trình thoáng, dễ đọc.

■ Dòng thứ 4: **void main(void)** là thành phần chính của mọi chương trình C (bạn có thể viết main() hoặc void main() hoặc main(void)). Tuy nhiên, bạn nên viết theo dạng **void main(void)** để chương trình rõ ràng hơn. Mọi chương trình C đều bắt đầu thi hành từ hàm main. Cặp dấu ngoặc () cho biết đây là khối hàm (function). Hàm void main(void) có từ khóa void đầu tiên cho biết hàm này không trả về giá trị, từ khóa void trong ngoặc đơn cho biết hàm này không nhận vào đối số.

■ Dòng thứ 5 và 7: cặp dấu ngoặc móc {} giới hạn thân của hàm. Thân hàm bắt đầu bằng dấu { và kết thúc bằng dấu }.


■ Dòng thứ 6: **printf("Bài học C đầu tiên.");**, chỉ thị cho máy in ra chuỗi ký tự nằm trong nháy kép ("). Hàng này được gọi là một câu lệnh, kết thúc một câu lệnh trong C phải là dấu chấm phẩy (;).

 **Chú ý:**


- Các từ include, stdio.h, void, main, printf phải viết bằng chữ thường.
- Chuỗi trong nháy kép cần in ra "Bạn có thể viết chữ HOA, thường tùy, ý".
- Kết thúc câu lệnh phải có dấu chấm phẩy.
- Kết thúc tên hàm không có dấu chấm phẩy hoặc bất cứ dấu gì.
- Ghi chú phải đặt trong cặp /* */, hoặc //.


Formatted: Bullets and Numbering


- Thân hàm phải được bao bởi cặp { }.
- Các câu lệnh trong thân hàm phải viết thụt vào.

 Bạn nhập đoạn chương trình trên vào máy. Dịch, chạy và quan sát kết quả.

Ctrl – F9: Dịch và chạy chương trình. Alt – F5: Xem màn hình kết quả.

 Sau khi bạn nhập xong đoạn chương trình vào máy. Bạn **Ấn** và **giữ phím Ctrl**, **gõ F9** để dịch và chạy chương trình. Khi đó bạn thấy chương trình chớp rất nhanh và không thấy kết quả gì cả. **Ban Ấn** và **giữ phím Alt**, **gõ F5** để xem kết quả, khi xem xong, bạn ấn phím bất kỳ để quay về màn hình soạn thảo chương trình.

 Bây giờ bạn sửa lại dòng thứ 6 bằng câu lệnh `printf("Bai hoc C dau tien.\n");`, sau đó dịch và chạy lại chương trình, quan sát kết quả.

 **Kết quả in ra màn hình**

```
Bai hoc C dau tien.
```


Ở dòng bạn vừa sửa có thêm `\n`, `\n` là ký hiệu xuống dòng sử dụng trong lệnh `printf`. Sau đây là một số ký hiệu khác.


+ **Các kí tự điều khiển:**

- `\n` : Nhảy xuống dòng kế tiếp canh về cột đầu tiên.
- `\t` : Canh cột tab ngang.
- `\r` : Nhảy về đầu hàng, không xuống hàng.
- `\a` : Tiếng kêu bip.

+ **Các kí tự đặc biệt:**


- `\\` : In ra dấu `\`
- `\"` : In ra dấu `"`
- `'` : In ra dấu `'`

 Bây giờ bạn sửa lại dòng thứ 6 bằng câu lệnh `printf("\tBai hoc C dau tien\n");`, sau đó dịch và chạy lại chương trình, quan sát kết quả.

 **Kết quả in ra màn hình**


```
Bai hoc C dau tien.
```

Khi chạy chương trình bạn nghe tiếng bip phát ra từ loa.

 Mỗi khi chạy chương trình bạn thấy rất bất tiện trong việc xem kết quả phải ấn tổ hợp phím `Alt – F5`. Để khắc phục tình trạng này bạn sửa lại chương trình như sau:

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	<code>/* Chương trình in ra cau bai hoc C dau tien */</code>
2	<code>#include <stdio.h></code>
3	<code>#include <conio.h></code>
4	
5	<code>void main(void)</code>
6	<code>{</code>
7	<code>printf("\t\tBai hoc C\rdau tien\n");</code>
8	<code>getch();</code>


9 }
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

 **Kết quả in ra màn hình**

dau tien. Bai hoc C —


■ Dòng thứ 3: chứa phát biểu tiền xử lý `#include <conio.h>`. Vì trong chương trình này ta sử dụng hàm thư viện của C là `getch`, do đó bạn cần phải có khai báo của hàm thư viện này để báo cho trình biên dịch C biết. **Nếu không khai báo chương trình sẽ báo lỗi.**

■ Dòng thứ 8: `getch()`; chờ nhận 1 ký tự bất kỳ từ bàn phím, nhưng không in ra màn hình. Vì thế ta sử dụng hàm này để khi chạy chương trình xong sẽ dừng lại ở màn hình kết quả, sau đó ta ấn phím bất kỳ sẽ quay lại màn hình soạn thảo.

 Bạn nhập đoạn chương trình trên vào máy. Dịch, chạy và quan sát kết quả.

2.2.2.2 Ví dụ 2

Dòng	File	Edit	Search	Run	Compile	Debug	Project	Option	Window	Help
1	/* Chương trình nhập và in ra màn hình giá trị biến*/									
2	#include <stdio.h>									
3	#include <conio.h>									
4										
5	void main(void)									
6	{									
7	int i;									
8	printf("Nhập vào một số: ");									
9	scanf("%d", &i);									
10	printf("Số bạn vừa nhập là: %d", i);									
11	getch();									
12	}									
	F1 Help	Alt-F8 Next Msg	Alt-F7 Prev Msg	Alt - F9 Compile	F9 Make	F10 Menu				

 **Kết quả in ra màn hình**

Nhập vào một số: 15 Số bạn vừa nhập là: 15. —

■ Dòng thứ 7: `int i`; là lệnh khai báo, mẫu tự `i` gọi là tên biến. Biến là một vị trí trong bộ nhớ dùng lưu trữ giá trị nào đó mà chương trình sẽ lấy để sử dụng. Mỗi biến phải thuộc một kiểu dữ liệu. Trong trường hợp này ta sử dụng biến `i` kiểu số nguyên (integer) viết tắt là `int`.

■ Dòng thứ 9: `scanf("%d", &i)`. Sử dụng hàm `scanf` để nhận từ người sử dụng một trị nào đó. Hàm `scanf` trên có 2 đối mục. Đối mục `"%d"` được gọi là chuỗi định dạng, cho biết loại dữ kiện mà người sử dụng sẽ nhập vào. Chẳng hạn, ở đây phải nhập vào là số nguyên. Đối mục thứ 2

&i có dấu & đi đầu gọi là address operator, dấu & phối hợp với tên biến cho hàm scanf biến đem trị gở từ bàn phím lưu vào biến i.

■ **Dòng thứ 10:** `printf("So ban vua nhap la: %d.\n", i);`. Hàm này có 2 đối mục. Đối mục thứ nhất là một chuỗi định dạng có chứa chuỗi văn bản **So ban vua nhap la: và %d** (ký hiệu khai báo chuyển đổi dạng thức) cho biết số nguyên sẽ được in ra. Đối mục thứ 2 là i cho biết giá trị lấy từ biến i để in ra màn hình.



Bạn nhập đoạn chương trình trên vào máy. Dịch, chạy và quan sát kết quả.

2.2.2.3 Ví dụ 3

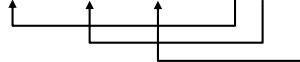
Dòng	File	Edit	Search	Run	Compile	Debug	Project	Option	Window	Help
1	/* Chuong trinh nhap vao 2 so a, b in ra tong*/									
2	#include <stdio.h>									
3	#include <conio.h>									
4										
5	void main(void)									
6	{									
7	int a, b;									
8	printf("Nhap vao so a: ");									
9	scanf("%d", &a);									
10	printf("Nhap vao so b: ");									
11	scanf("%d", &b);									
12	printf("Tong cua 2 so %d va %d la %d.", a, b, a+b);									
13	getch();									
14	}									
	F1 Help	Alt-F8 Next Msg	Alt-F7 PrevMsg	Alt - F9 Compile	F9 Make	F10 Menu				



Kết quả in ra màn hình

Nhap vao so a: 4
 Nhap vao so b: 14
 Tong cua 2 so 4 va 14 la 18.

■ **Dòng thứ 12:** `printf("Tong cua 2 so %d va %d la %d.\n", a, b, a+b);`



Bạn nhập đoạn chương trình trên vào máy. Dịch, chạy và quan sát kết quả.

2.2.2.4 Ví dụ 4

Dòng	File	Edit	Search	Run	Compile	Debug	Project	Option	Window	Help
1	/* Chuong trinh nhap vao ban kinh hinh tron. Tinh dien */									
2	#include <stdio.h>									
3	#include <conio.h>									
4										
5	#define PI 3.14									
6										

```

7 void main(void)
8 {
9     float fR;
10    printf("Nhập vào bán kính hình tròn: ");
11    scanf("%f", &fR);
12    printf("Diện tích hình tròn: %.2fn", 2*PI*fR);
13    getch();
14 }

```

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

Kết quả in ra màn hình

```

Nhập vào bán kính hình tròn: 1
Diện tích hình tròn: 6.28

```

■ **Dòng thứ 5:** `#define PI 3.14`, dùng chỉ thị `define` để định nghĩa hằng số `PI` có giá trị 3.14. Trước `define` phải có dấu `#` và cuối dòng không có dấu chấm phẩy.

■ **Dòng thứ 12:** `printf("Diện tích hình tròn: %.2f\n", 2*PI*fR);`. Hàm này có 2 đối mục. Đối mục thứ nhất là một chuỗi định dạng có chứa chuỗi văn bản **Diện tích hình tròn:** và `%.2f` (ký hiệu khai báo chuyên đổi dạng thức) cho biết dạng số chấm động sẽ được in ra, trong đó `.2` nghĩa là in ra với 2 số lẻ. Đối mục thứ 2 là biểu thức hằng `2*PI*fR`;



Bạn nhập đoạn chương trình trên vào máy. Dịch, chạy và quan sát kết quả.



Bài 3 :

CÁC THÀNH PHẦN TRONG NGÔN NGỮ C

3.1 Mục tiêu

Sau khi hoàn tất bài này học viên sẽ hiểu và vận dụng các kiến thức kỹ năng cơ bản sau:

- Khái niệm từ khóa
- Các kiểu dữ liệu
- Cách ghi chú
- Đặt tên biến
- Khai báo biến.
- Phạm vi sử dụng biến.

3.2 Nội dung

3.2.1 Từ khóa

Từ khóa là từ có ý nghĩa xác định dùng để khai báo dữ liệu, viết câu lệnh... Trong C có các từ khóa sau:

asm	const	else	for	interrupt	return	sizeof	void
break	continue	enum	goto	long	short	switch	volatile
case	default	extern	huge	near	static	typedef	while
cdecl	do	far	if	pascal	struct	union	
char	double	float	int	register	signed	unsigned	

☞ Các từ khóa phải viết bằng **chữ thường**

3.2.2 Tên

Khái niệm tên rất quan trọng trong quá trình lập trình, nó không những thể hiện rõ ý nghĩa trong chương trình mà còn dùng để xác định các đại lượng khác nhau khi thực hiện chương trình. Tên thường được đặt cho hằng, biến, mảng, con trỏ, nhân... Chiều dài tối đa của tên là 32 ký tự.

Tên biến hợp lệ là một chuỗi ký tự liên tục gồm: **Ký tự chữ, số và dấu gạch dưới**. Ký tự đầu của tên phải là **chữ hoặc dấu gạch dưới**. Khi đặt tên không được đặt trùng với các từ khóa.

Ví dụ 1 :

■ Các tên đúng: delta, a_1, Num_ODD, Case

■ Các tên sai:

3a_1	(ký tự đầu là số)
num-odd	(sử dụng dấu gạch ngang)
int	(đặt tên trùng với từ khóa)
del ta	(có khoảng trắng)
f(x)	(có dấu ngoặc tròn)

Lưu ý: Trong C, tên phân biệt chữ hoa, chữ thường

Ví dụ 2 : number khác Number

case khác Case
(case là từ khóa, do đó bạn đặt tên là Case vẫn đúng)

3.2.3 Kiểu dữ liệu

Có 4 kiểu dữ liệu cơ bản trong C là: char, int, float, double.

TT	Kiểu dữ liệu (Type)	Kích thước (Length)	Miền giá trị (Range)
1	unsigned char	1 byte	0 đến 255
2	char	1 byte	- 128 đến 127
3	enum	2 bytes	- 32,768 đến 32,767
4	unsigned int	2 bytes	0 đến 65,535
5	short int	2 bytes	- 32,768 đến 32,767
6	int	2 bytes	- 32,768 đến 32,767
7	unsigned long	4 bytes	0 đến 4,294,967,295
8	long	4 bytes	- 2,147,483,648 đến 2,147,483,647
9	float	4 bytes	$3.4 * 10^{-38}$ đến $3.4 * 10^{38}$
10	double	8 bytes	$1.7 * 10^{-308}$ đến $1.7 * 10^{308}$
11	long double	10 bytes	$3.4 * 10^{-4932}$ đến $1.1 * 10^{4932}$

3.2.4 Ghi chú

Trong khi lập trình cần phải ghi chú để giải thích các biến, hằng, thao tác xử lý giúp cho chương trình rõ ràng dễ hiểu, dễ nhớ, dễ sửa chữa và để người khác đọc vào dễ hiểu. Trong C có các ghi chú sau: // hoặc /* nội dung ghi chú */

Ví dụ 3 :

```
void main()
{
    int a, b;      //khai bao bien t kieu int
    a = 1;        //gan 1 cho a
    b = 3;        //gan 3 cho b
    /* thuật toán tìm số lớn nhất là
       nếu a lớn hơn b thì a lớn nhất
       ngược lại b lớn nhất */
    if (a > b) printf("max: %d", a);
    else printf("max: %d", b);
}
```

Khi biên dịch chương trình, C gặp cặp dấu ghi chú sẽ không dịch ra ngôn ngữ máy.

Tóm lại, đối với ghi chú dạng // dùng để ghi chú một hàng và dạng /* */ có thể ghi chú một hàng hoặc nhiều hàng.

3.2.5 Khai báo biến

3.2.5.1 Tên biến

Cách đặt tên biến như mục 2.

3.2.5.2 Khai báo biến

■ Cú pháp

Kiểu dữ liệu Danh sách tên biến;

☞ Kiểu dữ liệu: 1 trong các kiểu ở mục 3

Danh sách tên biến: gồm các tên biến có cùng kiểu dữ liệu, mỗi tên biến cách nhau dấu phẩy (,), cuối cùng là dấu chấm phẩy (;).

☞ Khi khai báo biến nên đặt tên biến theo quy tắc **Hungarian Notation**

Ví dụ 4 :

```
int ituai;          //khai báo biến ituai có kiểu int
float fTrongluong;  //khai báo biến fTrongluong có kiểu long
char ckitu1, ckitu2; //khai báo biến ckitu1, ckitu2 có kiểu char
```


Các biến khai báo trên theo quy tắc Hungarian Notation. Nghĩa là biến `ituoai` là kiểu `int`, bạn thêm chữ `i` (kí tự đầu của kiểu) vào đầu tên biến **tuoi** để trong quá trình lập trình hoặc sau này xem lại, sửa chữa... bạn dễ dàng nhận ra biến **ituoai có kiểu int** mà không cần phải di chuyển đến phần khai báo mới biết kiểu của biến này. Tương tự cho biến **fTrongluong**, bạn nhìn vào là biết ngay biến này có kiểu **float**.

3.2.5.3 Vừa khai báo vừa khởi gán

Có thể kết hợp việc khai báo với toán tử gán để biến nhận ngay giá trị cùng lúc với khai báo.

Ví dụ 5 :

- **Khai báo trước, gán giá trị sau:**

```
void main()
{
    int a, b, c;
    a = 1;
    b = 2;
    c = 5;
    ...
}
```

- **Vừa khai báo vừa gán giá trị:**

```
void main()
{
    int a = 1, b = 2, c = 5;
    ...
}
```

3.2.5.4 Phạm vi của biến

Khi lập trình, bạn phải nắm rõ phạm vi của biến. Nếu khai báo và sử dụng không đúng, không rõ ràng sẽ dẫn đến sai sót khó kiểm soát được, vì vậy bạn cần phải xác định đúng vị trí, phạm vi sử dụng biến trước khi sử dụng biến.

- Khai báo biến ngoài (biến toàn cục): Vị trí biến đặt bên ngoài tất cả các hàm, cấu trúc... Các biến này có ảnh hưởng đến toàn bộ chương trình. Chu trình sống của nó là bắt đầu chạy chương trình đến lúc kết thúc chương trình.

- Khai báo biến trong (biến cục bộ): Vị trí biến đặt bên trong hàm, cấu trúc.... Chỉ ảnh hưởng nội bộ bên trong hàm, cấu trúc đó.... Chu trình sống của nó bắt đầu từ lúc hàm, cấu trúc được gọi thực hiện đến lúc thực hiện xong.



Bài 4 : NHẬP / XUẤT DỮ LIỆU

4.1 Mục tiêu

- Sau khi hoàn tất bài này học viên sẽ hiểu và vận dụng các kiến thức kỹ năng cơ bản sau:
- Ý nghĩa, cách sử dụng hàm printf, scanf
 - Sử dụng khuôn dạng, ký tự đặc biệt, ký tự điều khiển trong printf, scanf.

4.2 Nội dung

4.2.1 Hàm printf

Kết xuất dữ liệu được định dạng.

■ **Cú pháp**

```
printf ("chuỗi định dạng", đối mục 1, đối mục 2,...);
```

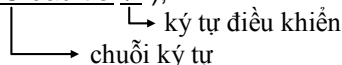
☞ *Khi sử dụng hàm phải khai báo tiền xử lý #include <stdio.h>*

- **printf**: tên hàm **phải viết bằng chữ thường**.
- **đối mục 1,...**: là các mục dữ kiện cần in ra màn hình. Các đối mục này có thể là biến, hằng hoặc biểu thức phải được định trị trước khi in ra.
- **chuỗi định dạng**: được đặt trong cặp nháy kép (" "), gồm 3 loại:
 - + Đối với **chuỗi kí tự ghi** như thế nào in ra giống như vậy.
 - + Đối với những **kí tự chuyên đổi dạng thức** cho phép kết xuất giá trị của các đối mục ra màn hình tạm gọi là mã định dạng. Sau đây là các dấu mô tả định dạng:
 - %c : Ký tự đơn
 - %s : Chuỗi
 - %d : Số nguyên thập phân có dấu
 - %f : Số chấm động (ký hiệu thập phân)
 - %e : Số chấm động (ký hiệu có số mũ)
 - %g : Số chấm động (%f hay %g)
 - %x : Số nguyên thập phân không dấu
 - %u : Số nguyên hex không dấu
 - %o : Số nguyên bát phân không dấu
 - l : Tiền tố dùng kèm với %d, %u, %x, %o để chỉ số nguyên dài (ví dụ %ld)

+ Các ký tự điều khiển và ký tự đặc biệt

- \n : Nhảy xuống dòng kế tiếp canh về cột đầu tiên.
- \t : Canh cột tab ngang.
- \r : Nhảy về đầu hàng, không xuống hàng.
- \a : Tiếng kêu bip.
- \\ : In ra dấu \
- \" : In ra dấu "
- \' : In ra dấu '
- %% : In ra dấu %

Ví dụ 1: printf("Bai hoc C dau tien");



Formatted: Bullets and Numbering

Kết quả in ra màn hình

Bai hoc C dau tien.
—

Ví dụ 2: `printf("Ma dinh dang\\\" in ra dau\" . \n");`

Kết quả in ra màn hình

Ma dinh dang" in ra dau ".
—

Ví dụ 3: giả sử biến i có giá trị = 5
`printf("So ban vua nhap la %d . \n", i);`

Kết quả in ra màn hình

So ban vua nhap la: 5.
—

Ví dụ 4: giả sử biến a có giá trị = 7 và b có giá trị = 4
`printf("Tong cua 2 so %d va %d la %d . \n", a, b, a+b);`

Kết quả in ra màn hình

Tong cua 2 so 7 va 4 la 11.
—

Ví dụ 5: sửa lại ví dụ 4
`printf("Tong cua 2 so %d va %d la %d . \n", a, b, a+b);`

☞ Kết quả in ra màn hình

Tong cua 2 so 7 va 4 la 11.
—

Ví dụ 6: sửa lại ví dụ 5

```
printf("Tong cua 2 so %5d va %3d la %1d .\n", a, b, a+b);
```

☞ Kết quả in ra màn hình

Tong cua 2 so 7 va 4 la 11.
—

☞ Dấu trừ trước bề rộng trường sẽ kéo kết quả sang trái

Ví dụ 7: sửa lại ví dụ 4

```
printf("Tong cua 2 so %2d va %02d la %04d .\n", a, b, a+b);
```

☞ Kết quả in ra màn hình

Tong cua 2 so 07 va 04 la 0011.
—

Ví dụ 8: giả sử int a = 6, b = 1234, c = 62

```
printf("%7d%7d%7d\n", a, b, c);
printf("%7d%7d%7d", 165, 2, 965);
```

☞ Kết quả in ra màn hình

6 1234 62	Số canh về bên phải bề rộng trường.
165 2 965	
—	

```
printf("%-7d%-7d%-7d\n", a, b, c);
printf("%-7d%-7d%-7d", 165, 2, 965);
```

☞ Kết quả in ra màn hình

6 1234 62	Số canh về bên trái bề rộng trường.
165 2 965	
—	

Ví dụ 9: giả sử float a = 6.4, b = 1234.56, c = 62.3

```
printf("%7.2d%7.2d%7.2d\n", a, b, c);
```

☞ **Kết quả in ra màn hình**

6.40	1234.56	62.30	Số canh về bên phải bề rộng trường.

☞ **Bề rộng trường bao gồm: phần nguyên, phần lẻ và dấu chấm động**

Ví dụ 10: giả sử float a = 6.4, b = 1234.55, c = 62.34

```
printf("%10.1d%10.1d%10.1d\n", a, b, c);
printf("%10.1d%10.1d%10.1d", 165, 2, 965);
```

☞ **Kết quả in ra màn hình**

6.4	1234.6	62.3	Số canh về bên phải bề rộng trường.
165.0	2.0	965.0	

```
printf("%-10.2d%-10.2d%-10.2d\n", a, b, c);
printf("%-10.2d%-10.2d%-10.2d", 165, 2, 965);
```

☞ **Kết quả in ra màn hình**

6.40	1234.55	62.34	Số canh về bên trái bề rộng trường.
165.00	2.00	965.00	

4.2.2 Hàm scanf

Định dạng khi nhập liệu.

■ **Cú pháp**

```
scanf ("chuỗi định dạng"[, đối mục 1, đối mục 2,...]);
```

☞ *Khi sử dụng hàm phải khai báo tiền xử lý #include <stdio.h>*

- **scanf**: tên hàm *phải viết bằng chữ thường*.
- **chuỗi định dạng**: được đặt trong cặp nháy kép (" ") là hình ảnh dạng dữ liệu nhập vào.
- **đối mục 1,...**: là danh sách các đối mục cách nhau bởi dấu phẩy, mỗi đối mục sẽ tiếp nhận giá trị nhập vào.

Ví dụ 11: `scanf("%d", &i);`

☞ Nhập vào 12abc, biến i chỉ nhận giá trị 12. Nhập 3.4 chỉ nhận giá trị 3.

Formatted: Bullets and Numbering

Ví dụ 12: `scanf("%d%d", &a, &b);`

☞ Nhập vào 2 số a, b phải cách nhau bằng **khoảng trắng** hoặc **enter**.

Ví dụ 13: `scanf("%d/%d/%d", &ngay, &thang, &nam);`

☞ Nhập vào ngày, tháng, năm theo dạng ngay/thang/nam (20/12/2002)

Ví dụ 14: `scanf("%d%*c%d%*c%d", &ngay, &thang, &nam);`

☞ Nhập vào ngày, tháng, năm với dấu phân cách /, -, ...; ngoại trừ số.

Ví dụ 15: `scanf("%2d%2d%4d", &ngay, &thang, &nam);`

☞ Nhập vào ngày, tháng, năm theo dạng dd/mm/yyyy.

4.3 Bài tập

1. *Viết chương trình đổi một số nguyên hệ 10 sang hệ 2.*
2. *Viết chương trình đổi một số nguyên hệ 10 sang hệ 16.*
3. *Viết chương trình đọc và 2 số nguyên và in ra kết quả của phép (+), phép trừ (-), phép nhân (*), phép chia (/). Nhận xét kết quả chia 2 số nguyên.*
4. *Viết chương trình nhập vào bán kính hình cầu, tính và in ra diện tích, thể tích của hình cầu đó.*
 Hướng dẫn: $S = 4r^2$ và $V = (4/3)r^3$.
5. *Viết chương trình nhập vào một số a bất kỳ và in ra giá trị bình phương (a^2), lập phương (a^3) của a và giá trị a^4 .*
6. *Viết chương trình đọc từ bàn phím 3 số nguyên biểu diễn ngày, tháng, năm và xuất ra màn hình dưới dạng "ngay/thang/nam" (chỉ lấy 2 số cuối của năm).*
7. *Viết chương trình nhập vào số giây từ 0 đến 86399, đổi số giây nhập vào thành dạng "gio:phut:giay", mỗi thành phần là một số nguyên có 2 chữ số.*
 Ví dụ: 02:11:05



Bài 5 :

CẤU TRÚC RỄ NHÁNH CÓ ĐIỀU KIỆN

(Cấu trúc chọn)

5.1 Mục tiêu

Sau khi hoàn tất bài này học viên sẽ hiểu và vận dụng các kiến thức kỹ năng cơ bản sau:

- Ý nghĩa lệnh, khối lệnh.
- Cú pháp, ý nghĩa, cách sử dụng lệnh if, lệnh switch.
- Một số bài toán sử dụng lệnh if, switch thông qua các ví dụ.
- So sánh đánh giá một số bài toán sử dụng lệnh if hoặc switch.
- Cách sử dụng các cấu trúc lồng nhau.

5.2 Nội dung

5.2.1 Lệnh và khối lệnh

5.2.1.1 Lệnh

Là một tác vụ, biểu thức, hàm, cấu trúc điều khiển...

Ví dụ 1:

```
x = x + 2;
printf("Day la mot lenh");
```

5.2.1.2 Khối lệnh

Là một dãy các câu lệnh được bọc bởi cặp dấu { }, các lệnh trong khối lệnh phải viết thụt vô 1 tab so với cặp dấu { }

Ví dụ 2:

```
{ //dau khoi
  a = 5;
  b = 6;
  printf("Tong %d + %d = %d", a, b, a+b);
} //cuoi khoi
```

} viết thụt vô 1 tab so với cặp { }

☞ Quên dùng cặp dấu { } bao bọc khi sử dụng khối lệnh, hoặc mở dấu { và quên đóng dấu }

5.2.2 Lệnh if

Câu lệnh if cho phép lựa chọn một trong hai nhánh tùy thuộc vào giá trị của biểu thức luận lý là đúng (true) hay sai (false) hoặc khác không hay bằng không.

5.2.2.1 Dạng 1 (if thiếu)

Quyết định sẽ thực hiện hay không một khối lệnh.

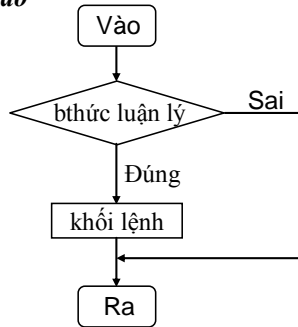
■ Cú pháp lệnh

**if (biểu thức luận lý)
khối lệnh;**

- ☞ từ khóa **if** phải viết bằng chữ thường
- ☞ kết quả của **biểu thức luận lý** phải là **đúng (1)** hoặc **sai (= 0)**

Formatted: Bullets and Numbering

Lưu đồ



☞ nếu **biểu thức luận lý** đúng thì thực hiện khối lệnh và thoát khỏi if, ngược lại không làm gì cả và thoát khỏi if.

☞ Nếu **khối lệnh** bao gồm từ 2 lệnh trở lên thì phải đặt trong dấu { }

Diễn giải:

+ Khối lệnh là một lệnh ta viết lệnh if như sau:

```
if (biểu thức luận lý)
    lệnh;
```

+ Khối lệnh bao gồm nhiều lệnh: lệnh 1, lệnh 2..., ta viết lệnh if như sau:

```
if (biểu thức luận lý)
{
    lệnh 1;
    lệnh 2;
    ...
}
```

☞ Không đặt dấu chấm phẩy sau câu lệnh if.

Ví dụ: `if(biểu thức luận lý);`

Biên dịch không báo lỗi nhưng khối lệnh không được thực hiện cho dù điều kiện đúng hay sai.

Ví dụ 3: Viết chương trình nhập vào 2 số nguyên a, b. Tìm và in ra số lớn nhất.

a. Phác họa lời giải

Trước tiên ta cho giá trị *a* là **giá trị lớn nhất bằng cách gán a cho max** (max là biến được khai báo cùng kiểu dữ liệu với a, b). Sau đó so sánh b với a, **nếu b lớn hơn a ta gán b cho max** và cuối cùng ta **được kết quả max là giá trị lớn nhất**.

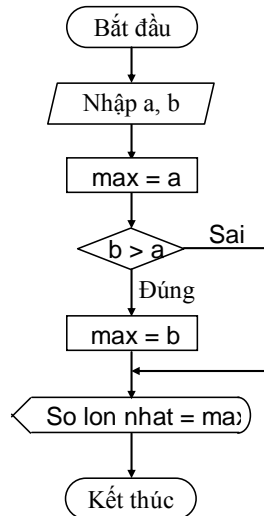
b. Mô tả quy trình xử lý (giải thuật)

Ngôn ngữ tự nhiên	Ngôn ngữ C
- Khai báo 3 biến a, b, max kiểu số nguyên	- int ia, ib, imax;
- Nhập vào giá trị a	- printf("Nhập vào số a: "); scanf("%d", &ia);
- Nhập vào giá trị b	- printf("Nhập vào số b: "); scanf("%d", &ib);
- Gán a cho max	- imax = ia;
- Nếu b > a thì gán b cho max	- if (ib > ia) imax = ib;
- In ra kết quả max	- printf("Số lớn nhất = %d\n", imax);

☞ **Biểu thức luận lý phải đặt trong cặp dấu ().** `if ib > ia` báo lỗi

Formatted: Bullets and Numbering

c. Mô tả bằng lưu đồ



d. Viết chương trình

File	Edit	Search	Run	Compile	Debug	Project	Option	Window	Help		
/* Chương trình tìm số lớn nhất từ 2 số nguyên a, b */											
#include <stdio.h>											
#include <conio.h>											
void main(void)											
{											
int ia, ib, imax;											
printf("Nhập vào số a: ");											
scanf("%d", &ia);											
printf("Nhập vào số b: ");											
scanf("%d",&ib);											
imax = ia;											
if (ib>ia)											
imax = ib;											
printf("Số lớn nhất = %d\n", imax);											
getch();											
}											
F1	Help	Alt-F8	Next Msg	Alt-F7	Prev Msg	Alt - F9	Compile	F9	Make	F10	Menu

☞ Kết quả in ra màn hình

Nhập vào số a : 10	Cho chạy lại chương trình và thử lại với: a = 7, b = 9 a = 5, b = 5 Quan sát và nhận xét kết quả
Nhập vào số b : 8	
Số lớn nhất = 10.	
—	

Ví dụ 4: Viết chương trình nhập vào 2 số nguyên a, b. Nếu a lớn hơn b thì hoán đổi giá trị a và b, ngược lại không hoán đổi. In ra giá trị a, b.

a. Phác họa lời giải

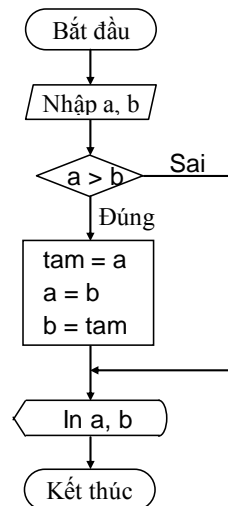
Nếu giá trị a lớn hơn giá trị b, bạn phải hoán chuyển 2 giá trị này cho nhau (nghĩa là a sẽ mang giá trị b và b mang giá trị a) bằng cách đem **giá trị a gởi (gán) cho biến tam** (biến tam

được khai báo theo kiểu dữ liệu của a, b), kể đến bạn *gán giá trị b cho a* và cuối cùng bạn *gán giá trị tam cho b*, rồi in ra a, b.

b. Mô tả quy trình thực hiện (giải thuật)

Ngôn ngữ tự nhiên	Ngôn ngữ C
- Khai báo 3 biến a, b, tam kiểu số nguyên - Nhập vào giá trị a - Nhập vào giá trị b - Nếu $a > b$ thì tam = a; a = b; b = tam; - In ra a, b	- int ia, ib, itam; - printf("Nhap vao so a: "); scanf("%d", &ia); - printf("Nhap vao so b: "); scanf("%d", &ib); - if (ia > ib) { itam = ia; ia = ib; ib = itam; } - printf("%d, %dn", ia, ib);

c. Mô tả bằng lưu đồ



d. Viết chương trình

```

File Edit Search Run Compile Debug Project Option Window Help
/* Chương trình hoán vị 2 số a, b nếu a > b */

#include <stdio.h>
#include <conio.h>

void main(void)
{
    int ia, ib, itam;
    printf("Nhap vao so a: ");
    scanf("%d", &ia);
    printf("Nhap vao so b: ");
    scanf("%d", &ib);
  
```

```

if (ia>ib)
{
    itam = ia; //hoan vi a va b
    ia = ib;
    ib = itam;
}
printf("%d, %d\n", ia, ib);
getch();
}
    
```

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

☞ **Kết quả in ra màn hình**

Nhap vao so a : 10 Nhap vao so b : 8 8, 10 _	Cho chạy lại chương trình và thử lại với: a = 1, b = 8 a = 2, b = 2 Quan sát và nhận kết quả
---	---

5.2.2.2 Dạng 2 (if đ ù)

Quyết định sẽ thực hiện 1 trong 2 khối lệnh cho trước.

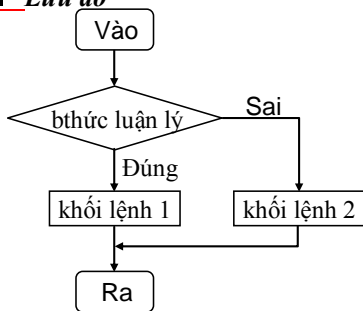
■ **Cú pháp lệnh**

```

if (biểu thức luận lý)
    khối lệnh 1;
else
    khối lệnh 2;
    
```

- ☞ từ khóa **if, else** phải viết bằng chữ thường
- ☞ kết quả của **biểu thức luận lý** phải là **đúng (1)** hoặc **sai (= 0)**

■ **Lưu đồ**



- ☞ nếu **biểu thức luận lý** đúng thì thực hiện khối lệnh 1 và thoát khỏi if ngược lại thực hiện khối lệnh 2 và thoát khỏi if.

☞ Nếu **khối lệnh 1, khối lệnh 2** bao gồm từ 2 lệnh trở lên thì phải đặt trong dấu { }

Vi dụ 5: Viết chương trình nhập vào 2 số nguyên a, b. In ra thông báo "a bằng b" nếu a = b, ngược lại in ra thông báo "a khác b".

a. Phác họa lời giải

So sánh a với b, nếu a bằng b thì in ra câu thông báo "a bằng b", ngược lại in ra thông báo "a khác b".

b. Mô tả quy trình xử lý (giải thuật)

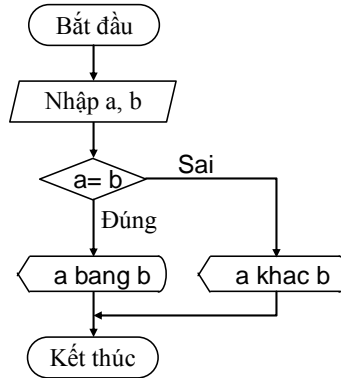
Ngôn ngữ tự nhiên	Ngôn ngữ C
- Khai báo 2 biến a, b kiểu số nguyên	- int ia, ib;
- Nhập vào giá trị a	- printf("Nhap vao so a: "); scanf("%d", &ia);
- Nhập vào giá trị b	- printf("Nhap vao so b: ");

Formatted: Bullets and Numbering

Formatted: Bullets and Numbering

<p>- Nếu $a = b$ thì in ra thông báo "a bằng b" Ngược lại (còn không thì) in ra thông báo "a khác b"</p>	<pre>scanf("%d", &ib); - if (ia == ib) printf("a bằng \n"); else printf("a khác \n");</pre>
---	---

c. Mô tả bằng lưu đồ



d. Viết chương trình

```

File Edit Search Run Compile Debug Project Option Window Help
/* Chương trình in ra thông báo "a bằng b" nếu a = b, ngược lại in ra "a khác b" */

#include <stdio.h>
#include <conio.h>

void main(void)
{
    int ia, ib;
    printf("Nhập vào số a: ");
    scanf("%d", &ia);
    printf("Nhập vào số b: ");
    scanf("%d", &ib);
    if (ia == ib)
        printf("a bằng \n");
    else
        printf("a khác \n");
    getch();
}
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu
    
```

🔗 Kết quả in ra màn hình

<p>Nhập vào số a : 10 Nhập vào số b : 8 a khác b. _</p>	<p>Cho chạy lại chương trình và thử lại với: a = 6, b = 6 a = 1, b = 5 Quan sát và nhận xét kết quả</p>
---	---

🔗 Sau else không có dấu chấm phẩy.

Ví dụ: `else; printf('a khác b\n');`

❌ `printf("a khác b\n");`; không thuộc else

Ví dụ 6: Viết chương trình nhập vào kí tự c. Kiểm tra xem nếu kí tự nhập vào là kí tự thường trong khoảng từ 'a' đến 'z' thì đổi sang chữ in hoa và in ra, ngược lại in ra thông báo "Kí tự bạn vừa nhập là: c".

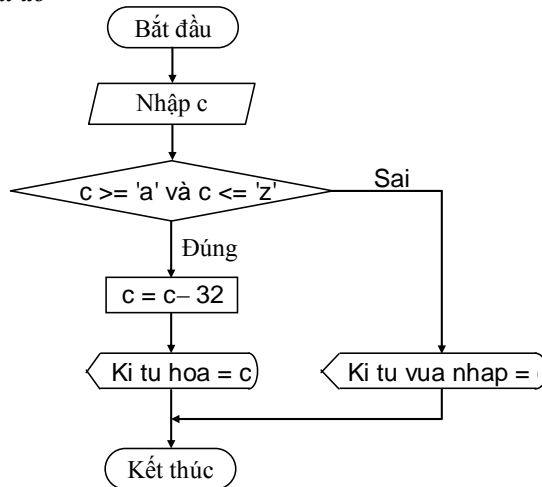
a. Phác họa lời giải

Trước tiên bạn phải kiểm tra xem nếu kí tự c thuộc khoảng 'a' và 'z' thì đổi kí tự c thành chữ in hoa bằng cách lấy kí tự c - 32 rồi gán lại cho chính nó ($c = c - 32$) (vì giữa kí tự thường và in hoa trong bảng mã ASCII cách nhau 32, ví dụ: A trong bảng mã ASCII là 65, B là 66..., còn a là 97, b là 98...), sau khi đổi xong bạn in kí tự c ra. Ngược lại, in câu thông báo "Kí tự bạn vừa nhập là: c".

b. Mô tả quy trình xử lý (giải thuật)

Ngôn ngữ tự nhiên	Ngôn ngữ C
- Khai báo biến c kiểu kí tự - Nhập vào kí tự c	- char c; - printf("Nhap vao 1 ki tu: "); scanf("%c", &c);
- Nếu $c \geq a$ và $c \leq z$ thì $c = c - 32$ in c ra màn hình	- if ($c \geq 'a' \ \&\& \ c \leq 'z'$) { $c = c - 32$; printf("Ki tu hoa la: %c\n", c); };
Ngược lại in ra thông báo Kí tự bạn vừa nhập là: c "	else printf("Ki tu ban vua nhap la: %c", c);

c. Mô tả bằng lưu đồ



d. Viết chương trình

File Edit Search Run Compile Debug Project Option Window Help
/* Chương trình nhập vào ký tự c, nếu c là chữ thường in ra chữ IN HOA */
#include <stdio.h>
#include <conio.h>
void main(void)
{
char c;

```
printf("Nhap vao 1 ki tu: ");
scanf("%c", &c);
if (c >= 'a' && c <= 'z') //hoac if(c >= 97 && c <= 122)
{
    c = c - 32; //doi thanh chu in hoa
    printf("Ki tu hoa la: %cn", c);
}
else
    printf("Ki tu ban vua nhap la: %a", c);
getch();
}
```

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

☞ **Kết quả in ra màn hình**

Nhap vao mot ki tu: g Ki tu hoa la: G _	Cho chạy lại chương trình và thử lại với: c = '!', c = '2', c = 'A', c = 'u' Quan sát và nhận xét kết quả
---	---

5.2.2.3 Cấu trúc else if

Quyết định sẽ thực hiện 1 trong n khối lệnh cho trước.

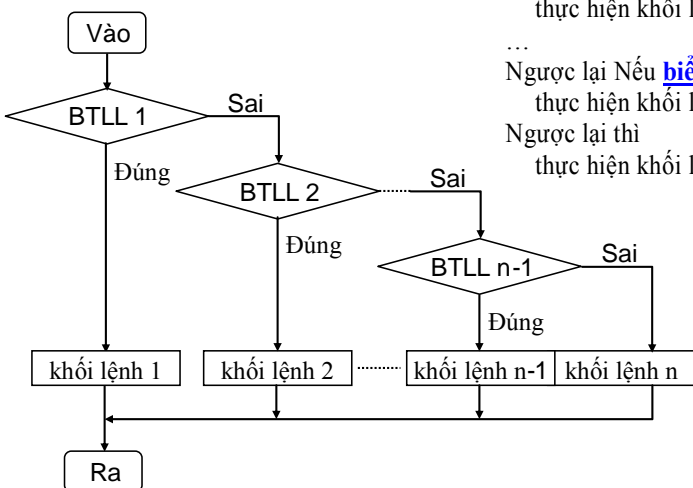
■ **Cú pháp lệnh**

```
if (biểu thức luận lý 1)
    khối lệnh 1;
else if (biểu thức luận lý 2)
    khối lệnh 2;
...
else if (biểu thức luận lý n-1)
    khối lệnh n-1;
else
    khối lệnh n;
```

- ☞ từ khóa **if, else if, else** phải viết bằng chữ thường
- ☞ kết quả của **biểu thức luận lý 1, 2..n** phải là đúng (1) hoặc sai (= 0)
- ☞ Nếu **khối lệnh 1, 2...n** bao gồm từ 2 lệnh trở lên thì phải đặt trong dấu { }

Formatted: Bullets and Numbering

■ **Lưu đồ**



Nếu **biểu thức luận lý 1** đúng thì thực hiện khối lệnh 1 và thoát khỏi cấu trúc if
 Ngược lại Nếu **biểu thức luận lý 2** đúng thì thực hiện khối lệnh 2 và thoát khỏi cấu trúc if
 ...
 Ngược lại Nếu **biểu thức luận lý n-1** đúng thì thực hiện khối lệnh n-1 và thoát khỏi cấu trúc if
 Ngược lại thì thực hiện khối lệnh n.

Formatted: Bullets and Numbering

Ví dụ 7: Viết chương trình nhập vào 2 số nguyên a, b. In ra thông báo "a lớn hơn b" nếu $a > b$, in ra thông báo "a nhỏ hơn b" nếu $a < b$, in ra thông báo "a bằng b" nếu $a = b$.

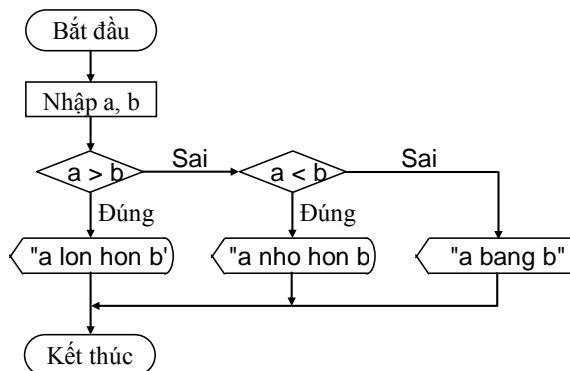
a. Phác họa lời giải

Trước tiên so sánh a với b. Nếu $a > b$ thì in ra thông báo "a lớn hơn b", ngược lại nếu $a < b$ thì in ra thông báo "a nhỏ hơn b", ngược với 2 trường hợp trên thì in ra thông báo "a bằng b".

b. Mô tả quy trình thực hiện (giải thuật)

Ngôn ngữ tự nhiên	Ngôn ngữ C
- Khai báo 2 biến a, b kiểu số nguyên - Nhập vào giá trị a - Nhập vào giá trị b - Nếu $a > b$ thì in ra thông báo "a lớn hơn b" Ngược lại Nếu $a < b$ thì in ra thông báo "a nhỏ hơn b" Ngược lại thì in ra thông báo "a bằng b"	<pre> - int ia, ib; - printf("Nhap vao so a: "); scanf("%d", &ia); - printf("Nhap vao so b: "); scanf("%d", &ib); - if (ia > ib) printf("a lon hon b"); else if (ia < ib) printf("a nho hon b"); else printf("a bang b"); </pre>

c. Mô tả bằng lưu đồ



d. Viết chương trình

```

File Edit Search Run Compile Debug Project Option Window Help
/* Chương trình nhập vào 2 số nguyên a, b. In ra thông báo a > b, a < b, a = b */
#include <stdio.h>
#include <conio.h>

void main(void)
{
    int ia, ib;
    printf("Nhap vao so a: ");
    scanf("%d", &ia);
    printf("Nhap vao so b: ");
    scanf("%d", &ib);
    if (ia > ib)
        printf("a lon hon b");
    else if (ia < ib)
        printf("a nho hon b");
    else
        printf("a bang b");
}
    
```

```

    printf("a nho hon bn");
else
    printf("a bang bn");
getch();
}

```

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

☞ Kết quả in ra màn hình

Nhập vào so a : 5 Nhập vào so b : 7 a nho hon b _	Cho chạy lại chương trình và thử lại với: a = 8, b = 4 a = 2, b = 2 Quan sát và nhận xét kết quả
--	---

Ví dụ 8: Viết chương trình nhập vào kí tự c. Kiểm tra xem nếu kí tự nhập vào là kí tự thường trong khoảng từ 'a' đến 'z' thì đổi sang chữ in hoa và in ra, nếu kí tự in hoa trong khoảng A đến Z thì đổi sang chữ thường và in ra, nếu kí tự là số từ 0 đến 9 thì in ra câu "Kí tự bạn vừa nhập là số...(in ra kí tự c)", còn lại không phải 3 trường hợp trên in ra thông báo "Bạn đã nhập kí tự...(in ra kí tự c)".

a. Phác họa lời giải

Nhập kí tự c vào, kiểm tra xem nếu kí tự c thuộc khoảng 'a' và 'z' đổi kí tự c thành chữ in hoa bằng cách lấy kí tự c - 32 rồi gán lại cho chính nó (c = c - 32) (vì giữa kí tự thường và in hoa trong bảng mã ASCII cách nhau 32, ví dụ: A trong bảng mã ASCII là 65, B là 66..., còn a là 97, b là 98...), sau khi đổi xong bạn in kí tự c ra. Ngược lại Nếu kí tự c thuộc khoảng 'A' và 'Z', đổi kí tự c thành chữ thường (theo cách ngược lại) và in ra. Ngược lại Nếu kí tự c thuộc khoảng '0' và '9' thì in ra thông báo "Kí tự bạn vừa nhập là số...". Ngược lại, in câu thông báo "Bạn đã nhập kí tự...".

b. Mô tả quy trình xử lý (giải thuật)

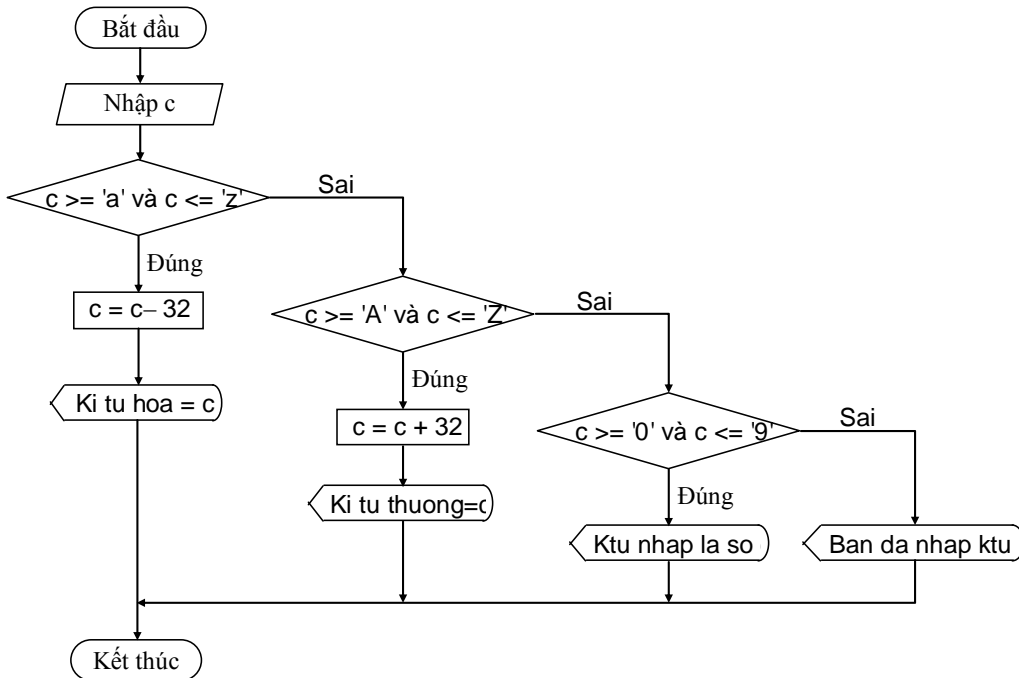
Ngôn ngữ tự nhiên	Ngôn ngữ C
- Khai báo biến c kiểu kí tự - Nhập vào kí tự c	- char c; - printf("Nhap vao 1 ki tu: "); scanf("%c", &c);
- Nếu c >= a và c <= z thì c = c - 32 in c ra màn hình	- if (c >= 'a' && c <= 'z') { c = c - 32; printf("Ki tu hoa la: %dn", c); };
Ngược lại Nếu c >= A và c <= Z thì c = c + 32 in c ra màn hình	else if(c >= 'A' && c <= 'Z') { c = c + 32; printf("Ki tu thuong la: %dn", c); };
Ngược lại Nếu c >= 0 và c <= 9 thì in thông báo "Kí tự bạn vừa nhập là số c" Ngược lại thì in thông báo "Bạn đã nhập kí tự c"	else if(c >= '0' && c <= '9') printf("Ki tu Ban vua nhap la so %d.", c); else printf("Ban da nhap ki tu %dn", c);

☞ Cũng như if, không đặt dấu chấm phẩy sau câu lệnh else if.

Ví dụ: else if(c >= 'A' && c <= 'Z');

Biên dịch không báo lỗi nhưng khối lệnh sau else if không được thực hiện.

c. Mô tả bằng lưu đồ



e. Viết chương trình

```

File Edit Search Run Compile Debug Project Option Window Help
/* Chương trình nhập và ki tu c. Đổi ra hoa, thường */

#include <stdio.h>
#include <conio.h>

void main(void)
{
    char c;
    printf("Nhập vào 1 ki tu: ");
    scanf("%c", &c);
    if (c >= 'a' && c <= 'z') //hoac if(c >= 97 && c <= 122)
    {
        c = c - 32; //doi thanh chu in hoa
        printf("Ki tu hoa la: %d\n", c);
    };
    else if(c >= 'A' && c <= 'Z') //hoac if(c >= 65 && c <= 90)
    {
        c = c + 32; //doi thanh chu thuong
        printf("Ki tu thuong la: %d\n", c);
    };
    else if(c >= '0' && c <= '9') //hoac if(c >= 48 && c <= 57)
        printf("Ki tu Ban vua nhap la so %d.", c);
}
  
```

```

else
    printf("Ban da nhap ki tu %\t", c);
    getch();
}

```

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

Kết quả in ra màn hình

Nhap vao mot ki tu: g Ki tu hoa la: G	Cho chạy lại chương trình và thử lại với: c = '!', c = '2', c = 'a', c = 'Z' Quan sát và nhận xét kết quả
--	---

Comment [n1]: Học hành vất vả nhưng kết quả chẳng ra đầu bầu vào chỗ nào cả. Về luyện công Ngày đi hót cứt trâu tối về ra chiêu thức hê hê!!!!!!

5.2.2.4 Cấu trúc if lồng

Quyết định sẽ thực hiện 1 trong n khối lệnh cho trước.

■ Cú pháp lệnh

Cú pháp là một trong 3 dạng trên, nhưng trong 1 hoặc nhiều khối lệnh bên trong phải chứa ít nhất một trong 3 dạng trên gọi là cấu trúc if lồng nhau. Thường cấu trúc if lồng nhau càng nhiều cấp độ phức tạp càng cao, chương trình chạy càng chậm và trong lúc lập trình dễ bị nhầm lẫn.

Lưu ý: Các lệnh **if...else** lồng nhau thì **else** sẽ luôn luôn kết hợp với **if** nào chưa có **else** gần nhất. Vì vậy khi gặp những lệnh if không có else, Bạn phải đặt chúng trong những **khối lệnh rõ ràng** để tránh bị hiểu sai câu lệnh.

Ví dụ 9: Bạn viết các dòng lệnh sau:

```

...
if (n > 0)
    if (a > b)
        x = a;
else
    x = b;
...

```

Mặc dù Bạn viết lệnh else thẳng hàng với if (n > 0), nhưng lệnh else ở đây được hiểu đi kèm với if (a > b), vì nó nằm gần với if (a > b) nhất và if (a > b) chưa có else. Để dễ nhìn và dễ hiểu hơn Bạn viết lại như sau:

```

...
if (n > 0)
    if (a > b)
        x = a;
    else
        x = b;
...

```

Còn nếu Bạn muốn lệnh else là của if (n > 0) thì Bạn phải đặt if (a > b) x = a trong một khối lệnh. Bạn viết lại như sau:

```

...
if (n > 0)
{
    if (a > b)
        x = a;
}
else
    x = b;
...

```

■ Lưu đồ

Tương tự 3 dạng trên. Nhưng trong mỗi khối lệnh có thể có một (nhiều) cấu trúc if ở 3 dạng trên.

Formatted: Bullets and Numbering

Formatted: Bullets and Numbering

Ví dụ 10: Viết chương trình nhập vào điểm của một học sinh. In ra xếp loại học tập của học sinh đó. (Cách xếp loại. Nếu điểm ≥ 9 , Xuất sắc. Nếu điểm từ 8 đến cận 9, Giỏi. Nếu điểm từ 7 đến cận 8, Khá. Nếu điểm từ 6 đến cận 7, TBKhá. Nếu điểm từ 5 đến cận 6, TBình. Còn lại là Yếu).

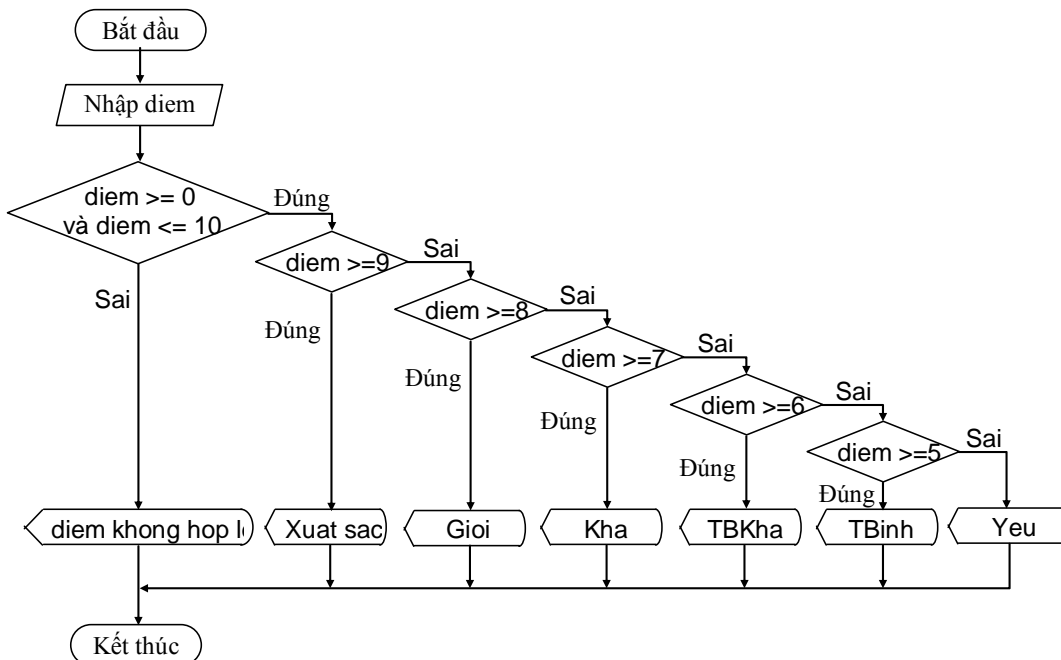
a. Phác họa lời giải

Điểm số nhập vào nếu hợp lệ ($0 \leq \text{điểm} \leq 10$), bạn tiếp tục công việc xếp loại, ngược lại thông báo "Nhập điểm không hợp lệ". Việc xếp loại bạn sử dụng cấu trúc else if.

b. Mô tả quy trình xử lý (giải thuật)

Ngôn ngữ tự nhiên	Ngôn ngữ C
- Khai báo biến điểm kiểu số thực - Nhập vào điểm số	- float fdiem; - printf("Nhap vao diem so: "); scanf("%f", &fdiem);
- Nếu điểm ≥ 0 và điểm ≤ 10 thì - Nếu điểm ≥ 9 thì in ra xếp loại = Xuất sắc Ngược lại Nếu điểm ≥ 8 thì in ra xếp loại = Giỏi Ngược lại Nếu điểm ≥ 7 thì in ra xếp loại = Khá Ngược lại Nếu điểm ≥ 6 thì in ra xếp loại = TBKhá Ngược lại Nếu điểm ≥ 5 thì in ra xếp loại = TBình Ngược lại thì in ra xếp loại = Yếu Ngược lại thì in ra "Bạn nhập điểm không hợp lệ"	- if (fdiem ≥ 0 && fdiem ≤ 10) - if (fdiem ≥ 9) printf("Xep loai = Xuat sac\n"); else if (fdiem ≥ 8) printf("Xep loai = Gioi\n"); else if (fdiem ≥ 7) printf("Xep loai = Khan\n"); else if (fdiem ≥ 6) printf("Xep loai = TBKhan\n"); else if (fdiem ≥ 5) printf("Xep loai = TBinh\n"); else printf("Xep loai = Yeu\n"); else printf("Ban nhap diem khong hop le.");

c. Mô tả bằng lưu đồ



d. Viết chương trình

File Edit Search Run Compile Debug Project Option Window Help
<pre>/* Chương trình danh gia điểm %o #include <stdio.h> #include <conio.h> void main(void) { float fdiem; printf("Nhập vào điểm số: "); scanf("%f", &fdiem); if (fdiem >=0 && fdiem <=10) if (fdiem >=9) printf("Xếp loại = Xuất sắc\n"); else if (fdiem >=8) printf("Xếp loại = Giỏi\n"); else if (fdiem >=7) printf("Xếp loại = Khá\n"); else if (fdiem >=6) printf("Xếp loại = TBKhá\n"); else if (fdiem >=5) printf("Xếp loại = TBình\n"); else printf("Xếp loại = Yếu\n"); else //if (fdiem >=0 && fdiem <=10) printf("Nhập điểm không hợp lệ\n"); getch(); }</pre>
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

Kết quả in ra màn hình

Nhập vào điểm số: 6.5 Xếp loại = TBKhá. _	Cho chạy lại chương trình và thử lại với: diem = 4, diem = 9, diem = 7, diem = 12 Quan sát và nhận xét kết quả
---	--

e. Bàn thêm về chương trình

Trong chương trình trên cấu trúc **else if** được lồng vào trong cấu trúc dạng 2, trong cấu trúc **else if** ta không cần đặt trong khối vì tất cả các **if** trong cấu trúc này đều có **else**, nên **else printf("Nhập điểm không hợp lệ!")** đương nhiên là thuộc về **if (fdiem >= 0 && fdiem <= 10)**. Giả sử trong cấu trúc **else if** không có dòng **else printf("Xếp loại = Yếu.\n")** thì khi đó dòng **else printf("Nhập điểm không hợp lệ!")** sẽ thuộc về cấu trúc **else if** chứ không thuộc về **if (fdiem >=0 && fdiem <= 10)**. Đối với trường hợp đó bạn cần phải đặt cấu trúc **else if** vào trong **{}**, thì khi đó dòng **else printf("Nhập điểm không hợp lệ.\n")** sẽ thuộc về **if (fdiem >= 0 && fdiem <= 10)**.

Ví dụ 11: Viết chương trình nhập vào 3 số nguyên a, b, c. Tìm và in ra số lớn nhất.

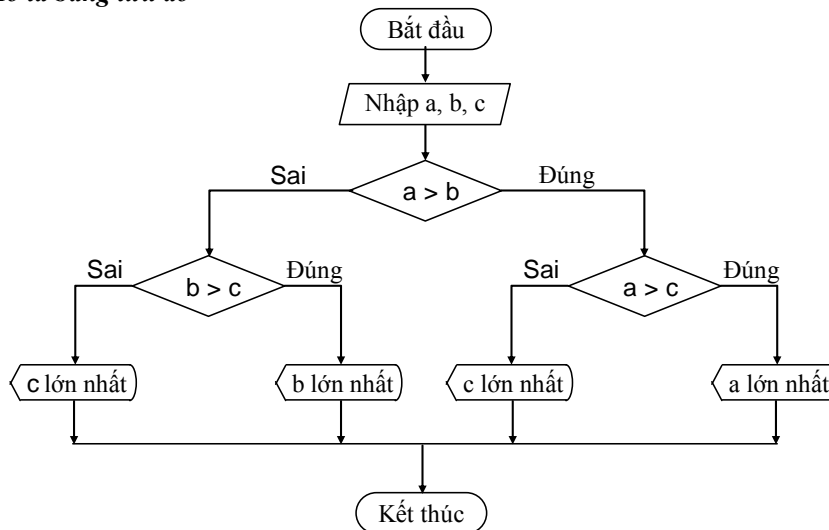
a. Phác họa lời giải

Trước tiên bạn so nếu $a > b$, mà $a > c$ thì a lớn nhất, ngược lại c lớn nhất, còn nếu $a \leq b$, mà $c > b$ thì b lớn nhất, ngược lại c lớn nhất.

b. Mô tả quy trình xử lý (giải thuật)

Ngôn ngữ tự nhiên	Ngôn ngữ C
- Khai báo 3 biến a, b, c kiểu số nguyên - Nhập vào số a - Nhập vào số b - Nhập vào số c - Nếu a > b thì - Nếu a > c thì a lớn nhất Ngược lại thì c lớn nhất Ngược lại - Nếu b > c thì b lớn nhất Ngược lại thì c lớn nhất	<pre> - int ia, ib, ic; - printf("Nhap vao so a: "); scanf("%d", &ia); - printf("Nhap vao so b: "); scanf("%d", &ib); - printf("Nhap vao so c: "); scanf("%d", &ic); - if (ia > ib) - if (ia > ic) printf("%d lon nhat\n", ia); else printf("%d lon nhat\n", ic); else - if (ib > ic) printf("%d lon nhat\n", ib); else printf("%d lon nhat\n", ic); </pre>

c. Mô tả bằng lưu đồ



d. Viết chương trình

```

File Edit Search Run Compile Debug Project Option Window Help
/* Chương trình nhập vào 2 số nguyên a, b, c. Tìm, in ra số lớn nhất */

#include <stdio.h>
#include <conio.h>

void main(void)
{
    int ia, ib, ic;
    printf("Nhap vao so a: ");

```

```
scanf("%d", &ia);
printf("Nhập vào số b: ");
scanf("%d", &ib);
printf("Nhập vào số c: ");
scanf("%d", &ic);
if (ia > ib)
    if (ia > ic)
        printf("%d lớn nhất", ia);
    else
        printf("%d lớn nhất", ic);
else
    if (ib > ic)
        printf("%d lớn nhất", ib);
    else
        printf("%d lớn nhất", ic);
getch();
}
```

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

☞ Kết quả in ra màn hình

Nhập vào số a: 4 Nhập vào số b: 5 Nhập vào số c: 3 5 lớn nhất. _	Cho chạy lại chương trình và thử lại với: a = 5, b = 4, c = 2 a = 2, b = 1, c = 10 a = 5, b = 5, c = 5 Quan sát và nhận xét kết quả
--	---

e. Bàn thêm về chương trình

Trong chương trình trên cấu trúc **dạng 2** được lồng vào trong cấu trúc **dạng 2**.

5.2.3 Lệnh switch

Lệnh switch cũng giống cấu trúc else if, nhưng nó mềm dẻo hơn và linh động hơn nhiều so với sử dụng if. Tuy nhiên, nó cũng có mặt hạn chế là kết quả của biểu thức phải là giá trị hằng nguyên (có giá trị cụ thể). Một bài toán sử dụng lệnh switch thì cũng có thể sử dụng if, nhưng ngược lại còn tùy thuộc vào giải thuật của bài toán.

5.2.3.1 Cấu trúc switch...case (switch thiếu)

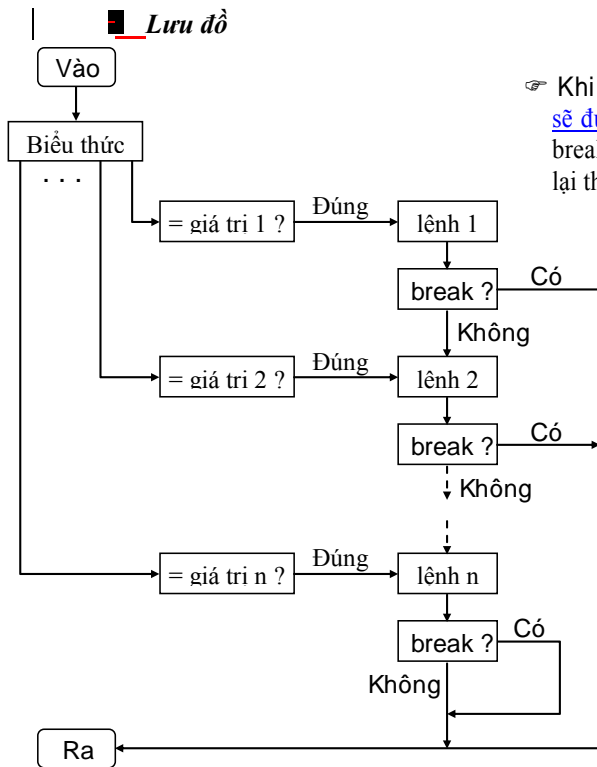
Chọn thực hiện 1 trong n lệnh cho trước.

■ Cú pháp lệnh

```
switch (biểu thức)
{
    case giá trị 1 : lệnh 1;
                  break;
    case giá trị 2 : lệnh 2;
                  break;
    ...
    case giá trị n : lệnh n;
                  [break;]
}
```

- ☞ từ khóa **switch, case, break** phải viết bằng chữ thường
- ☞ **biểu thức** phải là có kết quả là giá trị hằng nguyên (**char, int, long,...**)
- ☞ **Lệnh 1, 2...n** có thể gồm nhiều lệnh, nhưng không cần đặt trong cặp dấu { }

Formatted: Bullets and Numbering



☞ Khi giá trị của biểu thức bằng giá trị i thì lệnh i sẽ được thực hiện. Nếu sau lệnh i không có lệnh break thì sẽ tiếp tục thực hiện lệnh i + 1... Ngược lại thoát khỏi cấu trúc switch.

Ví dụ 12: Viết chương trình nhập vào số 1, 2, 3. In ra tương ứng 1, 2, 3 sao.

a. Viết chương trình

```
File Edit Search Run Compile Debug Project Option Window Help
/* Chương trình nhập vào số 1, 2, 3. In ra số sao tương ứng */
#include <stdio.h>
#include <conio.h>
void main(void)
{
    int i;
    printf("Nhập vào số 1, 2 hoặc 3: ");
    scanf("%d", &i);
    switch(i)
    {
        case 3: printf("**");
        case 2: printf("**");
        case 1: printf("**");
    };
    printf("An phím bat ky de ket thuat");
    getch();
}
```

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

☞ **Kết quả in ra màn hình**

Nhập vào số 1, 2 hoặc 3: 2 ** _	Cho chạy lại chương trình và thử lại với: i = 1, i = 3, i = 0, i = 4 Quan sát và nhận xét kết quả
---------------------------------------	---

b. Bàn thêm về chương trình

Trong chương trình trên khi nhập vào $i = 2$ lệnh `printf("**")` ở dòng case 2 được thi hành, nhưng do không có lệnh `break` sau đó nên lệnh `printf("**")` ở dòng case 1 tiếp tục được thi hành. Kết quả in ra **.

☞ **Không đặt dấu chấm phẩy sau câu lệnh switch.**

Ví dụ: `switch(i);`

Biên dịch không báo lỗi nhưng các lệnh trong switch không được thực hiện.

Ví dụ 13: Viết chương trình nhập vào tháng và in ra quý. (tháng 1 -> quý 1, tháng 10 -> quý 4)

a. Phác họa lời giải

Nhập vào giá trị tháng, kiểm tra xem tháng có hợp lệ (trong khoảng 1 đến 12). Nếu hợp lệ in ra quý tương ứng (1->3: quý 1, 4>6: quý 2, 7>9: quý 3, 10>12: quý 4).

b. Viết chương trình

```
File Edit Search Run Compile Debug Project Option Window Help
/* Chương trình nhập vào tháng. In ra quý tương ứng */
#include <stdio.h>
#include <conio.h>
void main(void)
{
    int ithang;
    printf("Nhập vào tháng: ");
    scanf("%d", &ithang);
    if (ithang > 0 && ithang <= 12)
        switch(ithang)
        {
            case 1:
            case 2:
            case 3: printf("Quý 1.\n");
                    break;
            case 4:
            case 5:
            case 6: printf("Quý 2.\n");
                    break;
            case 7:
            case 8:
            case 9: printf("Quý 3.\n");
                    break;
            case 10:
            case 11:
            case 12: printf("Quý 4.\n");
                    break;
        };
    else
        printf("Thang khong hop le\n");
    getch();
}
```

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

☞ **Kết quả in ra màn hình**

Nhập vào tháng: 4 Quý 2. _	Cho chạy lại chương trình và thử lại với: tháng = 7, tháng = 1, tháng = 13, tháng = Quan sát và nhận xét kết quả
----------------------------------	--

c. **Bàn thêm về chương trình**

Trong chương trình trên cấu trúc **switch...case** được lồng vào trong cấu trúc **if** dạng 2.

5.2.3.2 Cấu trúc **switch...case...default** (switch đủ)

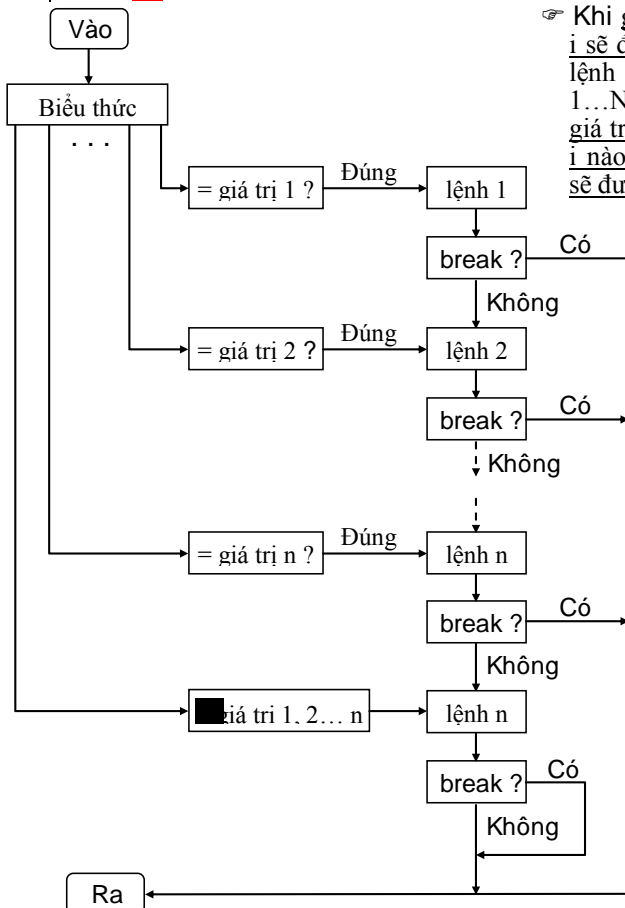
Chọn thực hiện 1 trong $n + 1$ lệnh cho trước.

■ **Cú pháp lệnh**

```
switch (biểu thức)
{
    case giá trị 1 : lệnh 1;
                  break;
    case giá trị 2 : lệnh 2;
                  break;
    ...
    case giá trị n : lệnh n;
                  break;
    default      : lệnh;
                  [break;]
}
```

- ☞ từ khóa **switch, case, break, default** phải viết bằng chữ thường
- ☞ **biểu thức** phải là có kết quả là giá trị nguyên (char, int, long,...)
- ☞ **Lệnh 1, 2...n** có thể gồm nhiều lệnh, nhưng không cần đặt trong cặp dấu { }

■ **Lưu đồ**



- ☞ Khi giá trị của biểu thức bằng giá trị i thì lệnh i sẽ được thực hiện. Nếu sau lệnh i không có lệnh **break** thì sẽ tiếp tục thực hiện lệnh $i + 1$... Ngược lại thoát khỏi cấu trúc **switch**. Nếu giá trị biểu thức không trùng với bất kỳ giá trị i nào thì lệnh tương ứng với từ khóa **default** sẽ được thực hiện.


Formatted: Bullets and Numbering

Formatted: Bullets and Numbering

Ví dụ 14: Viết lại chương trình ở **Ví dụ 12**

a. Viết chương trình

File	Edit	Search	Run	Compile	Debug	Project	Option	Window	Help
<pre> /* Chương trình nhập vào số 1, 2, 3. In ra số sao tương ứng */ #include <stdio.h> #include <conio.h> void main(void) { int i; printf("Nhập vào số 1, 2 hoặc 3: "); scanf("%d", &i); switch(i) { case 3: printf("**"); case 2: printf("**"); case 1: printf("**"); break; default: printf("Bạn phải nhập vào số 1, 2 hoặc 3."); }; getch(); } </pre>									
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu									

 **Kết quả in ra màn hình**

Nhập vào số 1, 2 hoặc 3: 3 *** _	Cho chạy lại chương trình và thử lại với: i = 1, i = 3, i = 0, i = 4 Quan sát kết quả
--	---

b. Bàn thêm về chương trình

Trong chương trình trên. Nếu bạn nhập vào 1, 2, 3 sẽ in ra số sao tương ứng. Ngoài các số này chương trình sẽ in ra câu thông báo "Bạn phải nhập vào số 1, 2 hoặc 3".

Ví dụ 15: Viết lại chương trình ở **Ví dụ 13**

a. Viết chương trình

File	Edit	Search	Run	Compile	Debug	Project	Option	Window	Help
<pre> /* Chương trình nhập vào tháng. In ra quy tương ứng */ #include <stdio.h> #include <conio.h> void main(void) { int ithang; printf("Nhập vào tháng: "); scanf("%d", &ithang); switch(ithang) { </pre>									

```

case 1: case 2: case 3 printf("Quy 1.\n");
                    break;
case 4: case 5: case 6: printf("Quy 2.\n");
                    break;
case 7: case 8: case 9: printf("Quy 3.\n");
                    break;
case 10: case 11: case 12: printf("Quy 4.\n");
                    break;
default                : printf("Ban phai nhap vao so trong khoang 1..12\n");
};
getch();
}

```

F1 Help **Alt-F8** Next Msg **Alt-F7** Prev Msg **Alt - F9** Compile **F9** Make **F10** Menu

☞ Kết quả in ra màn hình

Nhap vao thang: 4
Quy 2.

Cho chạy lại chương trình và thử lại với:
thang = 7, thang = 1, thang = 13, thang =
Quan sát kết quả

c. Bàn thêm về chương trình

Trong chương trình trên. Nếu bạn nhập vào 1 đến 12 sẽ in quý tương ứng. Ngoài các số này chương trình sẽ in ra câu thông báo "Bạn phải nhập vào số trong khoảng 1..12".

5.2.3.3 Cấu trúc switch lồng

Quyết định sẽ thực hiện 1 trong n khối lệnh cho trước.

■ Cú pháp lệnh

Cú pháp là một trong 2 dạng trên, nhưng trong 1 hoặc nhiều lệnh bên trong phải chứa ít nhất một trong 2 dạng trên gọi là cấu trúc switch lồng nhau. Thường cấu trúc switch lồng nhau càng nhiều cấp độ phức tạp càng cao, chương trình chạy càng chậm và trong lúc lập trình dễ bị nhầm lẫn.

■ Lưu ý

Tương tự 2 dạng trên. Nhưng trong mỗi lệnh có thể có một (nhiều) cấu trúc switch ở 2 dạng trên.

Ví dụ 16: Viết chương trình menu 2 cấp

a. Viết chương trình

File Edit Search Run Compile Debug Project Option Window Help

```

/* Chương trình menu 2 cấp */

#include <stdio.h>
#include <conio.h>

void main(void)
{
    int imenu, isubmenu;
    printf("-----\n");
    printf("  MAIN MENU  \n");
    printf("-----\n");
    printf("1. File\n");
}

```

Formatted: Bullets and Numbering

Formatted: Bullets and Numbering

```

printf("2. Edit\n");
printf("3. Search\n");
printf("Chon muc tuong ung: ");
scanf("%d", &iMENU);
switch(iMENU)
{
  case 1: printf("-----\n");
          printf("  MENU FILE  \n");
          printf("-----\n");
          printf("1. New\n");
          printf("2. Open\n");
          printf("Chon muc tuong ung: ");
          scanf("%d", &iSUBMENU);
          switch(iSUBMENU)
          {
            case 1: printf("Ban da chon chuc nang New File);
                    break;
            case 2: printf("Ban da chon chuc nang Open File);
                    }
          break; //break cua case-1
          case 2: printf("Ban da chon chuc nang Edit);
                    break;
          case 3: printf("Ban da chon chuc nang Search);
          };
  getch();
}

```

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

Kết quả in ra màn hình

<pre> ----- MAIN MENU ----- 1. File 2. Edit 3. Search Chon muc tuong ung: 1 ----- MENU FILE ----- 1. New 2. Open Chon muc tuong ung: 2 Ban da chon chuc nang Open File _ </pre>	<p>Cho chạy lại chương trình và thử lại với: mục chọn chức năng khác Quan sát kết quả. * Thêm các thành phần sau vào chương trình: - Thêm mục Save vào menu File. - Tạo menu Edit gồm 4 chức năng: Copy, Cut, Paste, Clear. - Tạo menu Search gồm 2 chức năng: Find, Replace. Chạy lại chương trình và thử với nhiều mục chọn khác nhau. Quan sát kết quả.</p>
---	--

5.3 Bài tập

5.3.1 Sử dụng lệnh if

1. *Viết lại chương trình ví dụ 3, sử dụng cấu trúc if dạng 2.*
2. *Viết lại chương trình ví dụ 11, sử dụng cấu trúc if dạng 1.*

3. **Viết lại chương trình ví dụ 11, sử dụng cấu trúc if dạng 2.**

4. **Viết chương trình nhập vào số nguyên dương, in ra thông báo số chẵn hay lẻ.**

Hướng dẫn: Nhập vào số nguyên dương x. Kiểm tra nếu x chia chẵn cho hai thì x là số chẵn (hoặc chia cho 2 dư 0) ngược lại là số lẻ.

5. **Viết chương trình nhập vào 4 số nguyên. Tìm và in ra số lớn nhất.**

Hướng dẫn: Ta có 4 số nguyên a, b, c, d. Tìm 2 số nguyên lớn nhất x, y của 2 cặp (a, b) và (c, d). Sau đó so sánh 2 số nguyên x, y để tìm ra số nguyên lớn nhất.

6. **Viết chương trình giải phương trình bậc 2: $ax^2 + bx + c = 0$, với a, b, c nhập vào từ bàn phím.**

Hướng dẫn: Nhập vào 3 biến a, b, c.

Tính Delta = $b^2 - 4*a*c$

Nếu Delta < 0 thì

Phương trình vô nghiệm

Ngược lại

Nếu Delta = 0 thì

$x_1 = x_2 = -b/(2*a)$

Ngược lại

$x_1 = (-b - \sqrt{\text{Delta}})/(2*a)$

$x_2 = (-b + \sqrt{\text{Delta}})/(2*a)$

Hết Nếu

Hết Nếu

7. **Viết chương trình nhập vào giờ phút giây (hh:mm:ss). Cộng thêm số giây nhập vào và in ra kết quả dưới dạng hh:mm:ss.**

Hướng dẫn: Nhập vào giờ phút giây vào 3 biến gio, phut, giay và nhập và giây cộng thêm vào biến them:

Nếu giay + them < 60 thì

giay = giay + them

Ngược lại

giay = (giay + them) / 60

phut = phut + 1

Nếu phut >= 60 thì

phut = phut / 60

gio = gio + 1

Hết nếu

Hết nếu

5.3.2 Sử dụng lệnh switch

8. **Viết chương trình nhập vào tháng, in ra tháng đó có bao nhiêu ngày.**

Hướng dẫn: Nhập vào tháng

Nếu là tháng 1, 3, 5, 7, 8, 10, 12 thì có 30 ngày

Nếu là tháng 4, 6, 9, 11 thì có 31 ngày

Nếu là tháng 2 và là năm nhuận thì có 29 ngày ngược lại 28 ngày

(Năm nhuận là năm chia chẵn cho 4)

9. **Viết chương trình trò chơi One-Two-Three ra cái gì ra cái này theo điều kiện:**

- Búa (B) thắng Kéo, thua Giấy.

- Kéo (K) thắng Giấy, thua Búa.

- Giấy (G) thắng Búa, thua Kéo.

Hướng dẫn: Dùng lệnh switch lồng nhau

10. Viết chương trình xác định biến ký tự color rồi in ra thông báo

- RED, nếu color = 'R' hoặc color = 'r'
- GREEN, nếu color = 'G' hoặc color = 'g'
- BLUE, nếu color = 'B' hoặc color = 'b'
- BLACK, nếu color có giá trị khác.

11. Viết chương trình nhập vào 2 số x, y và 1 trong 4 toán tử +, -, *, /. Nếu là + thì in ra kết quả x + y, nếu là - thì in ra x - y, nếu là * thì in ra x * y, nếu là / thì in ra x / y (nếu y = 0 thì thông báo không chia được)

5.4 Bài tập làm thêm**12. Viết lại bài tập 8, 9, 10, 11 sử dụng lệnh if.**

13. Viết chương trình nhập vào điểm 3 môn thi: Toán, Lý, Hóa của học sinh. Nếu tổng điểm ≥ 15 và không có môn nào dưới 4 thì in kết quả đậu. Nếu đậu mà các môn đều lớn hơn 5 thì in ra lời phê "Học đều các môn", ngược lại in ra "Học chưa đều các môn", các trường hợp khác là "Thi hỏng".

14. Viết chương trình nhập vào ngày tháng năm (dd:mm:yy), cho biết đó là thứ mấy trong tuần.

15. Viết chương trình nhập số giờ làm và lương giờ rồi tính số tiền lương tổng cộng. Nếu số giờ làm lớn hơn 40 thì những giờ làm dôi ra được tính 1,5 lần.

16. Viết chương trình nhập vào 3 giá trị nguyên dương a, b, c. Kiểm tra xem a, b, c có phải là 3 cạnh của tam giác không? Nếu là 3 cạnh của tam giác thì tính diện tích của tam giác theo công thức sau:

$$17. S = \sqrt{p * (p - a) * (p - b) * p}, \text{ với } p \text{ là } 1/2 \text{ chu vi của tam giác.}$$

Hướng dẫn: a, b, c là 3 cạnh của tam giác phải thỏa điều kiện sau:

$$(a + b) > c \text{ và } (a + c) > b \text{ và } (b + c) > a$$

18. Viết chương trình nhập vào 3 số nguyên rồi in ra màn hình theo thứ tự tăng dần.

19. Viết chương trình tính tiền điện gồm các khoảng sau:

- Tiền thuê bao điện kè: 1000đ/tháng
- Định mức sử dụng điện cho mỗi hộ là: 50 KW với giá 230đ/KW
- Nếu phần vượt định mức ≤ 50 KW thì tính giá 480đ/KW
- Nếu 50KW < phần vượt định mức < 100KW thì tính giá 700đ/KW
- Nếu phần vượt định mức ≤ 100 KW thì tính giá 900đ/KW

Chỉ số mới và cũ được nhập vào từ bàn phím

- In ra màn hình chỉ số cũ, chỉ số mới, tiền trả định mức, tiền trả vượt định mức, tổng tiền phải trả.

Bài 6 : CẤU TRÚC VÒNG LẶP

6.1 Mục tiêu

Sau khi hoàn tất bài này học viên sẽ hiểu và vận dụng các kiến thức kỹ năng cơ bản sau:

- Ý nghĩa, cách hoạt động của vòng lặp.
- Cú pháp, ý nghĩa, cách sử dụng lệnh for, while, do...while.
- Ý nghĩa và cách sử dụng lệnh break, continue.
- Một số bài toán sử dụng lệnh for, while, do...while thông qua các ví dụ.
- So sánh, đánh giá một số bài toán sử dụng lệnh for, while hoặc do...while.
- Cấu trúc vòng lặp lồng nhau.

6.2 Nội dung

6.2.1 Lệnh for

Vòng lặp xác định thực hiện lặp lại một số lần xác định của một (chuỗi hành động)

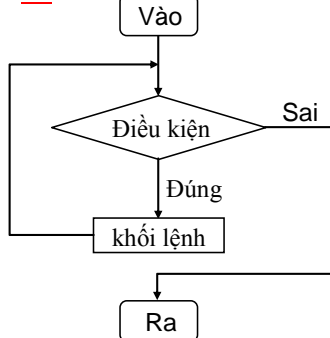
■ Cú pháp lệnh

**for (biểu thức 1; biểu thức 2; biểu thức 3)
khối lệnh;**

☞ từ khóa **for** phải viết bằng chữ thường

☞ Nếu **khối lệnh** bao gồm từ 2 lệnh trở lên thì phải đặt trong dấu { }

■ Lưu ý



☞ kiểm tra **điều kiện**
nếu **đúng** đúng thì
thực hiện khối lệnh;
lặp lại kiểm tra điều kiện
nếu **sai**
thoát khỏi vòng lặp.

Giải thích:

- + Biểu thức 1: khởi tạo giá trị ban đầu cho biến điều khiển.
- + Biểu thức 2: là quan hệ logic thể hiện điều kiện tiếp tục vòng lặp.
- + Biểu thức 3: phép gán dùng thay đổi giá trị biến điều khiển.

Nhận xét:

- + Biểu thức 1 bao giờ cũng chỉ được tính toán một lần khi gọi thực hiện for.
- + Biểu thức 2, 3 và thân for có thể thực hiện lặp lại nhiều lần.

Lưu ý:

+ **Biểu thức 1, 2, 3 phải phân cách bằng dấu chấm phẩy (;)**

+ Nếu biểu thức 2 không có, vòng for được xem là luôn luôn **đúng**. Muốn thoát khỏi vòng lặp for phải dùng một trong 3 lệnh **break**, **goto** hoặc **return**.

+ Với mỗi biểu thức có thể viết thành một dãy biểu thức con phân cách nhau bởi dấu phẩy. Khi đó các biểu thức con được xác định từ trái sang phải. Tính đúng sai của dãy biểu thức con trong biểu thức thứ 2 được xác định bởi biểu thức con cuối cùng.

+ Trong thân for (khối lệnh) có thể chứa một hoặc nhiều cấu trúc điều khiển khác.

+ Khi gặp lệnh **break**, cấu trúc lặp sâu nhất sẽ thoát ra.

+ Trong thân for có thể dùng lệnh **goto** để thoát khỏi vòng lặp đến vị trí mong muốn.

+ Trong thân for có thể sử dụng **return** để trở về một hàm nào đó.

+ Trong thân for có thể sử dụng lệnh continue để chuyển đến đầu vòng lặp (bỏ qua các câu lệnh còn lại trong thân).

Ví dụ 1: Viết chương trình in ra câu "Vi dụ sử dụng vòng lặp for" 3 lần.

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	/* Chương trình in ra câu "Vi dụ sử dụng vòng lặp for" 3 lần */
2	
3	#include <stdio.h>
4	#include <conio.h>
5	
6	#define MSG "Vi dụ sử dụng vòng lặp for".
7	
8	void main(void)
9	{
10	int i;
11	for(i = 1; i<=3; i++) /hoac for(i = 1; i<=3; i+=1)
12	printf("%s", MSG);
13	getch();
14	}
	F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

☞ **Kết quả in ra màn hình**

Vi dụ sử dụng vòng lặp for.	Bạn thay 2 dòng 11 và 12 bằng câu lệnh
Vi dụ sử dụng vòng lặp for.	for(i=1; i<=3; i++, printf("%s", MSG));
Vi dụ sử dụng vòng lặp for.	Chạy lại chương trình, quan sát và nhận xét kết quả.
—	

☞ Có dấu chấm phẩy sau lệnh for(i=1; i<=3; i++); **thì** các lệnh thuộc vòng lặp for sẽ không được thực hiện.

Ví dụ 2: Viết chương trình nhập vào 3 số nguyên. Tính và in ra tổng của chúng.

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	/* Chương trình nhập vào 3 số và tính tổng */
2	
3	#include <stdio.h>
4	#include <conio.h>
5	
6	void main(void)
7	{
8	int i, in, is;
9	is = 0;
10	for(i = 1; i<=3; i++)
11	{

12	printf("Nhap vao so thu %d :", i);
13	scanf("%d", &in);
14	is = is + in;
15	}
16	printf("Tong: %d", is);
17	getch();
18	}
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu	

☞ **Kết quả in ra màn hình**

Nhap vao so thu 1: 5	Bạn thay các dòng từ 9 đến 15 bằng câu lệnh:
Nhap vao so thu 2: 4	for(is=0, i=1; i<=3; printf("Nhap vao so thu %d: ", i), scanf("%d",
Nhap vao so thu 3: 2	&in), i++, is=is+in);
Tong: 11.	Chạy lại chương trình, quan sát và nhận xét kết quả.
_	

☞ Trong vòng lặp for có sử dụng từ 2 lệnh trở lên, nhớ sử dụng cặp ngoặc { } để bọc các lệnh đó lại. Dòng 12, 13, 14 thuộc vòng for dòng 10 do được bọc bởi cặp ngoặc { }. Nếu 3 dòng này không bọc bởi cặp ngoặc { }, thì chỉ dòng 12 thuộc vòng lặp for, còn 2 dòng còn lại không thuộc vòng lặp for.

Ví dụ 3: Viết chương trình nhập vào số nguyên n. Tính tổng các giá trị lẻ từ 0 đến n.

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	/* Chương trình nhập vào 3 số và tính tổng */
2	
3	#include <stdio.h>
4	#include <conio.h>
5	
6	void main(void)
7	{
8	int i, in, is = 0;
9	printf("Nhap vao so n: ");
10	scanf("%d", &in);
11	is = 0;
12	for(i = 0; i<=in; i++)
13	{
14	if (i % 2 != 0) //neu i la so le
15	is = is + i; //hoac is += i;
16	}
17	printf("Tong: %d", is);
18	getch();
19	}
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu	

☞ **Kết quả in ra màn hình**

Nhap vao so n : 5	Bạn thay các dòng từ 11 đến 16 bằng câu lệnh:
Tong: 9.	for(is=0, i=1; i<=n; is=is+i, i+=2);
_	Chạy lại chương trình, quan sát và nhận xét kết quả.

☞ Bạn có thể viết gộp các lệnh trong thân for vào trong lệnh for. Tuy nhiên, khi lập trình bạn nên viết lệnh for có đủ 3 biểu thức đơn và các lệnh thực hiện trong thân for mỗi lệnh một dòng để sau này có thể đọc lại dễ hiểu, dễ sửa chữa.

Ví dụ 4: Một vài ví dụ thay đổi biến điều khiển vòng lặp.

- Thay đổi biến điều khiển từ 1 đến 100, mỗi lần tăng 1:
`for(i = 1; i <= 100; i++)`
- Thay đổi biến điều khiển từ 100 đến 1, mỗi lần giảm 1:
`for(i = 100; i >= 1; i-)`
- Thay đổi biến điều khiển từ 7 đến 77, mỗi lần tăng 7:
`for(i = 7; i <= 77; i += 7)`
- Thay đổi biến điều khiển từ 20 đến 2, mỗi lần giảm 2:
`for(i = 20; i >= 2; i -= 2)`

Ví dụ 5: Đọc vào một loạt kí tự trên bàn phím. Kết thúc khi gặp dấu chấm '!'.

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	<code>/* Doc vao 1 loat ktu tren ban phim. Ket thuc khi gap dau cham */</code>
2	
3	<code>#include <stdio.h></code>
4	
5	<code>#define DAU_CHAM '!'</code>
6	
7	<code>void main(void)</code>
8	<code>{</code>
9	<code>char c;</code>
10	<code>for((c = getchar()) != DAU_CHAM;)</code>
11	<code>putchar(c);</code>
12	<code>}</code>
	F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

Kết quả in ra màn hình

a	Bạn thay các dòng từ 10 đến 11 bằng câu lệnh: for((c = getchar()) != DAU_CHAM; putchar(c)); Chạy lại chương trình, quan sát và nhận xét kết quả.
a	
4	
4	
.	
_	

Vòng lặp for vắng mặt biểu thức 1 và 3.

Ví dụ 6: Đọc vào một loạt kí tự trên bàn phím, đếm số kí tự nhập vào. Kết thúc khi gặp dấu chấm '!'.

Dòng	File Edit Search Run Cmpile Debug Project Option Window Help
1	<code>/* Doc vao 1 loat ktu tren ban phim, dem so ktu nhap vao. Ket thuc khi gap dau cham */</code>
2	
3	<code>#include <stdio.h></code>
4	<code>#include <conio.h></code>
5	
6	<code>#define DAU_CHAM '!'</code>
7	
8	<code>void main(void)</code>
9	<code>{</code>
10	<code>char c;</code>
11	<code>int idem;</code>
12	<code>for(idem = 0; (c = getchar()) != DAU_CHAM;)</code>
13	<code>idem++;</code>

14	printf("So ki tu: %d\n", idem);
15	getch();
16	}
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu	

☞ **Kết quả in ra màn hình**

afser. So ki tu: 5. _	Bạn thay các dòng từ 12 đến 13 bằng câu lệnh: for(idem = 0; (c = getchar()) != DAU_CHAM; idem++); Chạy lại chương trình, quan sát và nhận xét kết quả.
-----------------------------	---

☞ **Vòng lặp for vắng mặt biểu thức 3.**

Ví dụ 7: Đọc vào một loạt kí tự trên bàn phím, đếm số kí tự nhập vào. Kết thúc khi gặp dấu chấm '.'.

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	/* Doc vao 1 loat ktu tren ban phim dem so ktu nhap vao. Ket thuc khi gap dau cham
2	
3	#include <stdio.h>
4	#include <conio.h>
5	
6	#define DAU_CHAM '.'
7	
8	void main(void)
9	{
10	char c;
11	int idem = 0;
12	for(;;)
13	{
14	c = getchar();
15	if (c == DAU_CHAM) //nhap vao dau cham
16	break; //thoat vong lap
17	idem++;
18	}
19	printf("So ki tu: %d\n", idem);
20	getch();
21	}
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu	

☞ **Kết quả in ra màn hình**

afser. So ki tu: 5. _	Chạy lại chương trình, quan sát và nhận xét kết quả.
-----------------------------	--

☞ **Vòng lặp for vắng mặt cả ba biểu thức.**

Ví dụ 8: Nhập vào 1 dãy số nguyên từ bàn phím đến khi gặp số 0 thì dừng. In ra tổng các số nguyên dương.

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	/* Nhap vao 1 day so nguyen tu ban phim den khi gap so 0 thi dung. In ra tong cac
2	nguyen duong */
3	
4	#include <stdio.h>
5	#include <conio.h>

```

6
7 void main(void)
8 {
9     int in, itong = 0;
10    for(;;)
11    {
12        printf("Nhap vao 1 so nguyen: ");
13        scanf("%d", &in);
14        if (in < 0)
15            continue;    //in < 0 quay nguoc len dau vong lap
16        if (in == 0)
17            break;    //in = 0 thoat vong lap
18        itong += in;
19    }
20    printf("Tong: %d\n", itong);
21    getch();
22 }

```

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

☞ Kết quả in ra màn hình

Nhap vao 1 so nguyen8	Chạy lại chương trình với số liệu khác
Nhap vao 1 so nguyen: 9	Quan sát và nhận xét kết quả.
Nhap vao 1 so nguyen7	
Nhap vao 1 so nguyen: 3	
Nhap vao 1 so nguyen: 0	
Tong: 12	
_	

6.2.2 Lệnh break

Thông thường lệnh break dùng để thoát khỏi vòng lặp không xác định điều kiện dừng hoặc bạn muốn dừng vòng lặp theo điều kiện do bạn chỉ định. Việc dùng lệnh break để thoát khỏi vòng lặp thường sử dụng phối hợp với lệnh if. Lệnh break dùng trong for, while, do...while, switch. Lệnh break thoát khỏi vòng lặp chứa nó.

Ví dụ 9: Như ví dụ 7, 8

Sử dụng lệnh break trong switch để nhảy bỏ các câu lệnh kế tiếp còn lại.

6.2.3 Lệnh continue

Được dùng trong vòng lặp for, while, do...while. Khi lệnh continue thi hành quyền điều khiển sẽ trao qua cho biểu thức điều kiện của vòng lặp gần nhất. Nghĩa là lộn ngược lên đầu vòng lặp, tất cả những lệnh đi sau trong vòng lặp chứa continue sẽ bị bỏ qua không thi hành.

Ví dụ 10: Như ví dụ 8

6.2.4 Lệnh while

Vòng lặp thực hiện lặp lại trong khi biểu thức còn đúng.

■ Cú pháp lệnh

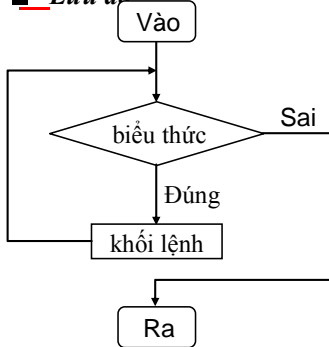
**while (biểu thức)
khối lệnh;**

☞ từ khóa **while** phải viết bằng chữ thường

☞ Nếu **khối lệnh** bao gồm từ 2 lệnh trở lên thì phải đặt trong dấu { }

Formatted: Bullets and Numbering

Lưu đồ



- ☞ Trước tiên **biểu thức** được kiểm tra
- nếu **sai** thì
- kết thúc vòng lặp while
- (khối lệnh không được thi hành 1 lần nào)
- nếu **đúng**
- thực hiện khối lệnh;
- lặp lại kiểm tra biểu thức

- + Biểu thức: có thể là một biểu thức hoặc nhiều biểu thức con. Nếu là nhiều biểu thức con thì cách nhau bởi dấu phẩy (,) và tính đúng sai của biểu thức được quyết định bởi biểu thức con cuối cùng.
- + Trong thân while (khối lệnh) có thể chứa một hoặc nhiều cấu trúc điều khiển khác.
- + Trong thân while có thể sử dụng lệnh continue để chuyển đến đầu vòng lặp (bỏ qua các câu lệnh còn lại trong thân).
- + Muốn thoát khỏi vòng lặp while tùy ý có thể dùng các lệnh **break, goto, return** như lệnh **for**.

Ví dụ 11: Viết chương trình in ra câu "Vi dụ su dung vong lap while" 3 lần.

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	/* Chuong trinh in ra cau "Vi du su dung vong lap while" 3 lan */
2	
3	#include <stdio.h>
4	#include <conio.h>
5	
6	#define MSG "Vi du su dung vong lap while"
7	
8	void main(void)
9	{
10	int i = 0;
11	while (i++ < 3)
12	printf("%s", MSG);
13	getch();
14	}
	F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

☞ **Kết quả in ra màn hình**

Vi dụ su dung vong lap while	Bạn thay 2 dòng 11 và 12 bằng câu lệnh
Vi dụ su dung vong lap while	while(printf("%s", MSG), ++i < 3);
Vi dụ su dung vong lap while	Chạy lại chương trình và quan sát kết quả.

Ví dụ 12: Viết chương trình tính tổng các số nguyên từ 1 đến n, với n được nhập vào từ bàn phím.

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	/* Chuong trinh tinh tong cac so nguyen tu 1 den n */
2	
3	#include <stdio.h>
4	#include <conio.h>
5	

```

6 void main(void)
7 {
8     int i = 0, in, is = 0;
9     printf("Nhap vao so n: ");
10    scanf("%d", &in);
11    while (i++ < in)
12        is = is + i;    //hoac is += i;
13    printf("Tong: %d", is);
14    getch();
15 }
    
```

☞ **Kết quả in ra màn hình**

Nhap vao so n : 5	Bạn thay các dòng từ 11 đến 12 bằng câu lệnh:
Tong: 15.	while(is = is+i, i++ < in);
_	Chạy lại chương trình, quan sát và nhận xét kết quả.

Ví dụ 13: Thay dòng `for(; (c = getchar()) != DAU_CHAM;)` ở ví dụ 5 thành dòng `while((c = getchar()) != DAU_CHAM)`

☞ **Chạy lại chương trình, quan sát và nhận xét kết quả.**

Ví dụ 14: Ở ví dụ 6, thay dòng `int dem;` thành dòng `int dem = 0;`, thay dòng `for(idem=0; (c = getchar()) != DAU_CHAM;)` thành dòng `while((c = getchar()) != DAU_CHAM)`

☞ **Chạy lại chương trình, quan sát và nhận xét kết quả.**

Ví dụ 15: Ở ví dụ 7 và 8, thay dòng `for(; ;)` thành dòng `while(1)`

☞ **Chạy lại chương trình, quan sát và nhận xét kết quả.**

6.2.5 Lệnh do...while

Vòng lặp thực hiện lặp lại cho đến khi biểu thức sai.

■ **Cú pháp lệnh**

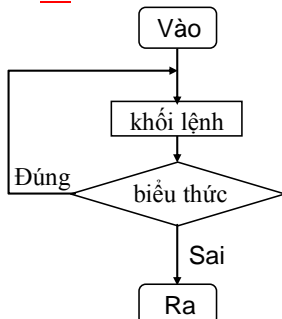
```

do
    khối lệnh;
while (biểu thức);
    
```

☞ từ khóa **do, while** phải viết bằng chữ thường

☞ Nếu **khối lệnh** bao gồm từ 2 lệnh trở lên thì phải đặt trong dấu { }

■ **Lưu đồ**



- ☞ Thực hiện khối lệnh
- Kiểm tra biểu thức
- Nếu **đúng** thì lặp lại thực hiện khối lệnh
- Nếu **sai** thì kết thúc vòng lặp (khối lệnh được thi hành 1 lần)

Formatted: Bullets and Numbering

Formatted: Bullets and Numbering

+ Biểu thức: có thể là một biểu thức hoặc nhiều biểu thức con. Nếu là nhiều biểu thức con thì cách nhau bởi dấu phẩy (,) và tính đúng sai của biểu thức được quyết định bởi biểu thức con cuối cùng.

+ Trong thân do...while (khối lệnh) có thể chứa một hoặc nhiều cấu trúc điều khiển khác.

+ Trong thân do...while có thể sử dụng lệnh continue để chuyển đến đầu vòng lặp (bỏ qua các câu lệnh còn lại trong thân).

+ Muốn thoát khỏi vòng lặp do...while tùy ý có thể dùng các lệnh **break**, **goto**, **return**.

Ví dụ 16: Viết chương trình kiểm tra password.

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	/* Chuong trinh kiem tra mat khau */
2	
3	#include <stdio.h>
4	
5	# define PASSWORD 12345
6	
7	void main(void)
8	{
9	int in;
10	do
11	{
12	printf("Nhap vao password: ");
13	scanf("%d", &in);
14	} while (in != PASSWORD)
15	}
	F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

☞ **Kết quả in ra màn hình**

Nhap vao password: 1123	Bạn thay các dòng từ 10 đến 14 bằng câu lệnh:
Nhap vao password: 1234	do{while(printf("Nhap vao password: "), scanf("%d", &in),
Nhap vao password: 1234	in != PASSWORD);
	Chạy lại chương trình và quan sát kết quả.

Ví dụ 17: Viết chương trình nhập vào năm hiện tại, năm sinh. In ra tuổi.

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	/* Chuong trinh in tuoi */
2	
3	#include <stdio.h>
4	
5	# define CHUC "Chuc ban vui ve (: \n)"
6	
7	void main(void)
8	{
9	unsigned char choi;
10	int inamhtai, inamsinh;
11	do
12	{
13	printf("Nhap vao nam hien tai: ");
14	scanf("%d", inamhtai);
15	printf("Nhap vao nam sinh: ");
16	scanf("%d", inamsinh);

17	printf("Ban %d tuoi, %s", inamhtai inamsinh, CHUC);
18	printf("Ban co muon tiep tục? (Y/N)");
19	choi = getch();
20	} while (choi == 'y' choi == 'Y');
21	}
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu	

☞ **Kết quả in ra màn hình**

Nhap vào nam hiện tại: 2002 Nhap vào năm sinh: 1980 Ban 22 tuổi, chúc bạn vui vẻ (:> Ban có muốn tiếp tục? (Y/N) _ (nếu gõ y hoặc Y tiếp tục thực hiện chương trình, ngược lại gõ các phím khác chương trình sẽ thoát)	Bạn lại chương trình với số liệu khác. Quan sát, đánh giá và nhận xét kết quả.
--	---

6.2.6 Vòng lặp lồng nhau

Ví dụ 18: Vẽ hình chữ nhật đặc bằng các dấu '*'

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	/* Vẽ hình chữ nhật đặc */
2	
3	#include <stdio.h>
4	#include <conio.h>
5	
6	void main(void)
7	{
8	int i, j, idai, irong;
9	printf("Nhap vào chiều dài: ");
10	scanf("%d", &idai);
11	printf("Nhap vào chiều rộng: ");
12	scanf("%d", &irong);
13	for (i = 1; i <= irong; i++)
14	{
15	for (j = 1; j <= idai; j++) //in một hàng với chiều dài dấu *
16	printf("*");
17	printf("\n"); //xuống dòng khi in xong 1 hàng
18	}
19	getch();
20	}
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu	

☞ **Kết quả in ra màn hình**

Nhap vào chiều dài: 10 Nhap vào chiều rộng: 5 ***** ***** ***** ***** ***** _	Bạn lại chương trình với số liệu khác. Quan sát, đánh giá và nhận xét kết quả.
--	---

Ví dụ 19: Vẽ hình chữ nhật đặc có chiều rộng = 10 hàng. Hàng thứ 1 = 10 số 0, hàng thứ 2 = 10 số 1...

Dòng	File	Edit	Search	Run	Compile	Debug	Project	Option	Window	Help
1	/* Vẽ hình chu nhật bằng các số từ 0 đến 9 */									
2										
3	#include <stdio.h>									
4	#include <conio.h>									
5										
6	void main(void)									
7	{									
8	int i = 0, ij;									
9	while (i <= 9)									
10	{									
11	ij = 0; //khởi tạo lại ij = 0 cho lần in kế tiếp									
12	while (ij++ <= 9) //in 1 hàng 10 số i									
13	printf("%d", i);									
14	printf("\n"); //xuống dòng khi in xong 1 hàng									
15	i++; //tăng i lên 1 cho vòng lặp kế tiếp									
16	}									
17	getch();									
18	}									
19										
<p>F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu</p>										

Kết quả in ra màn hình

0 0 0 0 0 0 0 0 0 0	Thay dòng 11, 12 thành câu lệnh for (ij = 0; ij <= 9; ij++) Chạy lại chương trình. Quan sát, đánh giá và nhận xét kết quả.
1 1 1 1 1 1 1 1 1 1	
2 2 2 2 2 2 2 2 2 2	
3 3 3 3 3 3 3 3 3 3	
4 4 4 4 4 4 4 4 4 4	
5 5 5 5 5 5 5 5 5 5	
6 6 6 6 6 6 6 6 6 6	
7 7 7 7 7 7 7 7 7 7	
8 8 8 8 8 8 8 8 8 8	
9 9 9 9 9 9 9 9 9 9	

Các lệnh lặp for, while, do...while có thể lồng vào chính nó, hoặc lồng vào lẫn nhau. Nếu không cần thiết không nên lồng vào nhiều cấp để gây nhầm lẫn khi lập trình cũng như kiểm soát chương trình.

6.2.7 So sánh sự khác nhau của các vòng lặp

- Vòng lặp for thường sử dụng khi biết được số lần lặp xác định.
- Vòng lặp thường while, do...while sử dụng khi không biết rõ số lần lặp.
- Khi gọi vòng lặp while, do...while, nếu biểu thức sai vòng lặp while sẽ không được thực hiện lần nào nhưng vòng lặp do...while thực hiện được 1 lần.

Số lần thực hiện ít nhất của while là 0 và của do...while là 1

6.3 Bài tập

1. Viết chương trình in ra bảng mã ASCII

2. *Viết chương trình tính tổng bậc 3 của N số nguyên đầu tiên.*
3. *Viết chương trình nhập vào một số nguyên rồi in ra tất cả các ước số của số đó.*
4. *Viết chương trình vẽ một tam giác cân bằng các dấu **
5. *Viết chương trình tính tổng nghịch đảo của N số nguyên đầu tiên theo công thức*

$$S = 1 + 1/2 + 1/3 + \dots + 1/N$$
6. *Viết chương trình tính tổng bình phương các số lẻ từ 1 đến N.*
7. *Viết chương trình nhập vào N số nguyên, tìm số lớn nhất, số nhỏ nhất.*
8. *Viết chương trình nhập vào N rồi tính giai thừa của N.*
9. *Viết chương trình tìm USCLN, BSCNN của 2 số.*
10. *Viết chương trình vẽ một tam giác cân rộng bằng các dấu *.*
11. *Viết chương trình vẽ hình chữ nhật rộng bằng các dấu *.*
12. *Viết chương trình nhập vào một số và kiểm tra xem số đó có phải là số nguyên tố hay không?*
13. *Viết chương trình tính số hạng thứ n của dãy Fibonacci.*
 Dãy Fibonacci là dãy số gồm các số hạng $p(n)$ với:
 $p(n) = p(n-1) + p(n-2)$ với $n > 2$ và $p(1) = p(2) = 1$
 Dãy Fibonacci sẽ là: 1 1 2 3 5 8 13 21 34 55 89 144...
14. *Viết chương trình tính giá trị của đa thức*

$$P_n = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x^1 + a_0$$
 Hướng dẫn đa thức có thể viết lại

$$P_n = (\dots(a_n x + a_{n-1})x + a_{n-2})x + \dots + a_0$$
 Như vậy trước tiên tính $a_n x + a_{n-1}$, lấy kết quả nhân với x, sau đó lấy kết quả nhân với x cộng thêm a_{n-2} , lấy kết quả nhân với x ... n gọi là bậc của đa thức.
15. *Viết chương trình tính x^n với x, n được nhập vào từ bàn phím.*
16. *Viết chương trình nhập vào 1 số từ 0 đến 9. In ra chữ số tương ứng. Ví dụ: nhập vào số 5, in ra "Năm".*
17. *Viết chương trình phân tích một số nguyên N thành tích của các thừa số nguyên tố.*
18. *Viết chương trình lặp lại nhiều lần công việc nhập một ký tự và in ra mã ASCII của ký tự đó, khi nào nhập số 0 thì dừng.*
19. *Viết chương trình tìm ước số chung lớn nhất và bội số chung nhỏ nhất của 2 số nguyên.*
20. *Viết chương trình in lá cờ nước Mỹ.*
21. *Viết chương trình tính dân số của một thành phố sau 10 năm nữa, biết rằng dân số hiện nay là 6.000.000, tỉ lệ tăng dân số hàng năm là 1.8% .*
22. *Viết chương trình tìm các số nguyên gồm 3 chữ số sao cho tích của 3 chữ số bằng tổng 3 chữ số. Ví dụ: $1*2*3 = 1+2+3$.*
23. *Viết chương trình tìm các số nguyên a, b, c, d khác nhau trong khoảng từ 0 tới 10 thỏa mãn điều kiện $a*d*d = b*c*c*c$*
24. *Viết chương trình tính tổ hợp N chập K (với $K \leq N$)*

$$C = ((N-k+1) * (N-k+2) * \dots * N) / (1*2*3* \dots * k)$$

Trong đó C là một tích gồm k phần tử với phần tử thứ I là $(N-k+1)/I$. Để viết chương trình này, bạn dùng vòng lặp For với biến điều khiển I từ giá trị đầu là 1 tăng đến giá trị cuối là k kết hợp với việc nhân dồn vào kết quả C.

25. Viết chương trình giải bài toán cổ điển sau:

Trăm trâu, trăm cỏ
 Trâu đực ăn năm
 Trâu nái ăn ba,
 Ba trâu già ăn một
 Hỏi mỗi loại trâu có bao nhiêu con.

26. Viết chương trình giải bài toán cổ điển sau:

Vừa gà vừa chó 36 con
 Bó lại cho tròn, đếm đủ 100 chân
 Hỏi có bao nhiêu gà, bao nhiêu chó

27. Viết chương trình in ra bảng cửu chương

28. Viết chương trình xác định xem một tờ giấy có độ dày 0.1 mm. Phải gấp đôi tờ giấy bao nhiêu lần để nó có độ dày 1m.

29. Viết chương trình tìm các số nguyên tố từ 2 đến N, với N được nhập vào.

30. Viết chương trình lặp đi lặp lại các công việc sau:

- Nhập vào một ký tự trên bàn phím.
- Nếu là chữ thường thì in ra chính nó và chữ HOA tương ứng.
- Nếu là chữ HOA thì in ra chính nó và chữ thường tương ứng.
- Nếu là ký số thì in ra chính nó.
- Nếu là một ký tự điều khiển thì kết thúc chương trình

31. Viết chương trình nhập vào x, n tính:

- $\sqrt{x \cdot \sqrt{x \cdot \sqrt{x \cdot \dots \sqrt{x}}}}$ (n dấu căn)
- $1 + \frac{x}{2} \cdot \frac{x^n}{n}$

32. Viết chương trình nhập vào N số nguyên, đếm xem có bao nhiêu số âm, bao nhiêu số dương và bao nhiêu số không.



Bài 7 :**HÀM****7.1 Mục tiêu**

Sau khi hoàn tất bài này học viên sẽ hiểu và vận dụng các kiến thức kỹ năng cơ bản sau:


- Khái niệm, cách khai báo về hàm.
- Cách truyền tham số, tham biến, tham trị.
- Sử dụng biến cục bộ, toàn cục trong hàm.
- Sử dụng tiền xử lý #define

7.2 Nội dung

Hàm là một chương trình con thực hiện một khối công việc được lặp đi lặp lại nhiều lần trong khi chạy chương trình hoặc dùng tách một khối công việc cụ thể để chương trình đỡ phức tạp.

7.2.1 Các ví dụ về hàm**Ví dụ 1:**

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	#include <stdio.h>
2	#include <conio.h>
3	
4	// khai báo prototype
5	void line();
6	
7	// hàm in 1 dòng dấu
8	void line()
9	{
10	int i;
11	for(i = 0; i < 19; i++)
12	printf("*");
13	printf("\n");
14	}
15	
16	void main(void)
17	{
18	line();
19	printf("* Minh hoa ve ham\n*");
20	line();
21	getch();
22	}
	F1 Help Alt-F8 Next Msg Alt-F7 PrevMsg Alt - F9 Compile F9 Make F10 Menu

 Kết quả in ra màn hình

```
*****
* Minh hoa ve ham *
*****
```

—

Giải thích chương trình

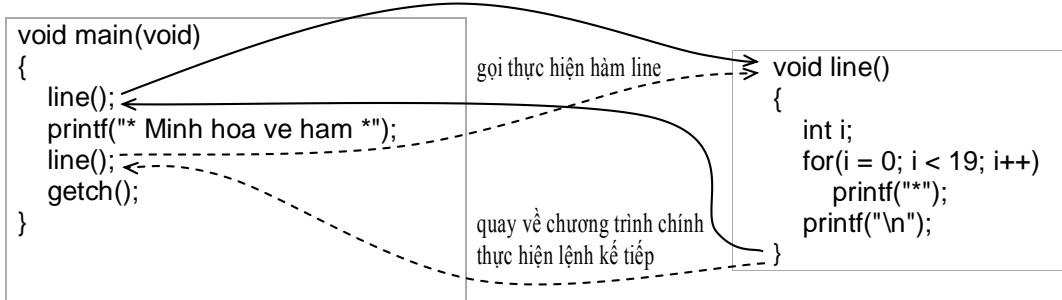
Dòng 8 đến dòng 14: định nghĩa hàm **line**, hàm này không trả về giá trị, thực hiện công việc in ra 19 dấu sao.

Dòng 5: khai báo prototype (nguyên mẫu_by mr Linh) sau tên hàm phải có dấu chấm phẩy

Trong hàm line có sử dụng biến i, biến i là biến cục bộ chỉ sử dụng được trong phạm vi hàm line.

Dòng 18 và 20: gọi thực hiện hàm line.

*** Trình tự thực hiện chương trình**



⊗ Không có dấu chấm phẩy sau tên hàm, phải có cặp dấu ngoặc () sau tên hàm nếu hàm không có tham số truyền vào. Phải có dấu chấm phẩy sau tên hàm khai báo prototype. Nên khai báo prototype cho dù hàm được gọi nằm trước hay sau câu lệnh gọi nó.

Ví dụ 2:

Dòng	File	Edit	Search	Run	Compile	Debug	Project	Option	Window	Help
1	#include <stdio.h>									
2	#include <conio.h>									
3										
4	// khai bao prototype									
5	int power(int, int);									
6										
7	// ham tinh so mu									
8	int power(int ix, int in)									
9	{									
10	int i, ip = 1;									
11	for(i = 1; i <= in; i++)									
12	ip *= ix;									
13	return ip;									
14	}									
15										
16	void main(void)									
17	{									
18	printf("2 mu 2 = %dn", power(2, 2));									
19	printf("2 mu 3 = %dn", power(2, 3));									
20	getch();									
	}									
	F1 Help	Alt-F8 Next Msg	Alt-F7 Prev Msg	Alt - F9 Compile	F9 Make	F10 Menu				

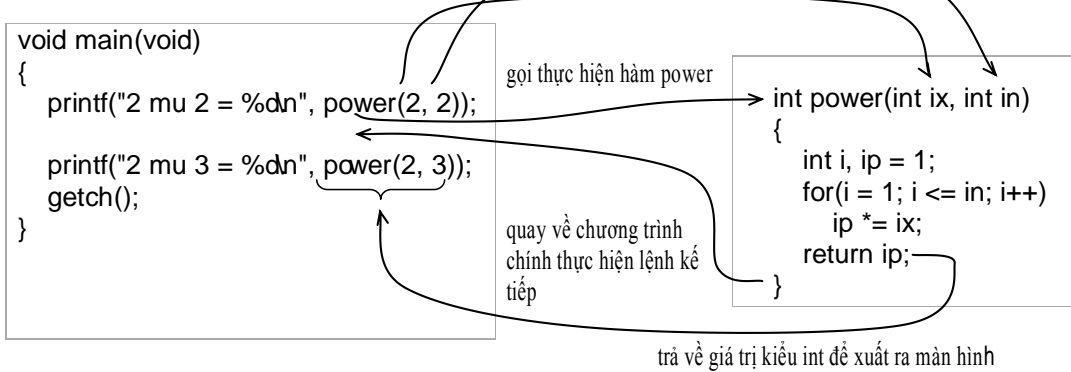
👉 Kết quả in ra màn hình

```
2 mu 2 = 4.
2 mu 3 = 8.
-
```

🔗 Giải thích chương trình

Hàm **power** có hai tham số truyền vào là *ix*, *in* có kiểu *int* và kiểu trả về cũng có kiểu *int*.
 Dòng 13: `return ip;` trả về giá trị sau khi tính toán
 Dòng 18: đối mục 2 và 3 có kiểu trả về là *int* sau khi thực hiện gọi `power`.
 Hai tham số *ix*, *in* của hàm `power` là dạng truyền tham trị.

*** Trình tự thực hiện chương trình**



👉 Quy tắc đặt tên hàm giống tên biến, hằng... Mỗi đối số cách nhau = dấu phẩy kèm theo kiểu dữ liệu tương ứng.

Ví dụ 3:

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	#include <stdio.h>
2	#include <conio.h>
3	
4	// khai bao prototype
5	void time(int & , int &);
6	
7	// ham doi phut thanh gio:phut
8	void time(int &ig, int &ip)
9	{
10	ig = ip / 60;
11	ip %= 60;
12	}
13	
14	void main(void)
15	{
16	int igio, iphut;

```

17 printf("Nhap vao so phut : ");
18 scanf("%d", &iphut);
19 time(igio, iphut);
20 printf("%02d:%02d\n", igio, iphut);
21 getch();
22 }
    
```

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

Kết quả in ra màn hình

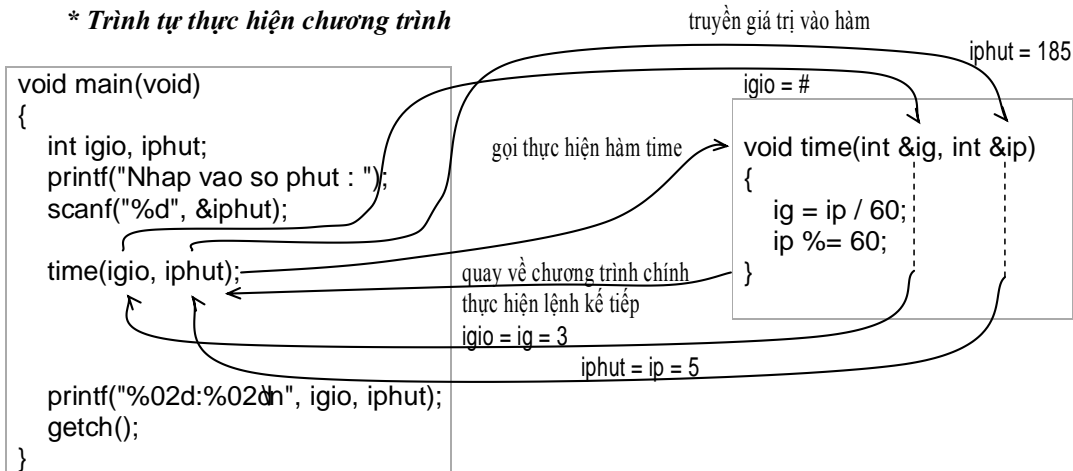
```

Nhap vao so phut: 185
03:05
-
    
```

Giải thích chương trình

Hàm `time` có hai tham số truyền vào là `ix`, `in` có kiểu `int`. 2 tham số này có toán tử địa chỉ & đi trước cho biết 2 tham số này là dạng truyền tham biến.

*** Trình tự thực hiện chương trình**



7.2.2 Tham số dạng tham biến và tham trị

Ví dụ 4:


<pre> void thamtri(int ix, int iy) { ix += 1; //cong ix them 1 iy += 1; //cong iy them 1 } void thambien(int &ix, int &iy) { ix += 1; //cong ix them 1 iy += 1; //cong iy them 1 } </pre>	<pre> void main(void) { int ia = 5, ib = 5; thamtri(ia, ib); printf("a = %d, b = %d", ia, ib); thambien(ia, ib); printf("a = %d, b = %d", ia, ib); } </pre>	<p>Kết quả in ra:</p> <pre> a = 5, b = 5 a = 6, b = 6 </pre>
---	---	--

Đối với hàm sử dụng lệnh return bạn chỉ có thể trả về duy nhất 1 giá trị mà thôi. Để có thể trả về nhiều giá trị sau khi gọi hàm bạn sử dụng hàm truyền nhiều tham số dạng tham biến.

7.2.3 Sử dụng biến toàn cục

Ví dụ 5:

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	#include <stdio.h>
2	#include <conio.h>
3	
4	// khai bao prototype
5	void oddeven();
6	void negative();
7	
8	//khai bao bien toan cuc
9	int inum;
10	
11	void main(void)
12	{
13	printf("Nhap vao 1 so nguyen : ");
14	scanf("%d", &inum);
15	oddeven();
16	negative();
17	getch();
18	}
19	
20	// ham kiem tra chan le
21	void oddeven()
22	{
23	if (inum % 2)
24	printf("%d la so le.\n", inum);
25	else
26	printf("%d la so chan.\n", inum);
27	}
28	
29	//ham kiem tra so am
30	void negative()
31	{
32	if (inum < 0)
33	printf("%d la so am.\n", inum);
34	else
35	printf("%d la so duong.\n", inum);
36	}
	F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

 **Kết quả in ra màn hình**

```
Nhap vao 1 so nguyen: 3
3 la so le.
3 la so duong.
_
```


Giải thích chương trình

Chương trình trên gồm 2 hàm **oddeven** và **negative**, 2 hàm này bạn thấy không có tham số để truyền biến inum vào xử lý nhưng vẫn cho kết quả đúng. Do chương trình sử dụng biến **inum** toàn cục (đòng.9) nên biến này có ảnh hưởng đến toàn bộ chương trình mỗi khi gọi và sử dụng nó. Xét tình huống sau: Giả sử trong hàm **negative** ta khai báo biến inum có kiểu int như sau:

```
void negative()
{
    int inum;
    ....
}
```

Khi đó chương trình sẽ cho kết quả sai! Do các câu lệnh trong hàm **negative** sử dụng biến inum sẽ sử dụng biến inum khai báo trong hàm **negative** và lúc này biến inum toàn cục không có tác dụng đối với các câu lệnh trong hàm này. Biến inum khai báo trong hàm **negative** chỉ có ảnh hưởng trong phạm vi hàm và chu trình sống của nó bắt đầu từ lúc gọi hàm đến khi thực hiện xong.

Cẩn thận khi đặt tên biến, xác định rõ phạm vi của biến khi sử dụng để có thể dễ dàng kiểm soát chương trình.

Ví dụ 6:

Dòng	File	Edit	Search	Run	Compile	Debug	Project	Option	Window	Help
1	#include <stdio.h>									
2	#include <conio.h>									
3										
4	#define PI 3.14									
5										
6	// khai bao prototype									
7	float area();									
8										
9	//khai bao bien toan cuc									
10	float frad;									
11										
12	void main(void)									
13	{									
14	printf("Nhap vao ban kinh hinh cau : ");									
15	scanf("%f", &frad);									
16	printf("Dien tich hinh cau: %10.3f", area());									
17	getch();									
18	}									
19										
20	// ham tinh dien tich hinh cau									
21	float area()									
22	{									
23	return (4*PI*frad*frad);									
24	}									
	F1 Help	Alt-F8 Next Msg	Alt-F7 Prev Msg	Alt - F9 Compile	F9 Make	F10 Menu				

Kết quả in ra màn hình

```
Nhap vao ban kinh hinh cau: 3.2
Dien tich hinh cau: 128.614
```

7.2.4 Dùng dẫn hướng #define

Sau đây là một vài ví dụ dùng dẫn hướng #define để định nghĩa hàm đơn giản

```
#define AREA_CIRCLE (frac) (4*PI*frac*frac) //tinh dien tich hinh cau
#define SUM (x, y) (x + y) //cong 2 so
#define SQR (x) (x*x) //tinh x binh phuong
#define MAX(x, y) (x > y) ? x : y //tim so lon nhâ giữa x va y
#define ERROR (s) printf("%s\n", s) //in thong bao voi chuoai s
```

Ví dụ 7: Trong ví dụ 6 xóa từ dòng 20 đến dòng 24, xóa dòng 6, 7; thêm dòng **AREA_CIRCLE (frac) (4*PI+frac*frac)** vào sau dòng 5.

Sửa dòng **printf("Dien tich hinh cau: %10.3f\n", area());** thành **printf("Dien tich hinh cau: %10.3f\n", AREA_CIRCLE(frac));**

Chạy lại chương trình, quan sát và nhận xét kết quả.

Ví dụ 8:

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	#include <stdio.h>
2	#include <conio.h>
3	
4	#define MAX(x, y) (x > y) ? x : y
5	
6	void main(void)
7	{
8	float a = 4.5, b = 6.1;
9	printf("So lon nhat la: %5.2f", MAX(a, b));
10	getch();
11	}
	F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

Kết quả in ra màn hình

So lon nhat la: 6.10	Thêm vào dòng 8 giá trị c = 10
–	Sửa lại dòng 9: MAX(a, b) thành MAX(MAX(a, b), c)
	Chạy lại chương trình, quan sát và nhận xét kết quả

7.3 Bài tập

1. *Viết hàm tính n!*
2. *Viết hàm tính tổng $S = 1+2+\dots+n$.*
3. *Viết hàm kiểm tra số nguyên tố.*
4. *Viết hàm tính số hạng thứ n trong dãy Fibonacci.*
5. *Viết hàm tìm số lớn nhất trong 2 số.*

Bài 8 : MẢNG VÀ CHUỖI

8.1 Mục tiêu

Sau khi hoàn tất bài này học viên sẽ hiểu và vận dụng các kiến thức kỹ năng cơ bản sau:

- Ý nghĩa, cách khai báo mảng, chuỗi.
- Nhập, xuất mảng, chuỗi.
- Khởi tạo mảng chuỗi.
- Một số kỹ thuật thao tác trên mảng, chuỗi.
- Dùng mảng làm tham số cho hàm.
- Một số hàm xử lý chuỗi

8.2 Nội dung

8.2.1 Mảng

Là tập hợp các phần tử có cùng dữ liệu. Giả sử bạn muốn lưu n số nguyên để tính trung bình, bạn không thể khai báo n biến để lưu n giá trị rồi sau đó tính trung bình.

Ví dụ 1 : bạn muốn tính trung bình 10 số nguyên nhập vào từ bàn phím, bạn sẽ khai báo 10 biến: a, b, c, d, e, f, g, h, i, j có kiểu int và lập thao tác nhập cho 10 biến này như sau:

```
printf("Nhập vào biến a: ");
```

```
scanf("%d", &a);
```

10 biến bạn sẽ thực hiện 2 lệnh trên 10 lần, sau đó tính trung bình:

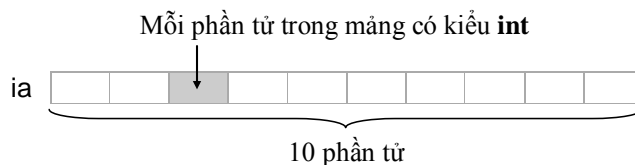
```
(a + b + c + d + e + f + g + h + i + j)/10
```

☞ Điều này chỉ phù hợp với n nhỏ, còn đối với n lớn thì khó có thể thực hiện được. Vì vậy khái niệm mảng được sử dụng

8.2.1.1 Cách khai báo mảng

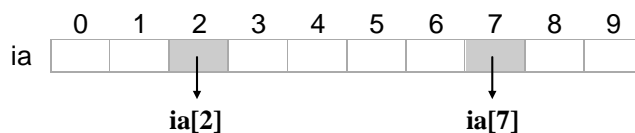
Ví dụ 2 : `int ia[10];` với `int` là kiểu mảng, `ia` là tên mảng, 10 số phần tử mảng

Ý nghĩa: *Khai báo một mảng số nguyên gồm 10 phần tử, mỗi phần tử có kiểu int.*



8.2.1.2 Tham chiếu đến từng phần tử mảng

Sau khi mảng được khai báo, mỗi phần tử trong mảng đều có chỉ số để tham chiếu. Chỉ số bắt đầu từ 0 đến n-1 (với n là kích thước mảng). Trong ví dụ trên, ta khai báo mảng 10 phần tử thì chỉ số bắt đầu từ 0 đến 9.



`ia[2]`, `ia[7]`... là phần tử thứ 3, 8... trong mảng xem như là một biến kiểu `int`.

8.2.1.3 Nhập dữ liệu cho mảng


```
for (i = 0; i < 10; i++) //vòng for 6 giá trị i chạy từ 0 đến 9
{
    printf("Nhap vao phan tu thu %d: ", i + 1);
    scanf("%d", &ia[i]);
}
```

8.2.1.4 Đọc dữ liệu từ mảng

```
for(i = 0; i < 10; i++)
    printf("%3d ", ia[i]);
```

Ví dụ 3 : Viết chương trình nhập vào n số nguyên. Tính và in ra trung bình cộng.

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	/* Tinh trung binh cong n so nguyen */
2	
3	#include <stdio.h>
4	#include <conio.h>
5	
6	void main(void)
7	{
8	int ia[50], i, in, isum = 0;
9	printf("Nhap vao gia tri n: ");
10	scanf("%d", &in);
11	
12	//Nhap du lieu vao mang
13	for(i = 0; i < in; i++)
14	{
15	printf("Nhap vao phan tu thu %d: ", i + 1);
16	scanf("%d", &ia[i]); //Nhap gia tri cho phan tu thu i
17	}
18	
19	//Tinh tong gia tri cac phan tu
20	for(i = 0; i < in; i++)
21	isum += ia[i]; //cong don tung phan tu vao isum
22	
23	printf("Trung binh cong: %.2f", (float) isum/in);
24	getch();
25	}
	F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

 **Kết quả in ra màn hình**

Nhap vao gia tri n: 3 Nhap vao phan tu thu 1: 7 Nhap vao phan tu thu 2: 3 Nhap vao phan tu thu 3: 6 Trung binh cong: 5.33 —	Bạn có thể gộp 2 lệnh for thành một vừa nhập vừa tính tổng, đưa hàng 21 sau hàng 16 và bỏ các hàng 19, 20, 21. Chạy và quan sát kết quả.
--	---

⊗ Điều gì sẽ xảy ra cho đoạn chương trình trên nếu bạn nhập $n > 50$ trong khi bạn chỉ khai báo mảng ia tối đa là 50 phần tử. Bạn dùng lệnh if để ngăn chặn điều này trước khi vào thực hiện lệnh for. Thay dòng 9, 10 bằng đoạn lệnh sau :

```
do
{
    printf("Nhập vào giá trị n: ");
    scanf("%d", &in);
} while (in <= 0 || in > 50); //chỉ chấp nhận giá trị nhập vào trong khoảng 1..50
```

☞ Chạy chương trình và nhập n với các giá trị -6, 0, 51, 6. Quan sát kết quả.

8.2.1.5 Sử dụng biến kiểu khác

Ngoài kiểu int, bạn có thể khai báo mảng kiểu char, float, double...

Ví dụ 4 : char cloai[20]; float ftemp[10]; cách tham chiếu, nhập dữ liệu, đọc dữ liệu như trên.

8.2.1.6 Kỹ thuật Sentinel

Sử dụng kỹ thuật này để nhập liệu giá trị cho các phần tử mảng mà không biết rõ số lượng phần tử sẽ nhập vào là bao nhiêu (không biết số n).

Ví dụ 5 : Viết chương trình nhập vào 1 dãy số dương rồi in tổng các số dương đó.

Phác họa lời giải: Chương trình yêu cầu nhập vào dãy số dương mà không biết trước số lượng phần tử cần nhập là bao nhiêu, vì vậy để chấm dứt nhập liệu khi thỏa mãn bằng cách nhập vào số âm hoặc không.

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	/* Nhập vào dãy số nguyên dương, in ra dãy chẵn, dãy lẻ */
2	
3	#include <stdio.h>
4	#include <conio.h>
5	#define MAX 50
6	
7	void main(void)
8	{
9	float fa[MAX], fsum = 0;
10	int i = 0;
11	do
12	{
13	printf("Nhập vào phần tử thứ %d: ", i + 1);
14	scanf("%f", &fa[i]); //Nhập giá trị cho phần tử thứ i
15	} while (fa[i++] > 0); //con nhập liệu khi giá trị phần tử > 0
16	
17	i--; //giảm i đi 1 lần cuối cùng tăng 1 trước khi thoát
18	//Tinh tong
19	for(int ij = 0; ij < i; ij++)
20	fsum += fa[ij]; //cong don tung phan tu vao isum
21	
22	printf("Tong : %5.2f\n", fsum);
23	getch();
24	}
	F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

☞ **Kết quả in ra màn hình**

Nhập vào phần tử thu 1: 1.	Bạn chạy lại chương trình và thử lại với số liệu khác. Quan sát kết quả.
Nhập vào phần tử thu 2: 3	
Nhập vào phần tử thu 3: 4.	
Nhập vào phần tử thu 4: 9	
Tổng : 8.80	
—	

☹ Điều gì sẽ xảy ra cho đoạn chương trình trên nếu bạn nhập số lượng phần tử vượt quá 50 trong khi bạn chỉ khai báo mảng fa tối đa là MAX = 50 phần tử. Bạn dùng lệnh break để thoát khỏi vòng lặp do...while trước khi bước sang phần tử thứ 51. Thêm đoạn lệnh sau vào trước dòng 13:

```

if (i >= MAX)           //kiem tra phần tử bước sang 51
{
    printf("Mảng đã đầy\n"); //thông báo "Mảng đã đầy"
    i++;                  //tăng i lên 1 do dòng 17 giảm i xuống 1
    break;               //thoát khỏi vòng lặp do...while
}

```

☞ Sửa dòng 5 thành #define MAX 4. Chạy chương trình và nhập các số 1, 2, 3, 5, 6, 5, 4. Quan sát kết quả.

8.2.1.7 Khởi tạo mảng

Vi dụ 6 : Có 4 loại tiền 1, 5, 10, 25 và 50 đồng. Hãy viết chương trình nhập vào số tiền sau đó cho biết số tiền trên gồm mấy loại tiền, mỗi loại bao nhiêu tờ.

Phác họa lời giải: Số tiền là 246 đồng gồm 4 tờ 50 đồng, 1 tờ 25 đồng, 2 tờ 10 đồng, 0 tờ 5 đồng và 1 tờ 1 đồng, Nghĩa là bạn phải xét loại tiền lớn trước, nếu hết khả năng mới xét tiếp loại kế tiếp.

Dòng	File	Edit	Search	Run	Compile	Debug	Project	Option	Window	Help
1	/* Nhập vào số tiền và đổi tiền ra các loại 50, 25, 10, 5, 1 */									
2										
3	#include <stdio.h>									
4	#include <conio.h>									
5	#define MAX 5									
6										
7	void main(void)									
8	{									
9	int itien[MAX] = {50, 25, 10, 5, 1}; //Khai báo và khởi tạo mảng với 5 phần tử									
10	int i, isotien, ito;									
11	printf("Nhập vào số tiền: ");									
12	scanf("%d", &isotien); //Nhập vào số tiền									
13	for (i = 0; i < MAX; i++)									
14	{									
15	ito = isotien/itien[i]; //Tìm số tờ của loại tiền thu i									
16	printf("%4d tờ %2d đồng", ito, itien[i]);									
17	isotien = isotien%itien[i]; //Số tiền còn lại sau khi đã loại trừ các loại tiền đã có									
18	}									

```

19 | getch()
20 | }

```

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

Kết quả in ra màn hình

Nhập vào số tiền: 246 4 tờ 50 đồng 1 tờ 25 đồng 2 tờ 10 đồng 0 tờ 5 đồng 1 tờ 1 đồng _	Bạn chạy lại chương trình và thử lại với số liệu khác. Quan sát kết quả.
--	---

Điều gì sẽ xảy nếu số phần tử mảng lớn hơn số mục, số phần tử dôi ra không được khởi tạo sẽ điền vào số 0. Nếu số phần tử nhỏ hơn số mục khởi tạo trình biên dịch sẽ báo lỗi.

Ví dụ 7: int itien[5] = {50, 25}, phần tử itien[0] sẽ có giá trị 50, itien[1] có giá trị 25, itien[2], itien[3], itien[4] có giá trị 0.
 int itien[3] = {50, 25, 10, 5, 1} trình biên dịch báo lỗi

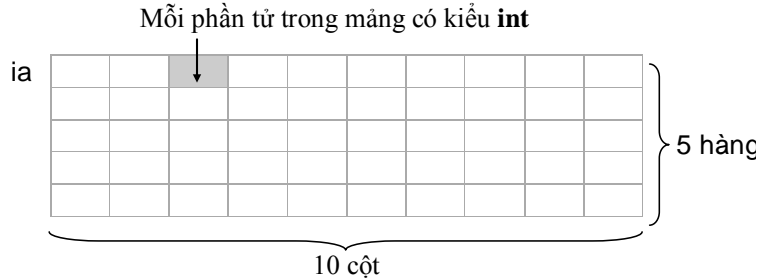
8.2.1.8 Khởi tạo mảng không bao hàm kích thước

Trong ví dụ trên giả sử ta khai báo int itien[] = {50, 25, 10, 5, 1}. Khi đó trình biên dịch sẽ đếm số mục trong danh sách khởi tạo và dùng con số đó làm kích thước mảng.

8.2.1.9 Mảng nhiều chiều

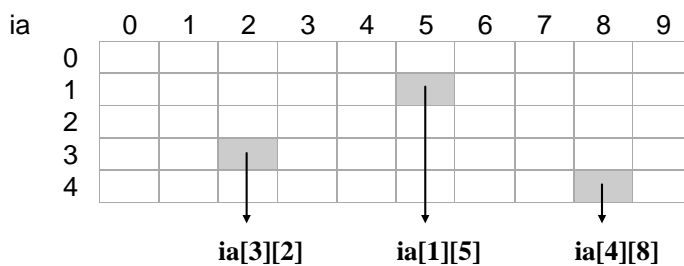
Ví dụ 8: khai báo mảng 2 chiều int ia[5][10]; với int là kiểu mảng, ia là tên mảng, số phần tử mảng là 5 x 10.

Ý nghĩa: **Khai báo một mảng 2 chiều số nguyên gồm 50 phần tử, mỗi phần tử có kiểu int.**



8.2.1.10 Tham chiếu đến từng phần tử mảng 2 chiều

Sau khi được khai báo, mỗi phần tử trong mảng 2 chiều đều có 2 chỉ số để tham chiếu, chỉ số hàng và chỉ số cột. Chỉ số hàng bắt đầu từ 0 đến số hàng - 1 và chỉ số cột bắt đầu từ 0 đến số cột - 1. Tham chiếu đến một phần tử trong mảng 2 chiều ia: ia[chỉ số hàng][chỉ số cột]



ia[3][2] là phần tử tại hàng 3 cột 2 trong mảng 2 chiều xem như là một biến kiểu `int`.

8.2.1.11 Nhập dữ liệu cho mảng 2 chiều

```
for (i = 0; i < 5; i++)          //vòng for có giá trị i chạy từ 0 đến 4 cho hàng
  for (ij = 0; ij < 10; ij++)    //vòng for có giá trị ij chạy từ 0 đến 9 cho cột
  {
    printf("Nhap vao phan tu ia[%d][%d]: ", i + 1, ij + 1);
    scanf("%d", &ia[i][ij]);
  }
```

* Thứ tự nhập dữ liệu vào mảng 2 chiều



8.2.1.12 Đọc dữ liệu từ mảng 2 chiều


Ví dụ 9 : in giá trị các phần tử mảng 2 chiều ra màn hình.

```
for (i = 0; i < 5; i++)          //vòng for có giá trị i chạy từ 0 đến 4 cho hàng
{
  for (ij = 0; ij < 10; ij++)    //vòng for có giá trị ij chạy từ 0 đến 9 cho cột
    printf("%3d ", ia[i][ij]);
  printf("\n");                  //xuống dòng để in hàng kế tiếp
}
```


Ví dụ 10 : Viết chương trình nhập vào 1 ma trận số nguyên $n \times n$. In ra ma trận vừa nhập vào và ma trận theo thứ tự ngược lại.

Dòng	File	Edit	Search	Run	Compile	Debug	Project	Option	Window	Help
1	/* Tinh trung binh cong n so nguyen */									
2										
3	#include <stdio.h>									
4	#include <conio.h>									
5	#define MAX 50;									
6										
7	void main(void)									
8	{									
9	int ia[MAX][MAX], i, ij, in;									
10	printf("Nhap vao cap ma tran: ");									
11	scanf("%d", &in);									
12										
13	//Nhap du lieu vao ma tran									
14	for (i = 0; i < in; i++) //vòng for có giá trị i chạy từ 0 đến in-1 cho hàng									
15	for (ij = 0; ij < in; ij++) //vòng for có giá trị ij chạy từ 0 đến in-1 cho cột									
16	{									
17	printf("Nhap vao phan tu ia[%d][%d]: ", i + 1, ij + 1);									
18	scanf("%d", &ia[i][ij]);									
19	}									
20										
21	//In matran									

22	for (i = 0; i < in; i++)	//vòng for có giá trị i chạy từ 0 đến in-1 cho hàng
23	{	
24	for (ij = 0; ij < in; ij++)	//vòng for có giá trị ij chạy từ 0 đến in-1 cho cột
25	printf("%3d ", ia[i][ij]);	
26	printf("\n");	//xuống dòng để in hàng kế tiếp
27	}	
28	printf("\n");	//Tạo khoảng cách giữa 2 ma trận
29		
30	//In ma trận theo thu tự ngược	
31	for (i = in-1; i >= 0; i--)	//vòng for có giá trị i chạy từ in-1 đến 0 cho hàng
32	{	
33	for (ij = in-1; ij >= 0; ij--)	//vòng for có giá trị ij chạy từ in-1 đến 0 cho cột
34	printf("%3d ", ia[i][ij]);	
35	printf("\n");	//xuống dòng để in hàng kế tiếp
36	}	
37	getch();	
38	}	
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu		

 **Kết quả in ra màn hình**

Nhập vào cấp ma trận: 2	Chạy và thử lại chương trình với n = 3, 5.
Nhập vào phần tử ia[1][1]: 7	Quan sát kết quả.
Nhập vào phần tử ia[1][2]: 4	- Sửa lại chương trình trên cho phép nhập vào ma trận m x n.
Nhập vào phần tử ia[2][1]: 6	Nghĩa là ma trận có m hàng và n cột. Bạn sửa lại chương trình
Nhập vào phần tử ia[2][2]: 15	bằng cách cho nhập vào giá trị m và n và sửa lại vòng for cho
7 4	hàng chạy m lần và vòng for cho cột chạy n lần.
6 15	
15 6	
4 7	
—	

 **Để khắc phục tình trạng người dùng nhập vào cấp ma trận > MAX, Bạn xem lại mục 3.1.4.**

8.2.1.13 Sử dụng biến kiểu khác trong mảng 2 chiều

Như mục 3.1.5.

8.2.1.14 Khởi tạo mảng 2 chiều

Ví dụ 11 : Vẽ chữ H lớn.


Dòng	File	Edit	Search	Run	Compile	Debug	Project	Option	Window	Help
1	/* Chương trình vẽ chữ H lớn */									
2										
3	#include <stdio.h>									
4	#include <conio.h>									
5	#define MAX 5									
6										
7	int H[MAX][MAX] = {{1, 0, 0, 0, 1},									
8	{1, 0, 0, 0, 1},									

```

9           {1, 1, 1, 1, 1},
10          {1, 0, 0, 0, 1},
11          {1, 0, 0, 0, 1}};
12 void main(void)
13 {
14     int i , ij;
15     for (i = 0; i < MAX; i++)
16     {
17         for (ij = 0; ij < MAX; ij++)
18             if (H[i][ij])
19                 printf("!");
20             else
21                 printf(" ");
22             printf("\n");
23     }
24     getch();
25 }

```

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

 **Kết quả in ra màn hình**

! ! ! ! !!!! ! ! ! ! _	Bạn sửa lại chương trình để in ra chữ C, B...
---------------------------------------	---

8.2.1.15 Dùng mảng 1 chiều làm tham số cho hàm

Ví dụ 12 : Tìm số lớn nhất

Dòng	File	Edit	Search	Run	Compile	Debug	Project	Option	Window	Help
1	/* Chương trình tìm số lớn nhất sử dụng hàm */									
2										
3	#include <stdio.h>									
4	#include <conio.h>									
5										
6	#define MAX 20									
7										
8	//Khai báo prototype									
9	int max(int, int);									
10										
11	//hàm tìm số lớn nhất trong mảng 1 chiều									
12	int max(int ia[], int in)									
13	{									
14	int i, imax;									
15	imax = ia[0]; //cho phần tử đầu tiên là max									
16	for (i = 1; i < in; i++)									
17	if (imax < ia[i]) //nếu số đang xét > max									
18	imax = ia[i]; //gán số này cho max									
19	return imax; //trả về kết quả số lớn nhất									
20	}									
21										
22	void main(void)									

```

23 {
24   int ia[MAX];
25   int i = 0, inum;
26   do
27   {
28     printf("Nhap vao mot so: ");
29     scanf("%d", &ia[i]);
30   } while (ia[i++] != 0);
31   i--;
32   inum = max(ia, i);
33   printf("So lon nhat la: %d\n", inum);
34   getch();
35 }

```

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

☞ Kết quả in ra màn hình

Nhap vao mot so: 12 Nhap vao mot so: 45 Nhap vao mot so: 3 Nhap vao mot so: 0 So lon nhat la: 45 —	Chạy lại chương trình và thử lại với số liệu khác. Thực hiện một số thay đổi sau: - Di chuyển dòng int a[MAX]; lên sau dòng số 10 - Sửa dòng int max(int, int); thành int max(int); - Sửa dòng int max(int a[], int n); thành int max(int n); - Sửa dòng num = max(a, i); thành num = max(i); Chạy lại chương trình, quan sát, nhận xét và đánh giá kết quả.
---	---

☞ Giải thích chương trình

Chương trình ban đầu hàm max có hai tham số truyền vào và kết quả trả về là giá trị max có kiểu nguyên, một tham số là mảng 1 chiều kiểu int và một tham số có kiểu int. Với chương trình sau khi sửa hàm max chỉ còn một tham số truyền vào nhưng cho kết quả như nhau. Do sau khi sửa chương trình mảng a[MAX] được khai báo lại là biến toàn cục nên hàm max không cần truyền tham số mảng vào cũng có thể sử dụng được. Tuy vậy, khi lập trình bạn nên viết như chương trình ban đầu là truyền tham số mảng vào (dạng tổng quát) để hàm max có thể thực hiện được trên nhiều mảng khác nhau. Còn với chương trình sửa lại bạn chỉ sử dụng hàm max được với mảng a mà thôi.

Ví dụ 13 : Bạn khai báo các mảng sau ia[MAX], ib[MAX], ic[MAX]. Để tìm giá trị lớn nhất của từng mảng. Bạn chỉ cần gọi hàm

- imax_a = max(ia, i);
- imax_b = max(ib, i);
- imax_c = max(ic, i);

Với chương trình sửa lại bạn không thể tìm được số lớn nhất của mảng b và c.

☞ **Bạn lưu ý rằng khi truyền mảng sang hàm, không tạo bản sao mảng mới. Vì vậy mảng truyền sang hàm có dạng tham biến. Nghĩa là giá trị của các phần tử trong mảng sẽ bị ảnh hưởng nếu có sự thay đổi trên chúng.**

Ví dụ 14 : Tìm số lớn nhất của 3 mảng a, b, c

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	/* Chuong trinh tim so lon nhat su dung ham */
2	
3	#include <stdio.h>
4	#include <conio.h>
5	
6	#define MAX 20
7	

```

8 //Khai bao prototype
9 int max(int, int);
10 int input(int);
11
12 //ham tim phan tu lon nhat trong mang 1 chieu
13 int max(int ia[], int in)
14 {
15     int i, imax;
16     imax = ia[0];           //cho phan tu dau tien la max
17     for (i = 1; i < in; i++)
18         if (max < ia[i])    //neu so dang xet > max
19             max = ia[i];    //gan so nay cho max
20     return imax;           //tra ve ket qua so lon nhat
21 }
22
23 //ham nhap lieu vao mang 1 chieu
24 int input(int ia[])
25 {
26     int i = 0;
27     do
28     {
29         printf("Nhap vao mot so: ");
30         scanf("%d", &ia[i]);
31     } while (ia[i++] != 0);
32     i--;
33     return i;
34 }
35
36 void main(void)
37 {
38     int ia[MAX], ib[MAX], ic[MAX];
39     int inum1, inum2, inum3;
40     printf("Nhap lieu cho mang a\n");
41     inum1 = max(ia, input(ia));
42     printf("Nhap lieu cho mang b\n");
43     inum2 = max(ib, input(ib));
44     printf("Nhap lieu cho mang c\n");
45     inum3 = max(ic, input(ic));
46     printf("So lon nhat cua mang a: %d, b: %d, c: %d.\n", inum1, inum2, inum3);
47     getch();
48 }

```

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

Kết quả in ra màn hình

Nhap lieu cho mang a: Nhap vao mot so: 12 Nhap vao mot so: 45 Nhap vao mot so: 3 Nhap vao mot so: 0 Nhap lieu cho mang b: Nhap vao mot so: 5 Nhap vao mot so: 15 Nhap vao mot so: 0	Nhap lieu cho mang c: Nhap vao mot so: 1 Nhap vao mot so: 5 Nhap vao mot so: 4 Nhap vao mot so: 0 So lon nhat cua mang a: 45, b: 15, c: 5. —
---	--

Chạy lại chương trình và thử lại với số liệu khác.

Viết thêm hàm tìm số nhỏ nhất.

Giải thích chương trình

Hàm input có kiểu trả về là int thông qua biến i (cho biết số lượng phần tử đã nhập vào) và 1 tham số là mảng 1 chiều kiểu int. Dòng 41, 43, 45 lần lượt gọi hàm input với các tham số là mảng a, b, c. Khi hàm input thực hiện việc nhập liệu thì các phần tử trong mảng cũng được cập nhật theo.

8.2.1.16 Dùng mảng 2 chiều làm tham số cho hàm

Ví dụ 15: Nhập vào 2 ma trận vuông cấp n số thập phân. Cộng 2 ma trận này lưu vào ma trận thứ 3 và tìm số lớn nhất trên ma trận thứ 3.

Dòng	File	Edit	Search	Run	Compile	Debug	Project	Option	Window	Help
1	/* cong ma tran */									
2										
3	#include <stdio.h>									
4	#include <conio.h>									
5										
6	#define MAX 20									
7										
8	//Khai bao prototype									
9	void input(float);									
10	void output(float);									
11	void add(float, float, float);									
12	float max(float);									
13										
14	//khai bao bien toan cuc									
15	int in;									
16										
17	//ham tim so lon nhat trong mang 2 chieu									
18	float max(float fa[][MAX])									
19	{									
20	float fmax;									
21	fmax = fa[0][0]; //cho phan tu dau tien la max									
22	for (int i = 0; i < in; i++)									
23	for (int ij = 0; ij < in; ij++)									
24	if (fmax < fa[i][ij]) //neu so dang xet > max									
25	fmax = fa[i][ij]; //gan so nay cho fmax									
26	return fmax; //tra ve ket qua so lon nhat									
27	}									
28										
29	//ham nhap lieu mang 2 chieu									
30	void input(float fa[][MAX])									
31	{									
32	for (int i = 0; i < in; i++)									
33	for (int ij = 0; ij < in; ij++)									
34	{									
35	printf("Nhap vao ptu[%d][%d]: ", i, ij);									
36	scanf("%f", &fa[i][ij]);									
37	}									
38	}									
39										
40	//ham in mang 2 chieu ra man hinh									

```

41 void output(float fa[][MAX])
42 {
43     for (int i = 0; i < in; i++)
44     {
45         for (int ij = 0; ij < n; ij++)
46             printf("%5.2f", fa[i][ij]);
47         printf("\n");
48     }
49 }
50
51 //ham cong 2 mang 2 chieu
52 void add(float fa[][MAX], float fb[][MAX], float fc[][MAX])
53 {
54     for (int i = 0; i < in; i++)
55         for (int ij = 0; ij < n; ij++)
56             fc[i, ij] = fa[i, ij] + fb[i, ij];
57 }
58
59 void main(void)
60 {
61     float fa[MAX][MAX], fb[MAX][MAX], fc[MAX][MAX];
62     printf("Nhap vao cap ma tran: ");
63     scanf("%d", &in);
64     printf("Nhap lieu ma tran àn");
65     input(fa);
66     printf("Nhap lieu ma tran b\n");
67     input(fb);
68     printf("Nhap lieu ma tran òn");
69     input(fc);
70     add(fa, fb, fc);
71     printf("Ma tran a:\n");
72     output(fa);
73     printf("Ma tran b:\n");
74     output(fb);
75     printf("Ma tran c:\n");
76     output(fc);
77     printf("So lon nhat cua ma tran c la: %5.2f, max(fc));
78     getch();
79 }

```

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

Kết quả in ra màn hình

Nhap vao cap ma tran : 2	Ma tran a:
Nhap lieu ma tran a:	5.20 4.00
Nhapvao ptu[0][0] : 5.2	7.10 9.00
Nhap vao ptu[0][1] : 4	Ma tran b:
Nhap vao ptu[1][0] : 7.1	12.00 3.40
Nhap vao ptu[1][1] : 9	9.60 11.00
Nhap lieu ma tran b:	Ma tran c:
Nhap vao ptu[0][0] : 12	17.20 7.40
Nhap vao ptu[0][1] : 3.4	16.70 20.00
Nhap vao ptu[1][0] : 9.6	So lon nhat cua ma tran c la: 20.00
Nhap vao ptu[1][1] : 11	-

Chạy lại chương trình và thử lại với số liệu khác.

Viết thêm hàm tìm số nhỏ nhất.

Giải thích chương trình

Trong chương trình khai báo biến in toàn cục do biến này sử dụng trong suốt quá trình chạy chương trình. Tham số truyền vào hàm là mảng hai chiều dưới dạng `a[][MAX]` vì hàm không dành chỗ cho mảng, hàm chỉ cần biết số cột để tham khảo đến các phần tử.

Ví dụ 16 : Mảng 2 chiều được khai báo `int ia[3][3]`

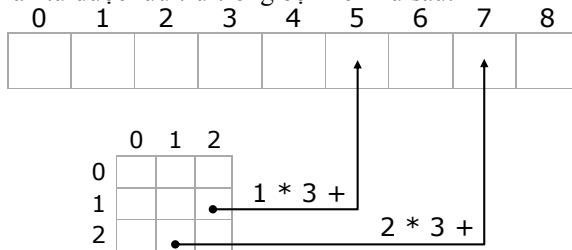
Truyền tham số vào hàm: `ia[][3]`,

để tham khảo đến phần tử `ptu[2][1]`,

hàm tính như sau:

$$2 * 3 + 1 = 7 \text{ (hỉ số hàng * số cột + chỉ số cột)}$$

`ia[3][3]` gồm 9 phần tử được lưu trữ trong bộ nhớ như sau:



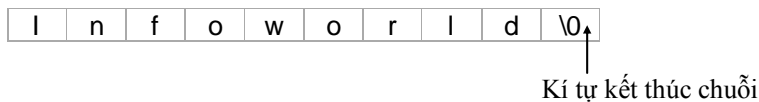
Giống như mảng 1 chiều khi truyền mảng 2 chiều sang hàm cũng không tạo bản sao mới.

8.2.2 Chuỗi

Chuỗi được xem như là một mảng 1 chiều gồm các phần tử có kiểu char như mẫu tự, con số và bất cứ ký tự đặc biệt như +, -, *, /, \$, #...

Theo quy ước, một chuỗi sẽ được kết thúc bởi ký tự **null** ('\0' : kí tự rỗng).

Ví dụ: chuỗi "Infoworld" được lưu trữ như sau:



8.2.2.1 Cách khai báo chuỗi


Ví dụ 17 : `char cname[30];`

Ý nghĩa: **Khai báo chuỗi cname có chiều dài 30 kí tự.** Do chuỗi kết thúc bằng kí tự null, nên khi bạn khai báo chuỗi có chiều dài 30 kí tự chỉ có thể chứa 29 kí tự.


Ví dụ 18 : Nhập vào in ra tên

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	<code>/* Chương trình nhập và in ra tên*/</code>
2	<code></code>
3	<code>#include <stdio.h></code>
4	<code>#include <conio.h></code>
5	<code></code>
6	<code>void main(void)</code>
7	<code>{</code>

8	char cname[30];
9	printf("Cho biet ten cua ban: ");
10	scanf("%s", cname);
11	printf("Chao ban %s", cname);
12	getch();
13	}
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu	

 **Kết quả in ra màn hình**

Cho biet ten cua ban: Minh Chao ban Minh _	Chạy lại chương trình và thử nhập tên: Mai Lan, Thanh Nhi Quan sát kết quả.
--	--


 **Lưu ý: không cần sử dụng toán tử địa chỉ & trong cname trong lệnh scanf("%s", fname), vì bản thân fname đã là địa chỉ.**

Dùng hàm scanf để nhập chuỗi có hạn chế như sau: Khi bạn thử lại chương trình trên với dữ liệu nhập vào là Mai Lan, nhưng khi in ra bạn chỉ nhận được Mai. Vì hàm scanf nhận vào dữ liệu đến khi gặp khoảng trắng thì kết thúc.


8.2.2.2 Hàm nhập (gets), xuất (puts) chuỗi

Sử dụng hàm gets, puts phải khai báo #include <stdio.h>

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	/* Chuong trinhnhap va in ra ten*/
2	
3	#include <stdio.h>
4	#include <conio.h>
5	
6	void main(void)
7	{
8	char cname[30];
9	puts("Cho biet ten cua ban: ");
10	gets(cname);
11	puts("Chao ban ");
12	puts(cname);
13	getch();
14	}
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu	


 **Kết quả in ra màn hình**

Cho biet ten cua ban: Mai Lan Chao ban Mai Lan _	Sửa dòng 9 thành printf("Cho biet ten cua ban: "); và từ dòng 11 đến 12 thành printf("Chao ban %s.\n", cname); Chạy lại chương trình vào thử nhập tên: Tuan Anh, Thanh Lan Quan sát kết quả.
--	--


 **Đối với hàm puts kí tự kết thúc chuỗi null (\0) được thay thế bằng kí tự newline (\n). Hàm gets và puts chỉ có 1 đối số và không sử dụng dạng thức trong nhập liệu cũng như xuất ra màn hình.**

8.2.2.3 Khởi tạo chuỗi

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	/* Chuong trinh nhap va in ra ten*/
2	
3	#include <stdio.h>
4	#include <conio.h>
5	
6	void main(void)
7	{
8	char cname[30];
9	char chao[] = "Chao ban";
10	printf("Cho biet ten cua ban: ");
11	gets(cname);
12	printf("%s %s\n", chao, cname);
13	getch();
14	}
	F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu


 **Kết quả in ra màn hình**

Cho biet ten cua ban: Mai Lan Chao ban Mai Lan _	Chạy lại chương trình vào thử nhập tên: Doan Trang Quan sát kết quả.
--	---

 **Chiều dài tối đa của chuỗi khởi tạo bằng số kí tự + 1 (kí tự null). Với chuỗi chao có chiều dài là 9.**

8.2.2.4 Mảng chuỗi

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	/* Chuong trinh nhap thang (so) và in ra thang (chu) tuong ung*/
2	
3	#include <stdio.h>
4	#include <conio.h>
5	
6	void main(void)
7	{
8	char cthang[12][15] = {"January", "February", "March", "April",
9	"May", "June", "July", "August", "September",
10	"October", "November", "December"};
11	int ithang;
12	printf("Nhap vao thang (1-12): ");
13	scanf("%d", &ithang);
14	printf("%s\n", cthang[ithang-1]);
15	getch();
16	}
	F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

 **Kết quả in ra màn hình**

Nhap vao thang (1-12): 2 February _	Chạy lại chương trình vào thử nhập vào các tháng khác Quan sát kết quả.
---	--

8.3 Bài tập

1. *Viết hàm tìm số lớn nhất, nhỏ nhất trong một mảng n số nguyên.*
2. *Viết hàm sắp xếp tăng dần, giảm dần của một dãy số cho trước.*
3. *Viết hàm tách tên và họ lót từ một chuỗi cho trước.*
4. *Viết hàm cắt bỏ khoảng trắng thừa ở giữa, hai đầu.*
5. *Viết hàm chuyển đổi 1 chuỗi sang chữ thường và 1 hàm chuyển đổi sang chữ HOA.*
6. *Viết hàm chuyển đổi 1 chuỗi sang dạng Title Case (kí tự đầu của mỗi từ là chữ HOA, các kí tự còn lại chữ thường)*
7. *Viết chương trình nhập vào 1 chuỗi và in ra chuỗi đảo ngược.*
Ví dụ: Nhập vào chuỗi "Lap trinh C can ban"
In ra "nab nac C hnirt paL"
8. *Viết chương trình nhập vào một chuỗi ký tự rồi đếm xem trong chuỗi đó có bao nhiêu chữ 'th'.*
9. *Biết rằng năm 0 là năm Canh thân (năm kỵ nhau có chu kì là 3, năm hợp nhau có chu kì là 4). Hãy viết chương trình cho phép gõ vào năm dương lịch (ví dụ 1997), xuất ra năm âm lịch (Đinh Sửu) và các năm kỵ và hợp.*
Có 12 chi: Tý, Sửu, Dần, Mão, Thìn, Tỵ, Ngọ, Mùi, Thân, Dậu, Tuất, Hợi.
Có 10 can: Giáp, Ất, Bính, Đinh, Mậu, Kỷ, Canh, Tân, Nhâm, Quý.
10. *Viết chương trình nhập vào 3 chữ số (305, 6, 28). Cho biết dòng chữ mô tả giá trị con số đó. Ví dụ 305 -> ba trăm lẻ năm.*
11. *Viết chương trình nhập vào một chuỗi sau đó in ra màn hình mỗi dòng là một từ. Ví dụ chuỗi "Lap trinh C". Kết quả in ra*
Lap
trinh
C
12. *Viết chương trình nhập vào một chuỗi các kí tự, ký số, khoảng trắng và dấu chấm câu. Cho biết chuỗi trên gồm bao nhiêu từ.*
13. *Viết chương trình nhập vào một chuỗi ký tự. Kiểm tra xem chuỗi đó có đối xứng không?*
14. *Viết chương trình nhập vào một chuỗi gồm các chữ cái (a -> z, A -> Z). Hãy đếm xem có bao nhiêu nguyên âm a, i, e, o, u.*
15. *Giả sử số phòng trong một khách sạn được cho bởi hằng số NUM_ROOM. Viết:*
 - a. Một khai báo dãy thích hợp để theo dõi phòng nào còn trống.
 - b. Một hàm tìm phòng nào còn trống.
 - c. Viết chương trình đơn giản để quản lý phòng khách sạn theo dạng một trình đơn chọn công việc gồm có 4 mục như sau:
 - Tìm phòng trống.
 - Trả phòng.
 - Liệt kê những phòng còn trống.
 - Liệt kê những phòng đã thuê.
 - Kết thúc.

16. *Viết chương trình mô tả văn bản của một bức điện tín. Nhập liệu bao gồm 1 hay nhiều dòng chứa một số từ, mỗi từ cách nhau khoảng trắng. In ra hóa đơn tính tiền với mỗi từ giá 100 đồng, phí trả thêm 50 đồng cho từ dài quá 8 kí tự. Hóa đơn có dạng sau:*

So tu	:	10
So tu có kích thước bình thường	:	$8 \times 100 = 800$ đồng
So tu có kích thước > 8 kí tự	:	$2 \times 150 = 300$ đồng
Tổng cộng	:	1100 đồng

17. *Viết chương trình kiểm tra xem có bao nhiêu người họ "Ly", "Tran"... trong 1 danh sách cho trước. Nếu không có thông báo "Không có người nào thuộc họ".*

18. *Viết chương trình nhập vào 1 chuỗi, sau đó chép sang chuỗi khác một chuỗi con từ chuỗi ban đầu có số kí tự chỉ định.*

Ví dụ: Chuỗi ban đầu "Le Thuy Doan Trang". Nếu số kí tự chỉ định là 2 thì chuỗi đích sẽ là "Le"

19. *Viết chương trình nhập vào 1 chuỗi, sau đó chép sang chuỗi khác một chuỗi con từ chuỗi ban đầu với vị trí bắt đầu và số kí tự chỉ định.*

Ví dụ: Chuỗi ban đầu "Le Thuy Doan Trang". Nếu vị trí ban đầu là 14 và số kí tự chỉ định là 5 thì chuỗi đích sẽ là "Trang"

20. *Viết chương trình nhập vào 1 chuỗi nguồn, ví dụ "Nguyen Minh Long", sau đó nhập vào 1 chuỗi con, ví dụ "Minh", chương trình sẽ xác định vị trí bắt đầu của chuỗi con ở vị trí nào trong chuỗi nguồn. Kết quả in ra màn hình như sau:*

- Chuỗi nguồn là : Nguyen Minh Long
- Chuỗi con là : Minh
- Vị trí bắt đầu của chuỗi con là : 8

21. *Viết chương trình thực hiện các yêu cầu sau:*

- Nhập vào 1 chuỗi bất kỳ, ví dụ : "Nguyen Minh Long
- Muốn xóa từ vị trí nào, ví dụ : 8
- Muốn xóa bao nhiêu kí tự, ví dụ : 5

Kết quả in ra màn hình:

- Chuỗi nguồn là : Nguyen Minh Long
- Chuỗi sau khi xóa : Nguyen Long



Bài 9 : CON TRỎ

9.1 Mục tiêu

Sau khi hoàn tất bài này học viên sẽ hiểu và vận dụng các kiến thức kỹ năng cơ bản sau:

- Ý nghĩa, cách khai báo con trỏ
- Sử dụng con trỏ trong mảng, chuỗi
- Truyền mảng và chuỗi giữa các hàm qua con trỏ
- Xử lý mảng dễ dàng qua con trỏ

9.2 Nội dung

9.2.1 Con trỏ?

Con trỏ dùng để truy cập biến thông qua địa chỉ biến và chương trình tham khảo biến gián tiếp qua địa chỉ này.

9.2.2 Khái báo biến con trỏ

Ví dụ 1:

Dòng	File	Edit	Search	Run	Compile	Debug	Project	Option	Window	Help
1	/* Cong hang so */									
2										
3	#include <stdio.h>									
4	#include <conio.h>									
5										
6	void main(void)									
7	{									
8	int ix = 6, iy = 7;									
9	int *px, *py;									
10	printf("x = %d, y = %d\n", ix, iy);									
11	px = &ix;									
12	py = &iy;									
13	*px += 10;									
14	*py += 10;									
15	printf("x = %d, y = %d\n", ix, iy);									
16	getch();									
17	}									
	F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu									

Kết quả in ra màn hình

x = 6, y = 7 x = 16, y = 17 —	Chạy chương trình và quan sát kết quả.
-------------------------------------	--

Giải thích chương trình

Khai báo ở dòng 9 cấp phát 2 bytes để lưu giữ địa chỉ của biến nguyên và vùng nhớ đó có tên là px, tương tự cho py. Dấu * cho biết biến này chứa địa chỉ chứ không phải giá trị, int cho biết địa chỉ đó sẽ trỏ đến các biến nguyên.

☞ Ví dụ trên cho thấy *px và *py là 2 biến con trỏ trỏ đến địa chỉ của 2 biến ix và iy (dòng 11 và 12), vì vậy khi nội dung của biến con trỏ *px và *py thay đổi thì nội dung của ix, iy cũng thay đổi theo.

Địa chỉ vùng nhớ

1203	6	ix
1204		
1205		
1206		
1207	7	iy
1208		
1209		
1210		

Sau phép gán px = &ix và py = &iy thì giá trị của px = 1203 và py = 1207

Địa chỉ vùng nhớ


2015		
2016		
2017	1203	px
2018		
2019	1207	py
2020		
2021		
2022		

*px là nội dung của px và *py là nội dung của py. Nên khi thực hiện 2 phép cộng gán *px += 10 và *py += 10 thì giá trị của ix sẽ là 16 và giá trị iy sẽ là 17.


9.2.3 Truyền địa chỉ sang hàm

Ví dụ 2:

Dòng	File	Edit	Search	Run	Compile	Debug	Project	Option	Window	Help	
1	/* Khoi tao 2 so */										
2											
3	#include <stdio.h>										
4	#include <conio.h>										
5											
6	void init (int, int);										
7											
8	void main(void)										
9	{										
10	int ix, iy;										
11	init(&ix, &iy);										
12	printf("x = %d, y = %d\n", ix, iy);										
13	getch();										
14	}										
15											
16	void init(int *px, int *py)										
17	{										
18	*px = 3; //gan 3 cho noi dung cua px										
19	*py = 5; //gan 5 cho ai dung cua py										
20	}										
	F1 Help	Alt-F8	Next Msg	Alt-F7	Prev Msg	Alt - F9	Compile	F9	Make	F10	Menu

 **Kết quả in ra màn hình**

x = 3, y = 5 —	Chạy chương trình và quan sát kết quả.
-------------------	--

 **Giải thích chương trình**

Ở dòng 9, gọi hàm `init` truyền 2 tham số là địa chỉ của biến `ix` và `iy`, nên khi nội dung của 2 biến con trỏ `*px` và `*py` thay đổi thì `ix` và `iy` của chương trình chính cũng thay đổi theo. Hàm `main(void)` đã sử dụng cách truy cập biến khác với hàm `init`, hàm `main(void)` gọi chúng là `ix`, `iy` còn hàm `init` gọi chúng là `*px`, `*py`. Hàm `init` đọc giá trị của biến con trỏ `*px`, `*py` từ vùng địa chỉ của chương trình gọi, sau khi thực hiện và trả kết quả về chương trình gọi.

9.2.4 Con trỏ và mảng


Ví dụ 3: Khai báo mảng sau `int num[] = {23, 54, 16, 72, 16, 84};`

Như đã nghiên cứu cách tham chiếu đến phần tử mảng thứ 5 là `num[4]`, còn với kiểu con trỏ là `*(num + 4)`. Nghĩa là `num[4]` tương đương với `*(num + 4)` và cách truy cập nội dung như nhau. Tương tự như vậy, cách tham khảo địa chỉ của phần tử mảng là `&num[4]` tương đương với `num + 4`.

9.2.5 Con trỏ trỏ đến mảng trong hàm

Ví dụ 4:

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	<code>/* Cong hang so vao mang */</code>
2	
3	<code>#include <stdio.h></code>
4	<code>#include <conio.h></code>
5	
6	<code>#define SIZE 4</code>
7	
8	<code>add(int *, int, int);</code>
9	
10	<code>void main(void)</code>
11	<code>{</code>
12	<code> int iarray[] = {2, 5, 6, 9};</code>
13	<code> int i, ix = 10;</code>
14	<code> add(iarray, SIZE, ix);</code>
15	<code> for (i = 0; i < SIZE; i++);</code>
16	<code> printf("%d ", *(iarray + i));</code>
17	<code> getch();</code>
18	<code>}</code>
19	
20	<code>void add(int *ptr, int inum, int ia)</code>
21	<code>{</code>
22	<code> int ij;</code>
23	<code> for (ij = 0; ij < inum; ij++)</code>
24	<code> *(ptr) = *(ptr++) + ia;</code>
25	<code>}</code>
	F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

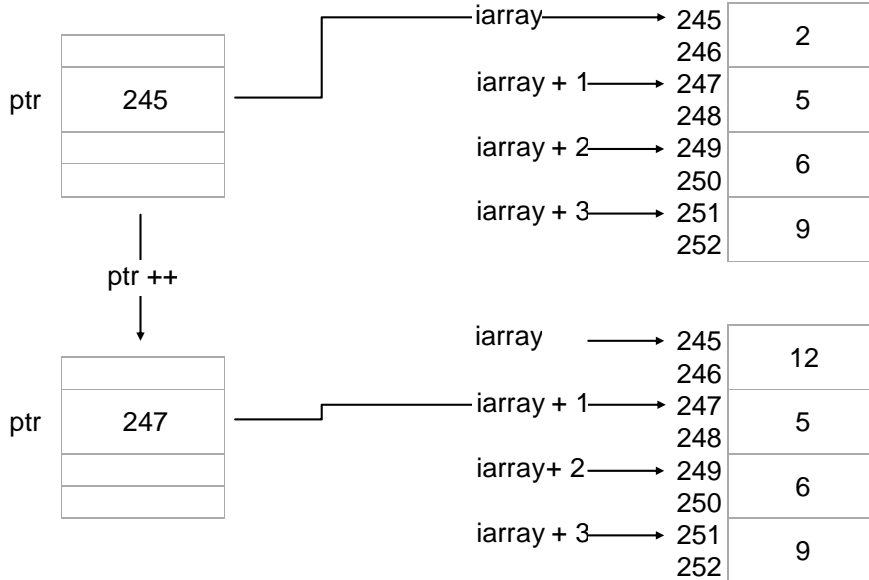
 **Kết quả in ra màn hình**

12 15 16 19 —	Chạy chương trình và quan sát kết quả.
------------------	--

Giải thích chương trình

Hàm gán địa chỉ của mảng vào biến con trỏ ptr, kích thước vào biến inum và hằng số vào biến ia. Sau đó dùng vòng lặp để cộng hằng vào từng phần tử của mảng.

Giả sử địa chỉ của ia là 245 khi truyền vào hàm add qua ptr, ptr sẽ có giá trị = 245



9.2.6 Con trỏ và chuỗi

Ví dụ 5:

```

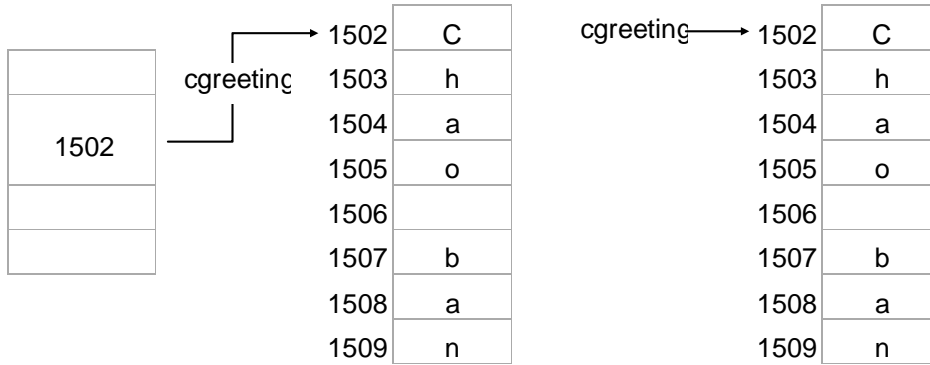
Dòng  File  Edit  Search  Run  Compile  Debug  Project  Option  Window  Help
1  /* Chuong trinh nhap va in ra ten*/
2
3  #include <stdio.h>
4  #include <conio.h>
5
6  void main(void)
7  {
8      char *cgreeting = "Chao ban";
9      char cname[30];
10     puts("Cho biet ten cua ban: ");
11     gets(cname);
12     puts(cgreeting);
13     puts(cname);
14     getch();
15 }
F1 Help  Alt-F8 Next Msg  Alt-F7 Prev Msg  Alt - F9 Compile  F9 Make  F10 Menu
    
```

Kết quả in ra màn hình

Cho biet ten cua ban: Minh Chao ban Minh _	Chạy chương trình và quan sát kết quả.
--	--

Giải thích chương trình

cgreeting là biến con trỏ được khởi tạo bằng phát biểu char *cgreeting = "Chao ban" thay vì char cgreeting[] = "Chao ban". Cả hai cách đều cho cùng kết quả và đều dành số byte cho chuỗi kèm theo kí tự null. Đối với mảng địa chỉ của kí tự đầu tiên của mảng sẽ là tên mảng, còn con trỏ sẽ có thêm biến con trỏ trỏ đến tên cgreeting.



char *cgreeting = "Chao ban"
(biến con trỏ)

char cgreeting[] = "Chao ban"
(hằng con trỏ)

9.2.7 Khởi tạo mảng con trỏ trỏ đến chuỗi

Vi dụ 6:

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	/* Chuong tinh nhap thang (so) va in ra thang (chu) tuong ung*/
2	
3	#include <stdio.h>
4	#include <conio.h>
5	
6	void main(void)
7	{
8	char *cthang[12]={"January", "February", "March", "April",
9	"May", "June", "July", "August", "September",
10	"October", "November", "December"};
11	int ithang;
12	printf("Nhap vao thang (-12): ");
13	scanf("%d", &ithang);
14	printf("%s\n", cthang[ithang]);
15	getch();
16	}
	F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

Kết quả in ra màn hình

Nhập vào thang (-12): 2 February _	Chạy lại chương trình vào thử nhập vào các tháng khác Quan sát kết quả.
--	--

Giải thích chương trình

Khai báo char *cthang[12] có ý nghĩa như sau: cthang là tên gọi, dấu * là kiểu con trỏ trỏ đến kí tự (char).

	Địa chỉ	0	1	2	3	4	5	6	7	8	9
cthang[0]	→ 1010	J	a	n	u	a	r	y	\0		
cthang[1]	→ 1020	F	e	b	r	u	a	r	y	\0	
cthang[2]	→ 1030	M	a	r	c	h	\0				
cthang[3]	→ 1040	A	p	r	i	l	\0				
cthang[4]	→ 1050	M	a	y	\0						
cthang[5]	→ 1060	J	u	n	e	\0					
cthang[6]	→ 1070	J	u	l	y	\0					
cthang[7]	→ 1080	A	u	g	u	s	t	\0			
cthang[8]	→ 1090	S	e	p	t	e	m	b	e	r	\0
cthang[9]	→ 1100	O	c	t	o	b	e	r	\0		
cthang[10]	→ 1110	N	o	v	e	m	b	e	r	\0	
cthang[11]	→ 1120	D	e	c	e	m	b	e	r	\0	

Mảng các chuỗi char cthang[12][10]

cthang[0]	1010	→ 1010	J	a	n	u	a	r	y	\0	
cthang[1]	1018	→ 1018	F	e	b	r	u	a	r	y \0	
cthang[2]	1027	→ 1027	M	a	r	c	h	\0			
cthang[3]	1033	→ 1033	A	p	r	i	l	\0			
cthang[4]	1039	→ 1039	M	a	y	\0					
cthang[5]	1043	→ 1043	J	u	n	e	\0				
cthang[6]	1048	→ 1048	J	u	l	y	\0				
cthang[7]	1053	→ 1053	A	u	g	u	s	t	\0		
cthang[8]	1060	→ 1060	S	e	p	t	e	m	b	e	r \0
cthang[9]	1070	→ 1070	O	c	t	o	b	e	r	\0	
cthang[10]	1078	→ 1078	N	o	v	e	m	b	e	r	\0
cthang[11]	1087	→ 1087	D	e	c	e	m	b	e	r	\0

Mảng các con trỏ trỏ đến các chuỗi char *cthang[12]

Khởi tạo mảng các con trỏ trỏ đến các chuỗi chiếm ít bộ nhớ hơn khởi tạo mảng chuỗi.

9.2.8 Xử lý con trỏ trỏ đến chuỗi

Ví dụ 7:

Dòng	File	Edit	Search	Run	Compile	Debug	Project	Option	Window	Help
1	/* Nhập danh sách tên và sắp xếp theo thu nhập tăng dần*/									
2										
3	#include <stdio.h>									
4	#include <conio.h>									
5	#include <string.h>									
6										
7	#define MAXNUM 5									

```

8 #define MAXLEN 10
9
10 void main(void)
11 {
12     char cname[MAXNUM][MAXLEN]; //mang chuoai
13     char *cptr[MAXNUM];         //mang con tro tro den chuoai
14     char *ctemp;
15     int i, ij, icount = 0;
16
17     //nhap danh sach ten
18     while (icount < MAXNUM)
19     {
20         printf("Nhap vao ten nguoi thu %d: ", icount + 1);
21         gets(cname[icount]);
22         cptr[icount++] = cname[icount]; //con tro den ten
23     }
24
25     //sap xep danh sach theo thu tu tang dan
26     for (i = 0; i < icount- 1; i ++)
27         for (ij = i + 1; ij < icount; ij ++)
28             if (strcmp(cptr[i], cptr[ij]) > 0)
29             {
30                 ctemp = cptr[i];
31                 cptr[i] = cptr[ij];
32                 cptr[ij] = ctemp;
33             }
34
35     //In danh sach da sap xep
36     printf("Danh sach sau khi sap xep");
37     for (i = 0; i < icount; i ++)
38         printf("Ten nguoi thu %d : %s", i + 1, cptr[i]);
39     getch();
40 }

```

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

Kết quả in ra màn hình

Nhap vao ten nguoi thu 1: Minh
 Nhap vao ten nguoi thu 2: Lan
 Nhap vao ten nguoi thu 3: Anh
 Nhap vao ten nguoi thu 4: Trang
 Nhap vao ten nguoi thu 5: Quan
 Danh sach sau khi sap xep:
 Ten nguoi thu 1: Anh
 Ten nguoi thu 2: Lan
 Ten nguoi thu 3: Minh
 Ten nguoi thu 4: Quan
 Ten nguoi thu 5: Trang

Chạy lại chương trình và thử nhập với dữ liệu khác.

Quan sát kết quả.

Giải thích chương trình

Trong chương trình dùng cả mảng chuỗi char `cname[MAXNUM][MAXLEN]` và mảng con trỏ trỏ đến chuỗi char `*cptr[MAXNUM]`;

<code>cptr[0]</code>	1010	→	1010	M	i	n	h	\0	
<code>cptr[1]</code>	1016	→	1016	L	a	n	\0		
<code>cptr[2]</code>	1022	→	1022	A	n	h	\0		
<code>cptr[3]</code>	1028	→	1028	T	r	a	n	g	\0
<code>cptr[4]</code>	1034	→	1034	Q	u	a	n	\0	

↑ Mảng các con trỏ trỏ đến chuỗi trước khi sắp xếp

<code>cptr[0]</code>	1022	→	1010	M	i	n	h	\0	
<code>cptr[1]</code>	1016	→	1016	L	a	n	\0		
<code>cptr[2]</code>	1010	→	1022	A	n	h	\0		
<code>cptr[3]</code>	1034	→	1028	T	r	a	n	g	\0
<code>cptr[4]</code>	1028	→	1034	Q	u	a	n	\0	

↑ Mảng các con trỏ trỏ đến chuỗi sau khi sắp xếp

9.2.9 Con trỏ trỏ đến con trỏ

Ví dụ 8:

```

Dòng  File  Edit  Search  Run  Compile  Debug  Project  Option  Window  Help
1  /* In ma trận*/
2
3  #include <stdio.h>
4  #include <conio.h>
5
6  #define ROWS 4
7  #define COLS 5
8
9  void main(void)
10 {
11     int itable[ROWS][COLS] = {{10, 12, 14, 16, 18},
12                               {11, 13, 15, 17, 19},
13                               {20, 22, 24, 26, 28},
14                               {21, 23, 25, 27, 29}};
15     int i, ij, ix = 10;
16
17     for (i = 0; i < ROWS; i++)
18         for (ij = 0; ij < COLS; ij++)
19             (*(table + i) + ij) += ix;
20
21     for (i = 0; i < ROWS; i++)
22     {
23         for (ij = 0; ij < COLS; ij++)
24             printf("%4d", *(table + i) + ij);
25         printf("\n");
26     }
27     getch();
28 }
F1 Help  Alt-F8 Next Msg  Alt-F7 Prev Msg  Alt - F9 Compile  F9 Make  F10 Menu
    
```

☞ Kết quả in ra màn hình

20 22 24 26 28 21 23 25 27 29 30 32 34 36 38 31 33 35 37 39 _	Chạy chương trình và quan sát kết quả.
---	--

☞ Giải thích chương trình

Trong chương trình dùng cả mảng chuỗi char `cname[MAXNUM][MAXLEN]` và mảng con trỏ đến chuỗi char `*cptr[MAXNUM]`;

9.3 Bài tập

Làm lại các bài tập ở bài Mảng và chuỗi sử dụng biến con trỏ.



☞ Đối với biến khai báo kiểu con trỏ **nhanvien *nv** thì tham chiếu đến phần tử **manv**:
nv -> manv.

Ví dụ 3: Nhập và in danh sách nhân viên.

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	<code>/* Danh sach nhan vien */</code>
2	
3	<code>#include <stdio.h></code>
4	<code>#include <conio.h></code>
5	<code>#include <stdlib.h></code>
6	
7	<code>#define MAX 50</code>
8	
9	<code>void main(void)</code>
10	<code>{</code>
11	<code> struct nhanvien</code>
12	<code> {</code>
13	<code> int manv;</code>
14	<code> char hoten[30];</code>
15	<code> };</code>
16	<code> nhanvien snv[MAX];</code>
17	<code> char ctam[10];</code>
18	<code> int i, in;</code>
19	<code> printf("Nhap vao so nhan vien: ");</code>
20	<code> gets(ctam);</code>
21	<code> in = atoi(ctam);</code>
22	
23	<code> //Nhap danh sach nhan vien</code>
24	<code> for(i = 0; i < in; i++)</code>
25	<code> {</code>
26	<code> printf("Nhap vao ma nhan vien thu %d: ", i + 1);</code>
27	<code> gets(ctam);</code>
28	<code> snv[i].manv = atoi(ctam);</code>
29	<code> printf("Nhap vao ho ten: ");</code>
30	<code> gets(snv[i].hoten);</code>
31	<code> }</code>
32	
33	<code> //in danh sach nhan vien</code>
34	<code> for(i = 0; i < in; i++)</code>
35	<code> printf("%5d %s\n", snv[i].manv, snv[i].hoten);</code>
36	<code> getch();</code>
37	<code>}</code>
	F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

☞ **Kết quả in ra màn hình**

Nhap vao so nhan vien: 2	Chạy và thử lại chương trình với dữ liệu khác.
Nhap vao ma nhan vien thu 1: 123	Quan sát kết quả.
Nhap vao ho ten: Le Thuy Doan Tran	
Nhap vao ma nhan vien thu 2: 35	

Nhập vào họ tên: Le Nguyen Tuan An 123 Le Thuy Doan Trang 35 Le Nguyen Tuan Anh —
--

☞ Trong chương trình trên dùng tổ hợp 2 dòng 20 và 21 gồm 2 lệnh gets, atoi để nhập một số nguyên tránh lỗi do scanf và vùng đệm bàn phím gây ra.

10.2.1.4 Khởi tạo structure

Ví dụ 4: Nhập vào bảng số xe, cho biết xe đó đăng kí ở tỉnh nào.

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	/* Xac dinh bien so xe */
2	
3	#include <stdio.h>
4	#include <conio.h>
5	#include <stdlib.h>
6	
7	#define MAX 6
8	
9	void main(void)
10	{
11	struct tinh
12	{
13	int ma;
14	char *ten;
15	};
16	tinh sds[MAX] = {{60, "Dong Nai"}, {61, "Binh Duong"}, {62, "Long An"},
17	{63, "Tien Giang"}, {64, "Vinh Long"}, {65, "Can Tho"}};
18	char ctam[10];
19	int i, in;
20	printf("Nhap vào bien số xe");
21	gets(ctam);
22	in = atoi(ctam);
23	
24	for(i = 0; i < MAX; i++)
25	if (sds[i].ma == in)
26	printf("Xe đăng kí ở tỉnh %s", sds[i].ten);
27	getch();
28	}
	F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

☞ Kết quả in ra màn hình

Nhập vào biến số xe: 62 Xe đăng kí ở tỉnh Long An —	Chạy và thử lại chương trình với 65H5-1246, 60F47712, 64F44542 Quan sát kết quả.
---	--

☞ Dòng 22 đổi chuỗi sang số nguyên, ở ví dụ trên sau khi dòng này thực hiện giá trị của in = 62.

10.2.1.5 Structure lồng nhau

Ví dụ 5: Nhập và in danh sách nhân viên.

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	/* Danh sach nhan vien */
2	
3	#include <stdio.h>
4	#include <conio.h>
5	#include <stdlib.h>
6	
7	#define MAX 50
8	
9	void main(void)
10	{
11	struct giacanh
12	{
13	char vo_chong[30];
14	char con;
15	};
16	
17	struct nhanvien
18	{
19	int manv;
20	char hoten[0];
21	giacanh canhan;
22	};
23	nhanvien snv[MAX];
24	char ctam[10];
25	int i, in;
26	printf("Nhap vao so nhan vien: ");
27	gets(ctam);
28	in = atoi(ctam);
29	
30	//Nhap danh sach nhan vien
31	for(i = 0; i < in; i++)
32	{
33	printf("Nhap vao ma nhan vien thu %d: ", i + 1);
34	gets(ctam);
35	snv[i].manv = atoi(ctam);
36	printf("Nhap vao ho ten: ");
37	gets(snv[i].hoten);
38	printf("Cho biet ten vo (hoac chong): ");
39	gets(snv[i].canhan.vo_chong);
40	printf("So con: ");
41	gests(ctam);
42	}
43	
44	//in danh sach nhan vien


```

45     for(i = 0; i < in; i++)
46     {
47         printf("Ma so: %d\nHo ten: %s\n Ho ten vo (hoac chong): %s\n So con: %d",
48             snv[i].manv, snv[i].hoten, snv[i].canhan.vo_chong, snv[i].canhan.con);
49         getch();
50 }#include <stdio.h>
#include <conio.h>
#include <stdlib.h>

#define MAX 50

void main(void)
{
    struct giacanh
    {
        char vo_chong[30];
        char con;
    };


    struct nhanvien
    {
        int manv;
        char hoten[30];
        giacanh canhan;
    };
    nhanvien snv[MAX];
    char ctam[10];
    int i, in;
    printf("Nhap vao so nhan vien: ");
    gets(ctam);
    in = atoi(ctam);

    //Nhap danh sach nhan vien
    for(i = 0; i < in; i++)
    {
        printf("Nhap vao ma nhan vien thu %d: ", i + 1);
        gets(ctam);
        snv[i].manv = atoi(ctam);
        printf("Nhap vao ho ten: ");
        gets(snv[i].hoten);
        printf("Cho biet ten vo (hoac chong): ");
        gets(snv[i].canhan.vo_chong);
        printf("So con: ");
        gets(ctam);
    }


    //in danh sach nhan vien
    for(i = 0; i < in; i++)

```

	<pre>{ printf("Ma so: %d\nHo ten: %s\n Ho ten vo (hoac chong): %s\n So con: %d", snv[i].manv, snv[i].hoten, snv[i].canhan.vo_chong, snv[i].canhon); getch(); }</pre>
	F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

 **Kết quả in ra màn hình**

Nhập vào số nhân viên: 3 Nhập vào mã nhân viên thu 1: 123 Nhập vào họ tên: Le Thuy Doan Trang Nhập vào mã nhân viên thu 2: 35 Nhập vào họ tên: Le Nguyen Tuan Anh 123 Le Thuy Doan Trang 35 Le Nguyen Tuan Anh _	Chạy và thử lại chương trình với dữ liệu khác. Quan sát kết quả.
---	---

 Trong chương trình trên dùng tổ hợp 2 dòng 20 và 21 gồm 2 lệnh gets, atoi để nhập một số nguyên tránh lỗi do scanf và vùng đệm bàn phím gây ra.

10.2.1.6 Truyền structure sang hàm

Giống như mảng, bạn có thể truyền vào hàm qua tham biến.

Ví dụ 6: Sửa lại ví dụ 3, sử dụng hàm cho nhập và in danh sách

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	/* Danh sach nhan vien */
2	
3	#include <stdio.h>
4	#include <conio.h>
5	#include<stdlib.h>
6	
7	#define MAX 50
8	
9	//Khai bao structure toan cuc
10	struct nhanvien
11	{
12	int manv;
13	char hoten[30];
14	};
15	
16	//Khai bao prototype
17	void input(nhanvien, int);
18	void output(nhanvien, int);
19	
20	//Ham nhap danh sach
21	void input(nhanvien snv[], int in)
22	{
23	char ctam[10];

```

24   for(int i = 0; i < in; i++)
25   {
26       printf("Nhap vao ma nhan vien thu %d: ", i + 1);
27       gets(ctam);
28       snv[i].manv = atoi(ctam);
29       printf("Nhap vao ho ten: ");
30       gets(snv[i].hoten);
31   }
32 }
33
34 //Ham in danh sach ra man hinh
35 void output(nhanvien snv[], int in)
36 {
37     for(i = 0; i < in; i++)
38         printf("%5d %s\n", snv[i].manv, snv[i].hoten);
39 }
40
41 void main(void)
42 {
43     nhanvien snv[MAX];
44     char ctam[10];
45     int i, in;
46     printf("Nhap vao so nhan vien: ");
47     gets(ctam);
48     in = atoi(ctam);
49     input(snv, in);
50     output(snv, in);
51     getch();
52 }

```


F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

Kết quả in ra màn hình

<p>Nhap vao so nhan vien: 3 Nhap vao ma nhan vien thu 1: 123 Nhap vao ho ten: Le Thuy Doan Tran Nhap vao ma nhan vien thu 2: 35 Nhap vao ho ten: Le Nguyen Tuan An 123 Le Thuy Doan Trang 35 Le Nguyen Tuan Anh _</p>	<p>Chạy và thử lại chương trình với dữ liệu khác. Quan sát kết quả.</p>
--	--

Giải thích chương trình

Ở chương trình này ta phải khai báo struct nhanvien là biến toàn cục, vì khi định nghĩa hàm input và output có sử dụng kiểu dữ liệu struct nhanvien.

 **Bạn lưu ý rằng khi truyền struct sang hàm, không tạo bản sao mới. Vì vậy struct truyền sang hàm có dạng tham biến. Nghĩa là giá trị của các phần tử trong struct sẽ bị ảnh hưởng nếu có sự thay đổi trên chúng.**

Ví dụ 7: Sửa lại ví dụ 6, từ dòng 20 đến dòng 32 như sau:

```
//Ham nhap tung nhan vien
nhanvien newnv()
{
    nhanvien snv;
    printf("Ma nhan vien: ");
    gets(ctam);
    snv.manv = atoi(ctam);
    printf("Ho ten: ");
    gets(snv.hoten);
    return (snv);
}

//Ham nhap danh sach nhan vien
void input(nhanvien snv[], int in)
{
    for(int i = 0; i < in; i++)
    {
        printf("Nhap vao nhan vien thu %d: ", i + 1);
        snv[i] = newnv();
    }
}
```

☞ Hàm newnv có kiểu trả về là struct nhanvien

10.2.2 Enum

Một biến là kiểu dữ liệu enum có thể nhận được một giá trị nào đó trong các giá trị được liệt kê.

10.2.2.1 Định nghĩa kiểu enum

Ví dụ 8: định nghĩa kiểu enum day

từ khóa ← tên → dấu ; kết thúc enum

```
enum day{ SUN, MON, TUE, WED, THU, FRI, SAT };
```

các giá trị liệt kê

các giá trị được bọc trong móc

Mỗi tên thứ (SUN, MON ... SAT) trong day sẽ được đánh số lần lượt từ 0 đến 6 (SUN là 1, MON là 2... SAT là 6). Nếu bạn muốn bắt đầu bằng giá trị khác thì gán giá trị mong muốn vào và trị kế tiếp sẽ tăng lên 1.

☞ enum phải viết bằng chữ thường

10.2.2.2 Cách khai báo biến có kiểu enum

Ví dụ 9: enum day ngay; hoặc day ngay;

Khai báo biến ngay có kiểu enum day.

☞ vừa tạo enum day vừa khai báo biến ngay

```
enum day{ SUN, MON, TUE, WED, THU, FRI, SAT} ngay;
```

10.2.2.3 Sử dụng enum trong chương trình

Ví dụ 10: Tính tiền lương tuần cho nhân viên. Thứ bảy và Chủ nhật được tính phụ trội

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	/* Tinh tien luong tuan cho nhan vien */
2	
3	#include <stdio.h>
4	#include <conio.h>
5	
6	#define PHU_TROI_T7 1.5
7	#define PHU_TROI_CN 2.0
8	
9	//dinh nghĩa enum
10	enum tuan {CHU_NHAT, THU_HAI, THU_BA, THU_TU, THU_NAM, THU_SAU, THU_BAY};
11	typedef enum tuan ngay_tuan; //dinh nghĩa ngay_tuan là tuan
12	
13	void main(void)
14	{
15	int igio;
16	float fLuongCB, fLuongNgay, fTongLuong;
17	char cngay[][4] = {"Chu Nhat", "Thu Hai", "Thu Ba", "Thu Tu", "Thu Nam", "Thu
18	Sau", "Thu Bay"};
19	ngay_tuan engay;
20	ngay_tuan ngay_mai(ngay_tuan); //khai bao prototype
21	
22	printf("Nhap vao luong can ban: ");
23	scanf("%f", &fLuongCB);
24	
25	luong = 0.0;
26	printf("Nhap vao so gio lam viec tu hai den Chu nhat");
27	engay = CHU_NHAT;
28	do
29	{
30	engay = ngay_mai(engay);
31	printf("Nhap vao gio lam viec ngay %s :", cngay[engay]);
32	scanf("%d", &igio);
33	swith(engay)
34	{
35	case THU_HAI: case THU_BA: case THU_TU: case THU_NAM: case THU_SAU:
36	fLuongNgay = fLuongCB;
37	break;
38	case THU_BAY:
39	fLuongNgay = fLuongCB * PHU_TROI_T7;
40	break;
41	case CHU_NHAT:
42	fLuongNgay = fLuongCB * PHU_TROI_CN;
43	break;
44	}
45	fTongLuong += fLuongNgay * igio;
46	} while (ngay != CHU_NHAT);

```

47 printf("Tong luong tuan = %8.2f dong\n", fTongLuong);
48 getch();
49 }
50
51 //ham chon ngay ke tiep
52 ngay_tuan ngay_mai(ngay_tuan en)
53 {
54     ngay_tuan engay_ke;
55     switch(en)
56     {
57         case CHU_NHAT: engay_ke = THU_HAI; break;
58         case THU_HAI : engay_ke = THU_BA; break;
59         case THU_BA : engay_ke = THU_TU; break;
60         case THU_TU : engay_ke = THU_NAM; break;
61         case THU_NAM : engay_ke = THU_SAU; break;
62         case THU_SAU : engay_ke = THU_BAY; break;
63         case THU_BAY : engay_ke = CHU_NHAT; break;
64     }
65     return (engay_ke);
66 }

```

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

☞ Kết quả in ra màn hình

```

Nhap vao luong can ban: 250
Nhap vao so gio lam viec tu Thu hai den Chu n
Nhap vao gio lam viec ngay Thu Hai: 7
Nhap vao gio lam viec ngay Thu Ba: 8
Nhap vao gio lam viec ngay Thu Tu: 6
Nhap vao gio lam viec ngay Thu Nam: 7
Nhap vao gio lam viec ngay Thu Sau: 8
Nhap vao gio lam viec ngay Thu Bay: 7
Nhap vao gio lam viec ngay Chu Nhat: 6
Tong luong tuan = 14625.00 dong.
_

```

Hàm chọn ngày kế tiếp trên khá dài, bạn thay từ dòng 54 đến 65 bằng câu lệnh

return (++en > 6 ? 0 : en); hoặc

return (++en % 7);

Chạy lại chương trình, quan sát, nhận xét và đánh giá kết quả với dữ liệu khác.

☞ Giải thích chương trình

Ở chương trình này ta phải khai báo struct nhanvien là biến toàn cục, vì khi định nghĩa hàm input và output có sử dụng kiểu dữ liệu struct nhanvien.

10.3 Bài tập

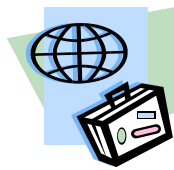
1. Định nghĩa 1 dãy cấu trúc có thể được dùng làm danh bạ điện thoại, gồm có tên, địa chỉ, số điện thoại, với số mẫu tin tối đa là 40. Viết chương trình với các chức năng sau: nhập thông mới, tìm kiếm số điện thoại, in danh sách theo quận.

2. Viết chương trình đọc vào tên, địa chỉ, sắp xếp tên và địa chỉ theo thứ tự alphabet, sau đó hiển thị danh sách đã được sắp xếp.

3. Viết chương trình nhận vào các thông tin sau: Tên đội bóng, số trận thắng, số trận hòa, số trận thua. In ra đội bóng có số điểm cao nhất (với 1 trận thắng = 3 điểm, 1 trận hòa = 1 điểm và 1 trận thua = 0 điểm).

4. Xây dựng cấu trúc gồm: Họ tên, ngày sinh, trường, số báo danh, điểm thi. Trong đó, điểm thi là cấu trúc gồm 3 môn: Toán, Lý, Hóa. Nhập liệu vào khoảng 10 thí sinh, tìm và in ra các thí sinh có tổng điểm 3 môn ≥ 15 .

5. Viết chương trình tạo lập và tìm kiếm dữ liệu. Nội dung yêu cầu gồm: Nhập họ và tên, địa chỉ (gồm: Quận, phường, tổ), tuổi, lương. Tìm kiếm những người ở Quận 3 có tuổi dưới 30 thu nhập từ 500.000đ trở lên và in ra màn hình.



Bài 11 :

TẬP TIN

11.1 Mục tiêu

Sau khi hoàn tất bài này học viên sẽ hiểu và vận dụng các kiến thức kỹ năng cơ bản sau:

- Ý nghĩa của việc sử dụng tập tin (file)
- Mở, đóng file
- Ghi, đọc file số nguyên, mảng, chuỗi.
- Một số hàm xử lý tập tin.

11.2 Nội dung

11.2.1 Ví dụ ghi, đọc số nguyên

Dòng	File	Edit	Search	Run	Compile	Debug	Project	Option	Window	Help
1	/* Ghi n so nguyen vao file va doc ra tu file*/									
2										
3	#include <stdio.h>									
4	#include <conio.h>									
5	#include <stdlib.h>									
6										
7	void main(void)									
8	{									
9	FILE *f;									
10	int in, i;									
11	printf("Nhap vao so n: ");									
12	scanf("%d", &in);									
13										
14	//Ghi file									
15	if((f = fopen("int_data.dat", "wb")) == NULL) //mo file									
16	{									
17	printf("Khong the mo file!\n");									
18	exit(0);									
19	}									
20	else									
21	for(i = 1; i <= in; i++)									
22	fwrite(&i, sizeof(int), 1, f); //ghi file									
23	fclose(f); //dong file									
24										
25	//Doc file									
26	f = fopen("int_data.dat", "rb");									
27	while(fread(&i, sizeof(int), 1, f) == 1)									
28	printf("%d ", i);									
29	fclose(f);									
30	getch();									
31	}									
	F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu									

☞ **Kết quả in ra màn hình**

Nhập vào số n: 10 1 2 3 4 5 6 7 8 9 10 —	Chạy và thử lại chương trình với dữ liệu khác. Quan sát kết quả.
--	---

☞ **Giải thích chương trình**

- **Dòng 9: FILE *f;** : khai báo biến con trỏ f có kiểu cấu trúc FILE.
- **Dòng 15: if(f = fopen("int_data.dat", "wb") == NULL)** : là câu lệnh mở tập tin có tên int_data.dat ở mode "w" (ghi) dạng "b" (nhị phân), sau khi lệnh này thực hiện xong trả về dạng con trỏ FILE và gán cho f, nếu kết quả trả về = NULL thì không thể mở được tập tin, tập tin mở ở mode "w" nếu trên đĩa đã có sẵn tập tin này thì nội dung của nó sẽ bị ghi đè, nếu chưa có thì tập tin sẽ được tạo mới.
- **Dòng 22: fwrite(&i, sizeof(int), 1, f);** : ghi thông tin vào tập tin, thông tin được ghi vào mỗi lần là một số nguyên i. Hàm này có 4 đối số: địa chỉ để ghi cấu trúc, kích thước của cấu trúc và số cấu trúc sẽ ghi, sau cùng là con trỏ để trỏ tới tập tin.
- **Dòng 23: fclose(f);** : đóng tập tin
- **Dòng 26: f = fopen("int_data.dat", "rb");** : mở tập tin có tên int_data.dat ở mode "r" (đọc) dạng "b" (nhị phân). Tập tin phải có sẵn trên đĩa.
- **Dòng 27: while(fread(&i, sizeof(int), 1, f) == 1)** : đọc thông tin từ tập tin, mỗi lần đọc một số nguyên và lưu vào biến i. Mỗi lần đọc thành công giá trị trả về sẽ là số cấu trúc thực sự được đọc, nếu giá trị trả về = 0 báo hiệu kết thúc file.


Formatted: Bullets and Numbering

☞ **Từ khóa FILE phải viết bằng chữ in hoa. Sử dụng fopen, fwrite, fread, fclose phải khai báo #include <stdio.h>, NULL phải viết hoa.**

11.2.2 Ghi, đọc mảng

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	/* Ghi n số nguyên vào file và đọc ra từ file*/
2	
3	#include <stdio.h>
4	#include <conio.h>
5	#include <stdlib.h>
6	
7	#define MAX 5
8	
9	void main(void)
10	{
11	FILE *f;
12	int i, ia[MAX], ib[MAX];
13	for (i = 0; i < 10; i++)
14	{
15	printf("Nhập vào một số: ");
16	scanf("%d", &ia[i]);
17	}
18	
19	if((f = fopen("array.dat", "wb")) == NULL)
20	{
21	printf("Không thể mở file!\n");
22	exit(0);

23	}	
24	fwrite(ia, sizeof(ia), 1, f);	//ghi mang vao file
25	fclose(f);	
26		
27	f = fopen("array.dat", "rb");	
28	fread(ib, sizeof(ib), 1, f);	//doc mang tu file
29	for (i = 0; i < 10; i++)	
30	printf("%d ", ib[i]);	
31	fclose(f);	
32	getch();	
33	}	
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu		

 **Kết quả in ra màn hình**

Nhap vao mot so: 3 Nhap vao mot so: 6 Nhap vao mot so: 8 Nhap vao mot so: 1 Nhap vao mot so: 9 3 6 8 1 9 —	Chạy và thử lại chương trình với dữ liệu khác. Quan sát kết quả.
--	---

11.2.3 Ghi, đọc structure

Dòng	File	Edit	Search	Run	Compile	Debug	Project	Option	Window	Help
1	/*	Danh	sach	nhan	vien	*/				
2										
3	#include	<stdio.h>								
4	#include	<conio.h>								
5	#include	<stdlib.h>								
6										
7	#define	MAX	50							
8										
9	void	main	(void)							
10	{									
11	FILE	*f;								
12	struct	nhanvien								
13	{									
14	int	manv;								
15	char	hoten[30];								
16	};									
17	nhanvien	snv[MAX],	snv1[MAX];							
18	char	ctam[10];								
19	int	i, in;								
20	printf	("Nhap	vao	so	nhan	vien:");				
21	gets	(ctam);								
22	in	=	atoi	(ctam);						
23										
24	//Nhap	danh	sach	nhan	vien	va	ghi	vao	file	
25	if	((f	=	fopen	("struct.dat",	"wb"))	==	NULL)		
26	{									

```

27     printf("Khong the mo file\n");
28     exit(0);
29 }
30 fwrite(&in, sizeof(int), 1, f);           //ghi so nhan vien vao file
31 for(i = 0; i < in; i++)
32 {
33     printf("Nhap vao ma nhan vien thu %d: ", i + 1);
34     gets(ctam);
35     snv[i].manv= atoi(ctam);
36     printf("Nhap vao ho ten: ");
37     gets(snv[i].hoten);
38     fwrite(&snv[i], sizeof(nhanvien), 1, f); //ghi tung nhan vien vao file
39 }
40 fclose(f);
41
42 //doc danh sach nhan vien tu file va in ra
43 f = fopen("struct.dat", "rb");
44 fread(&in, sizeof(in), 1, f);           //doc so nhan vien
45 for(i = 0; i < in; i++)
46 {
47     fread(&snv1[i], sizeof(nhanvien), 1, f); //doc tung nhan vien in ra man hinh
48     printf("%5d %s\n", snv[i].manv, snv[i].hoten);
49 }
50 getch();
51 }

```

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

Kết quả in ra màn hình

Nhập vào số nhân viên: 2 Nhập vào mã nhân viên thứ 1: 123 Nhập vào họ tên: Le Thuy Doan Tran Nhập vào mã nhân viên thứ 2: 35 Nhập vào họ tên: Le Nguyen Tuan Anh 123 Le Thuy Doan Tran 35 Le Nguyen Tuan Anh -	Chạy và thử lại chương trình với dữ liệu khác. Quan sát kết quả.
---	---

11.2.4 Các mode khác để mở tập tin

Ở 3 ví dụ trên chỉ sử dụng 2 mode "w" (ghi) và "r" (đọc), sau đây là một số mode khác:

- "a": mở để nội thêm, thông tin sẽ được ghi thêm vào cuối của tập tin đã có hoặc tạo tập tin mới nếu chưa có trên đĩa.
- "r+": mở để vừa đọc vừa ghi, tập tin phải có sẵn trên đĩa.
- "w+": mở để vừa đọc vừa ghi, nội dung của tập tin đã có trên đĩa sẽ bị ghi đè lên.
- "a+": mở để đọc và nội thêm, nếu trên đĩa chưa có tập tin nó sẽ được tạo mới.

11.2.5 Một số hàm thao tác trên file khác

Xem bài Các hàm chuẩn

11.3 Bài tập

Thêm chức năng ghi, đọc file ở các bài tập của bài Mạng và chuỗi, Các dữ liệu tự tạo.

Bài 12 :**ĐỆ QUY****12.1 Mục tiêu**

Sau khi hoàn tất bài này học viên sẽ hiểu và vận dụng các kiến thức kỹ năng cơ bản sau:

- Ý nghĩa, phương pháp hoạt động của đệ quy.
- Có thể thay vòng lặp bằng đệ quy.

12.2 Nội dung

Bất cứ một hàm nào đó có thể triệu gọi hàm khác, nhưng ở đây một hàm nào đó có thể tự triệu gọi chính mình. Kiểu hàm như thế được gọi là **hàm đệ quy**.

Phương pháp đệ quy thường dùng phổ biến trong những ứng dụng mà cách giải quyết có thể được thể hiện bằng việc áp dụng liên tiếp cùng giải pháp cho những tập hợp con của bài toán.

Ví dụ 1: tính $n!$

$n! = 1*2*3*...*(n-2)*(n-1)*n$ với $n \geq 1$ và $0! = 1$.

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	/* Ham tinh giai thua */
2	
3	#include <stdio.h>
4	#include <conio.h>
5	
6	void main(void)
7	{
8	int in;
9	long giai thua(int);
10	printf("Nhap vao so n: ");
11	scanf("%d", &in);
12	printf("%d! = %ld\n", in, giai thua(in));
13	getch();
14	}
15	
16	long giai thua(int in)
17	{
18	int i;
19	long ltich = 1;
20	if (in == 0)
21	return (1L);
22	else
23	{
24	for (i = 1; i <= in; i++)
25	ltich *= i;
26	return (ltich);
27	}
28	}
	F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

Kết quả in ra màn hình

Nhập vào số n: 5 5! = 120. —	Thử lại chương trình với số liệu khác.
------------------------------------	--

Với $n! = 1 * 2 * 3 * \dots * (n-2) * (n-1) * n$,
ta viết lại như sau: $(1 * 2 * 3 * \dots * (n-2) * (n-1)) * n = n * (n-1)! \dots = n * (n-1) * (n-2)! \dots$

Ta viết lại hàm giaithua bằng đệ quy như sau:

```

Dòng File Edit Search Run Compile Debug Project Option Window Help
1 /* Hàm tính giai thừa */
2
3 long giaithua(int in)
4 {
5     int i;
6     if (in == 0)
7         return (1L);
8     else
9         return (in * giaithua(in- 1));
10 }
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu
    
```

Chạy lại chương trình, quan sát, nhận xét và đánh giá kết quả

Giải thích hoạt động của hàm đệ quy giaithua

Ví dụ giá trị truyền vào hàm giaithua qua biến in = 5.

- Thứ tự gọi thực hiện hàm giaithua

giaithua(in)	return(in * giaithua(in- 1))
5	5 * giaithua(4) = 5 * ?
4	4 * giaithua(3) = 4 * ?
3	3 * giaithua(2) = 3 * ?
2	2 * giaithua(1) = 2 * ?
1	1 * giaithua(0) = 1 * ?

Khi tham số in = 0 thì return về giá trị 1L (giá trị 1 kiểu long). Lúc này các giá trị ? bắt đầu định trị theo thứ tự ngược lại.

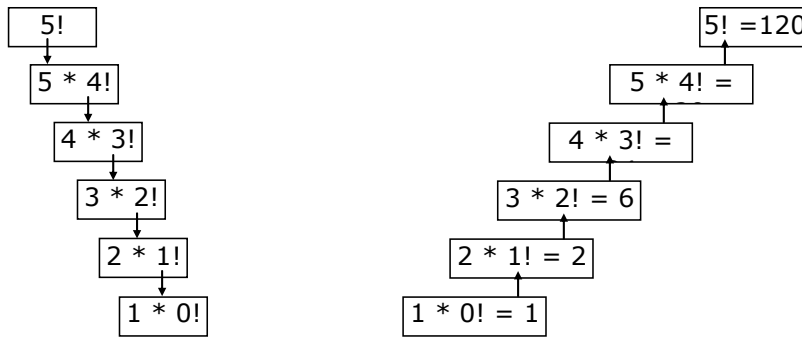
- Định trị theo thứ tự ngược lại

giaithua(in)	return(in * giaithua(in- 1))
1	1 * giaithua(0) = 1 * 1 = 1
2	2 * giaithua(1) = 2 * 1 = 2
3	3 * giaithua(2) = 3 * 2 = 6
4	4 * giaithua(3) = 4 * 6 = 24
5	5 * giaithua(4) = 5 * 24 = 120

Kết quả sau cùng ta có 5! = 120.

Formatted: Bullets and Numbering

Formatted: Bullets and Numbering



Thứ tự gọi đệ quy

Định trị theo thứ tự ngược lại

Ví dụ 2: Dãy số Fibonacci

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ... Bắt đầu bằng 0 và 1, các số tiếp theo bằng tổng hai số đi trước.

Dãy Fibonacci được khai báo đệ quy như sau:

$$\begin{cases} \text{Fibonacci}(0) = 0 \\ \text{Fibonacci}(1) = 1 \\ \text{Fibonacci}(n) = \text{Fibonacci}(n-1) + \text{Fibonacci}(n-2) \end{cases}$$

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	/* Tính số fibonacci thu n */
2	
3	#include <stdio.h>
4	#include <conio.h>
5	
6	void main(void)
7	{
8	long in;
9	long fibonacci(long);
10	printf("Nhap vao so n: ");
11	scanf("%ld", &in);
12	printf("Fibonacci(%ld) = %ld\n", in, fibonacci(in));
13	getch();
14	}
15	
16	long fibonacci(long in)
17	{
18	if (in == 0 in == 1)
19	return in;
20	else
21	return fibonacci(in-1) + fibonacci(in-2);
22	}
	F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

Kết quả in ra màn hình

Nhap vao so n: 10 Fibonacci(10) = 55. —	Thử lại chương trình với số liệu khác.
---	--

Sử dụng đệ quy hay vòng lặp

Phương pháp đệ quy không phải bao giờ cũng là giải pháp hữu hiệu nhất. Giải pháp vòng lặp có hiệu quả về mặt thời gian và vùng nhớ. Còn với đệ quy mỗi lần gọi đệ quy máy phải dành một số vùng nhớ để trữ các trị, thông số và biến cục bộ. Do đó, đệ quy tốn nhiều vùng nhớ, thời gian truyền đổi mục, thiết lập vùng nhớ trung gian và trả về kết quả... Nhưng sử dụng phương pháp đệ quy trông chương trình đẹp mắt hơn vòng lặp và tính thuyết phục của nó. Điều cốt lõi khi thiết đặt chương trình phải làm thế nào hàm đệ quy có thể chấm dứt thông qua điều kiện cơ bản.

12.3 Bài tập

- Viết hàm đệ quy tính tổng n số nguyên dương đầu tiên:
$$\text{tong}(n) = n + \text{tong}(n - 1)$$



Bài 13 :**TRÌNH SOẠN THẢO CỦA BORLAND C**

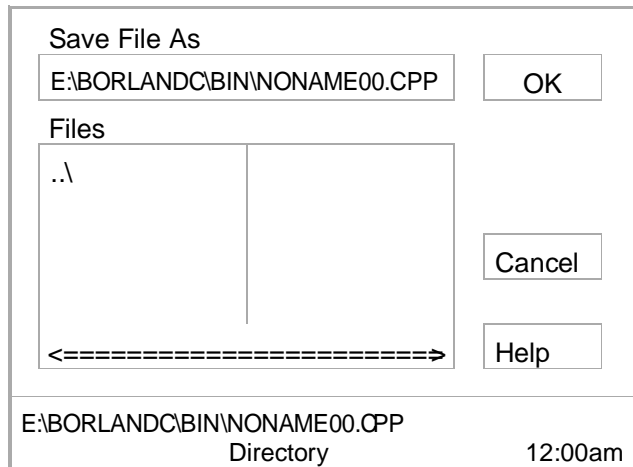
BC có hệ thống menu nhiều cấp. Để chọn một mục trong Menu bạn ấn phím **F10** (kích hoạt Menu), dùng các phím mũi tên di chuyển vết sáng đến mục muốn chọn ấn **Enter** hoặc ấn phím có kí tự đổi màu (**phím chọn nhanh màu đỏ**). Có thể chọn nhanh mục menu trên thanh menu chính bạn ấn tổ hợp phím **Alt + phím có kí tự màu đỏ**. Ví dụ: ấn tổ hợp phím **Alt + F** kích hoạt menu File.

13.1 Mở tập tin soạn thảo mới

Chọn menu File -> chọn mục New -> tạo file soạn thảo mới có tên mặc định là NONAME00.CPP, NONAME01.CPP... tùy theo số lần mục New được chọn.

13.2 Lưu tập tin**13.2.1 Nếu là tập tin soạn thảo mới chưa lưu**

Ấn phím **F2** hoặc chọn menu File -> Save hoặc chọn menu File -> Save As sẽ xuất hiện hộp thoại Save File As



- + Chọn đường dẫn cần lưu tập tin ở hộp Files, chọn ..\ để trở về thư mục cha thư mục hiện tại.
- + Đặt tên tập tin ở hộp Save File As
- + Chọn OK
- + Hoặc có thể gõ [ô đĩa:][đường dẫn]<tên tập tin>, chọn OK.

☞ Dùng phím TAB để chuyển đổi vết sáng giữa các mục trong hộp thoại.

Ví dụ: muốn lưu tập tin có tên BT_IF1.CPP vào thư mục D:\BAITAPC

- + Bạn gõ vào hộp Save File As D: -> Enter (OK), danh sách các thư mục, tập tin của D hiển thị ở hộp Files, chọn thư mục BAITAPC ở hộp Files, gõ tên BT_IF1.CPP vào hộp Save File As, chọn OK.
- + Hoặc nếu bạn nhớ rõ đường dẫn, gõ vào hộp Save File As D:\BAITAPC\BT_IF1, chọn OK.

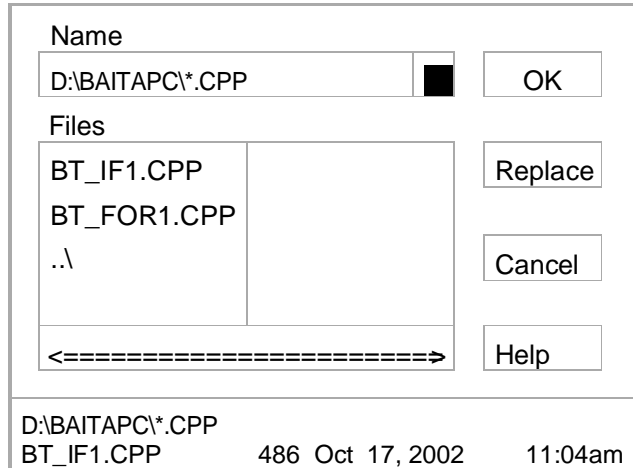
13.2.2 Nếu là tập tin đã lưu ít nhất 1 lần hoặc được mở bằng lệnh Open:

- + Ấn **F2** hoặc chọn menu File -> Save, nội dung tập tin hiện hành sẽ được cập nhật nếu có thay đổi.
- + Chọn menu File -> Save As, xuất hiện hộp thoại Save File As, thực hiện các bước như mục 2.1 (nghĩa là bạn muốn lưu nội dung tập tin hiện hành với đường dẫn, tên tập tin khác).

☞ Để biết tập tin đang soạn thảo đã lưu hay chưa, bạn xem ở góc dưới trái cửa sổ, nếu có dấu hoa thị là văn bản của bạn có thay đổi và chưa được lưu.

13.3 Mở tập tin

Mở một tập tin đã có trên đĩa. Ấn phím F3 hoặc chọn menu File -> Open, hộp thoại Open a File xuất hiện:



- + Chọn đường dẫn cần mở tập tin ở hộp Files, chọn ..\ để trở về thư mục cha thư mục hiện tại.
- + Chọn tập tin cần mở ở hộp Files.
- + Chọn OK
- + Hoặc có thể gõ [ổ đĩa:][đường dẫn]<tên tập tin>, chọn OK.

Ví dụ: muốn mở tập tin có tên BT_IF1.CPP chứa trong thư mục D:\BAITAPC

- + Bạn gõ vào hộp Name D: -> Enter (OK), danh sách các thư mục, tập tin của D hiển thị ở hộp Files, chọn thư mục BAITAPC ở hộp Files, chọn tập tin BT_IF1.CPP, chọn OK.
- + Hoặc bạn có thể gõ vào hộp Name D:\BAITAPC*.CPP để hiển thị danh sách các tập tin có phần mở rộng CPP ở hộp Files, chọn tập tin BT_IF1.CPP, chọn OK.
- + Hoặc nếu bạn nhớ rõ đường dẫn, gõ vào hộp Name D:\BAITAPC\BT_IF1.CPP, chọn OK.

13.4 Các phím, tổ hợp phím thường dùng

13.4.1 Các phím di chuyển con trỏ

Phím / Tổ hợp phím	Chức năng
←	Di chuyển con trỏ sang trái một ký tự
→	Di chuyển con trỏ sang phải một ký tự
↑	Di chuyển con trỏ lên trên một dòng
↓	Di chuyển con trỏ xuống dưới một dòng
Home	Di chuyển con trỏ về đầu dòng
End	Di chuyển con trỏ về cuối dòng
PgUp (Page Up)	Lật lùi lại một trang màn hình
PgDn (Page Down)	Lật tới một trang màn hình
Ctrl – PgUp	Di chuyển con trỏ về đầu tập tin
Ctrl – PgDn	Di chuyển con trỏ về cuối tập tin

Backspace	Xóa một ký tự bên trái con trỏ
Del (Delete)	Xóa một ký tự tại vị trí con trỏ
Ins (Insert)	Chuyển đổi giữa chế độ ghi chèn và ghi đè
Enter	Xuống một dòng

13.4.2 Các phím thao tác trên khối

Phím / Tổ hợp phím	Chức năng
Shift –	Đánh dấu chọn một ký tự bên phải
Shift –	Đánh dấu chọn một ký tự bên trái
Shift –	Đánh dấu chọn một hàng trên vị trí con trỏ
Shift –	Đánh dấu chọn một hàng tại vị trí con trỏ
Shift – Home	Đánh dấu chọn từ đầu hàng đến vị trí con trỏ
Shift – End	Đánh dấu chọn từ vị trí con trỏ đến cuối hàng
Shift – PgUp	Đánh dấu chọn một trang lui màn hình
Shift – PgDn	Đánh dấu chọn một trang tới màn hình
Ctrl – Shift –	Đánh dấu chọn một từ bên trái
Ctrl – Shift –	Đánh dấu chọn một từ bên phải
Ctrl – Shift – End	Đánh dấu chọn từ vị trí con trỏ đến cuối tập tin
Ctrl – Shift – Home	Đánh dấu chọn từ vị trí con trỏ đến đầu tập tin

13.4.3 Các thao tác xóa

Phím	Chức năng
Backspace	Xóa một ký tự bên trái con trỏ
Del (Delete)	Xóa một ký tự tại vị trí con trỏ
Ctrl – Y	Xóa dòng tại vị trí con trỏ
Ctrl – K – Y	Xóa khối
Ctrl – Q – Y	Xóa từ vị trí con trỏ đến cuối dòng
Ctrl – T	Xóa một từ tại vị trí con trỏ
Inset	Bật / tắt chế độ viết chèn / đè

13.4.4 Các thao tác copy, di chuyển

Phím / Tổ hợp phím	Chức năng
Ctrl – Insert	Sao chép khối chọn vào Clipboard
Shift – Delete	Cắt khối chọn vào Clipboard
Ctrl – Delete	Xóa khối chọn
Shift – Insert	Dán thông tin từ Clipboard vào vị trí con trỏ
Ctrl – K – R	Đọc thông tin từ tập tin vào cửa sổ soạn thảo
Ctrl – K – W	Ghi thông tin từ cửa sổ soạn thảo vào tập tin

13.4.5 Các thao tác khác

Phím / Tổ hợp phím	Chức năng
F3	Tạo tập tin mới hoặc nạp tập tin từ đĩa vào cửa sổ

	soạn thảo
Alt – F3	Đóng tập tin tại cửa sổ hiện hành
F2	Lưu tập tin hiện hành
F6	Chuyển đổi qua lại giữa các cửa sổ đang soạn thảo
F5	Chuyển đổi cửa sổ soạn thảo maximize  store
Alt – Backspace	Phục hồi lại thao tác trước đó
Ctrl – K – H	Ẩn / hiện dấu khối
Ctrl – Q – F	Tìm kiếm
Ctrl – L	Lập lại lần tìm kiếm sau cùng
Ctrl – Q – A	Tìm kiếm và thay thế
Ctrl – Q – [, Ctrl – Q –]	Xác định cặp ngoặc bao 1 khối lệnh
F1	Gọi giúp đỡ
Shift – F1	Hiện cửa sổ giúp đỡ theo mục
Ctrl – F1	Hiện cửa sổ giúp đỡ về hàm, toán tử... tương ứng tại vị trí con trỏ.

13.5 Ghi một khối ra đĩa

Đánh dấu chọn khối bằng các phím thao tác trên khối. Ấn tổ hợp phím Ctrl - K - W, xuất hiện hộp thoại Write Block to File, thực hiện các bước như lưu tập tin.

13.6 Chèn nội dung file từ đĩa vào vị trí con trỏ

Di chuyển con trỏ đến vị trí cần chèn nội dung, Ấn tổ hợp phím Ctrl - K - R, xuất hiện hộp thoại Read Block from File, thực hiện các bước như mở tập tin.

13.7 Tìm kiếm văn bản trong nội dung soạn thảo

Ấn tổ hợp phím Ctrl - Q - F hoặc chọn menu Search -> Find, hộp thoại Find Text xuất hiện:

- + Gõ nội dung cần tìm vào hộp Text to Find.
- + Nếu cần đánh dấu / bỏ chọn các mục sau:
 - Casesensitive: phân biệt chữ hoa chữ thường.
 - Whole words only: tìm văn bản đứng riêng một từ.
 - Forward: Tìm xuôi.
 - Backward: Tìm ngược.
- + Chọn OK.

Khi tìm xong, muốn tìm tiếp ấn tổ hợp phím Ctrl - L hoặc chọn menu Search -> Search again.

13.8 Tìm và thay thế văn bản trong nội dung soạn thảo

Ấn tổ hợp phím Ctrl - Q - A hoặc chọn menu Search -> Replace, hộp thoại Find Text xuất hiện:

- + Gõ nội dung cần thay thế vào hộp Text to Find.
- + Gõ nội dung mới vào hộp New Text.
- + Nếu cần đánh dấu / bỏ chọn các mục sau:
 - Casesensitive: phân biệt chữ hoa chữ thường.
 - Whole words only: tìm văn bản đứng riêng một từ.
 - Forward: Tìm xuôi.
 - Backward: Tìm ngược.
- + Chọn OK để thay thế từng văn bản được tìm thấy, chọn Change All để thay thế tất cả.

13.9 Sửa lỗi cú pháp

Khi biên dịch chương trình, nếu thành công bạn sẽ nhận được thông báo từ cửa sổ Compile (dòng cuối): **Success: Press any key**, ngược lại là thông báo lỗi **Error: Press any key**.

Nếu là thông báo lỗi, khi ấn phím bất kỳ cửa sổ Message xuất hiện chứa danh sách các lỗi. Thông báo lỗi đầu tiên được làm sáng và dòng có lỗi trong chương trình cũng được làm sáng. kèm theo dấu đỏ cho biết trình biên dịch phát hiện vị trí lỗi. Dùng phím mũi tên để di chuyển đến các thông báo lỗi khác, bạn sẽ thấy vệt sáng trong chương trình cũng sẽ chuyển đến dòng chứa lỗi tương ứng. Nếu bạn Enter tại dòng thông báo lỗi nào thì con trỏ sẽ chuyển vào cửa sổ soạn thảo tại dòng chứa lỗi tương ứng.

Ví dụ: In ra "Hello".

File	Edit	Search	Run	Compile	Debug	Project	Option	Window	Help
<pre>#include <stdio.h> #include <conio.h> void main(void) { printf("Hello"; gech(); }</pre>									
Message									
Compiling HELLO.CPP									
Error HELLO.CPP 5: Function call missing)									
Error HELLO.CPP 6: Function 'gech' should have a prototype									
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu									

Vệt sáng nằm ở thông báo lỗi đầu tiên và dòng chứa lỗi tương ứng trong chương trình cũng được làm sáng: Lỗi ở dòng 5, không đóng ngoặc hàm printf.

13.10 Chạy từng bước

13.10.1 Mỗi lần 1 bước

Ở mỗi bước thực hiện ta phải bấm phím F7. Với lần bấm F7 đầu tiên, dòng đầu tiên trong chương trình (dòng main()) sẽ được làm sáng, dòng được làm sáng này được gọi là dòng chuẩn bị thực hiện, và nó sẽ được thực hiện ở lần bấm phím F7 tiếp theo. Mỗi lần bấm phím F7 dòng đang được làm sáng sẽ được thực hiện, sau đó trở về màu bình thường, và tùy theo nội dung của dòng đó mà một dòng lệnh tiếp theo nào đó sẽ được làm sáng để chuẩn bị thực hiện ở bước tiếp theo.

Ta cũng có thể dùng phím F8 thay cho F7 với những dòng không có lời gọi hàm được khai báo trong chương trình. Sự khác nhau giữa F7 và F8 chỉ xảy ra khi dòng được làm sáng có lời gọi hàm được khai báo trong chương trình.

Như vậy nhờ chạy từng bước, ta có thể dễ dàng nắm được các lỗi logic trong chương trình.

13.10.2 Tái lập lại quá trình gỡ rối

Bấm Ctrl-F2 hoặc vào menu Run chọn Program reset. Khi đó bộ nhớ dùng cho việc gỡ rối sẽ được giải tỏa, không có dòng nào được làm sáng và kết thúc quá trình gỡ rối.

13.10.3 Dừng cửa sổ Watch

Lần từng bước thường được dùng kèm với việc sử dụng cửa sổ Watch để theo dõi giá trị của biến trong mỗi bước thực hiện để dễ tìm ra nguyên nhân chương trình thực hiện sai.

Để làm điều đó ta phải nhập vào các biến cần theo dõi, bằng cách chọn mục Add watch của menu Break/Watch hoặc có thể bấm Ctrl-F7, sau đó nhập tên biến vào tại vị trí con trỏ trong cửa sổ Add watch và bấm Enter. Để nhập thêm tên biến vào cửa sổ này phải lập lại chức năng này hoặc bấm phím Insert.

Trong soạn thảo nếu chưa nhìn thấy cửa sổ Watch, ta bấm phím F5, khi đó trên màn hình sẽ đồng thời hiện ra cả 2 cửa sổ, để chuyển đổi giữa 2 cửa sổ này bấm phím F6. Mỗi biến trên cửa sổ Watch thực hiện trên 1 dòng. Khi cửa sổ Watch được chọn sẽ có 1 dòng được làm sáng để chỉ rằng biến đó đang được chọn. Giá trị trong cửa sổ Watch sẽ thay đổi theo kết quả của từng bước thực hiện.

13.11 Sử dụng Help (Giúp đỡ)

- Ấn phím F1 để kích hoạt màn hình Help chính.
- Muốn xem Help của hàm trong soạn thảo, di chuyển con trỏ đến vị trí hàm đó ấn tổ hợp phím Ctrl- F1
- Ấn tổ hợp phím Shift - F1 để xem danh sách các mục Help
- Ấn tổ hợp phím Alt - F1 để quay về màn hình Help trước đó.



BÀI 14 :

CÁC HỆ ĐẾM

14.1 Khái niệm

Các chữ số cơ bản của một hệ đếm là các chữ số dùng để biểu diễn mọi số trong hệ đếm ấy. Hệ đếm thường gặp nhất là hệ thập phân (hệ 16). Nhưng do bản chất nhị phân của các thiết bị điện tử cho nên hầu hết dạng biểu diễn dữ liệu và các phép đại số đều thực hiện bằng hệ nhị phân (hệ 2). Hệ bát phân (hệ 8) rất ít dùng và hệ thập phân (hệ 10) là hệ chúng ta đang sử dụng để biểu diễn một con số nào đó trong cuộc sống hằng ngày.

Ví dụ 1: Hệ nhị phân gồm 2 chữ số : 0, 1
 Hệ bát phân gồm 8 chữ số : 0, 1, 2, 3, 4, 5, 6, 7
 Hệ thập phân gồm 10 chữ số : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
 Hệ thập lục phân gồm 16 chữ số : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

■ **Các chữ số trong một hệ đếm được sắp xếp theo quy tắc:** Bất kỳ cơ số N nguyên dương nào, có N ký hiệu khác nhau để biểu diễn các số trong hệ thống. Giá trị của N ký hiệu này được sắp xếp từ 0 đến N-1.

Ví dụ 2: Hệ nhị phân có cơ số N = 2 : các chữ số được đánh từ 0..1
 Hệ bát phân có cơ số N = 8 : các chữ số được đánh từ 0..7
 Hệ thập phân có cơ số N = 10 : các chữ số được đánh từ 0..9
 Hệ thập lục phân có cơ số N = 16 : các chữ số được đánh từ 0..9, A..F

■ Do hệ thập lục phân có 16 chữ số, mà trong hệ thông chữ viết chỉ biểu diễn được 9 chữ số, vì vậy người ta chọn các ký tự A..F để biểu diễn 10..15 và nó cũng được xem như 1 chữ số (A, B...F) thay vì phải viết 10, 11...15 (2 chữ số)

14.2 Quy tắc

Để biểu diễn một số của một hệ đếm, ta dùng chỉ số đặt ở góc dưới phải số đó.

0110₂ : biểu thị số nhị phân.

08₈ : biểu thị số bát phân.

23₁₆ : biểu thị số thập lục phân.

Đối với hệ thập phân ta có thể ghi chỉ số hoặc không ghi (nhầm hiểu), vì số thập phân là số mà ta sử dụng quen thuộc hằng ngày. Do đó, ta sử dụng công thức sau để chuyển đổi từ các hệ đếm sang hệ thập phân (cơ số 10) :

$$X = a_n a_{n-1} \dots a_1 a_0 = a_n b^n + a_{n-1} b^{n-1} + \dots + a_1 b^1 + a_0 b^0 \quad (*)$$

trong đó,

- **b** : là cơ số hệ đếm.
- **a₀...a_n** : là các chữ số trong một hệ đếm.
- **X** : là số thuộc một hệ đếm cơ số b.

■ **Bảng các giá trị tương đương ở hệ thập phân, nhị phân, bát phân, thập lục phân. (**)**

Thập phân	Nhị phân	Bát phân	Thập lục phân
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6

7	0111	7	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

14.3 Chuyển đổi giữa các hệ

14.3.1 Chuyển đổi giữa hệ 2 và hệ 10

■ Chuyển đổi từ hệ 2 sang hệ 10

Ví dụ 3: $X = 01011_2$, để chuyển sang hệ 10 ta dùng công thức (*)
 $X = 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$
 $= 0 + 8 + 0 + 2 + 1$
 $= 11$

Ví dụ 4: $X = 1011010_2$, để chuyển sang hệ 10 ta dùng công thức (*)
 $X = 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0$
 $= 64 + 0 + 16 + 8 + 0 + 2 + 0$
 $= 90$

■ Chuyển đổi từ hệ 10 sang hệ 2

Ví dụ 5: $X = 11$

11	2		
1	5	2	
	1	2	2
	0	1	2
		1	0

11 chia 2 = 5 dư 1

5 chia 2 = 2 dư 1

2 chia 2 = 1 dư 0

1 chia 2 = 0 dư 1

$\rightarrow 1011_2 \rightarrow$ kết quả hệ nhị phân $\leftarrow 1011_2 \leftarrow$

Ví dụ 6: $X = 90$

90	2		
0	45	2	
	1	22	2
	0	11	2
		1	5
		1	2
		0	1
			0

90 chia 2 = 45 dư 0

45 chia 2 = 22 dư 1

22 chia 2 = 11 dư 0

11 chia 2 = 5 dư 1

5 chia 2 = 2 dư 1

2 chia 2 = 1 dư 0

1 chia 2 = 0 dư 1

$\rightarrow 1011010_2 \rightarrow$ kết quả hệ nhị phân $\leftarrow 1011010_2 \leftarrow$

14.3.2 Chuyển đổi giữa hệ 8 và hệ 10

■ Chuyển đổi từ hệ 8 sang hệ 10

Ví dụ 7: $X = 2106_8$, để chuyển sang hệ 10 ta dùng công thức (*)
 $X = 2 \cdot 8^3 + 1 \cdot 8^2 + 0 \cdot 8^1 + 6 \cdot 8^0$
 $= 1024 + 64 + 0 + 6$
 $= 1094$

Ví dụ 8: $X = 130_8$, để chuyển sang hệ 10 ta dùng công thức (*)
 $X = 1 \cdot 8^2 + 3 \cdot 8^1 + 0 \cdot 8^0$
 $= 64 + 24 + 0$
 $= 88$

■ Chuyển đổi từ hệ 10 sang hệ 8

Ví dụ 9: $X = 1094$

1094		8		1094 chia 8 = 136 dư 6
6		136		8
0		17		8
1		2		8
2		0		2

$\xrightarrow{\quad} 2106_8 \xrightarrow{\quad} \text{kết quả hệ bát phân} \xleftarrow{\quad} 2106_8 \xleftarrow{\quad}$

Ví dụ 10: $X = 88$

88		8		88 chia 8 = 11 dư 0
0		11		8
3		1		8
1		0		1

$\xrightarrow{\quad} 130_8 \xrightarrow{\quad} \text{kết quả hệ bát phân} \xleftarrow{\quad} 130_8 \xleftarrow{\quad}$

14.3.3 Chuyển đổi giữa hệ 16 và hệ 10

■ Chuyển đổi từ hệ 16 sang hệ 10

Ví dụ 11: $X = F40_{16}$, để chuyển sang hệ 10 ta dùng công thức (*)
 $X = 15 \cdot 16^2 + 4 \cdot 16^1 + 0 \cdot 16^0$
 $= 3840 + 64 + 0$
 $= 3904$

Ví dụ 12: $X = 1D_{16}$, để chuyển sang hệ 10 ta dùng công thức (*)
 $X = 1 \cdot 16^1 + 13 \cdot 16^0$
 $= 16 + 13$
 $= 29$

Chuyển đổi từ hệ 10 sang hệ 16

Ví dụ 13: $X = 3904$

3904		16
0		244
		16
4		15
		16
15		0

3904 chia 16 = 244 dư 0

244 chia 16 = 15 dư 4

15 chia 16 = 0 dư 15

Số 15 tương ứng trong hệ 16 là F (xem bảng (**))

→ kết quả hệ thập lục phân ←

Ví dụ 14: $X = 29$

29		16
13		1
		16
1		0

29 chia 16 = 1 dư 13

1 chia 16 = 0 dư 1

Số 13 tương ứng trong hệ 16 là D (xem bảng (**))

→ kết quả hệ thập lục phân ←

14.3.4 Chuyển đổi giữa hệ 2 và hệ 16

Chuyển đổi từ hệ 2 sang hệ 16

Ví dụ 15: $X = 01011_2$, để chuyển sang hệ 16 ta tra trong bảng (**)

$X = B_{16}$
 Điền giải: $0 \ 1011_2 = B_{16}$

Ví dụ 16: $X = 1011010_2$, để chuyển sang hệ 16 ta tra trong bảng (**)

$X = 5A_{16}$
 Điền giải: $101 \ 1010_2 = 5A_{16}$

Chuyển đổi từ hệ 16 sang hệ 2

Ví dụ 17: $X = B_{16}$, để chuyển sang hệ 2 ta tra trong bảng (**)

$X = 1011_2$
 Điền giải: $B_{16} = 1011_2$

Ví dụ 18: $X = 5A_{16}$, để chuyển sang hệ 2 ta tra trong bảng (**)

$X = 1011010_2$
 Điền giải: $5 \ A_{16} = 1011010_2$

Bài 15 :**BIỂU THỨC VÀ PHÉP TOÁN****15.1 Biểu thức**

Là sự phối hợp của những toán tử và toán hạng.

Ví dụ 1:
 $a + b$
 $b = 1 + 5 * 2/i$
 $a = 6 \% (7 + 1)$
 $x++ * 2/4 + 5 - \text{power}(i, 2)$

Toán hạng sử dụng trong biểu thức có thể là hằng số, biến, hàm.

15.2 Phép toán

Trong C có 4 nhóm toán tử chính yếu sau đây:

15.2.1 Phép toán số học

+ : cộng

- : trừ

***** : nhân

/ : chia

% : lấy phần dư

áp dụng trên tất cả các toán hạng có kiểu dữ liệu char, int, float, double (kể cả long, short, unsigned)

áp dụng trên các toán hạng có kiểu dữ liệu char, int, long

* Thứ tự ưu tiên:

Đảo dấu +, - () *, /, % +, -

Ví dụ 2:
 $10\%4 = 2$ (10 chia 4 dư 2); $9\%3 = 0$ (9 chia 3 dư 0)

 $3 * 5 + 4 = 19$
 $6 + 2 / 2 - 3 = 4$
 $-7 + 2 * ((4 + 3) * 4 + 8) = 65$

☞ chỉ sử dụng cặp ngoặc () trong biểu thức, cặp ngoặc đơn được thực hiện theo thứ tự ưu tiên từ trong ra ngoài.

15.2.2 Phép quan hệ

> : lớn hơn

>= : lớn hơn hoặc bằng

< : nhỏ hơn

<= : nhỏ hơn hoặc bằng

== : bằng

!= : khác

* Thứ tự ưu tiên:

>, >=, <, <= ==, !=

☞ Kết quả của phép toán quan hệ là số nguyên kiểu int, bằng 1 nếu đúng, bằng 0 nếu sai. Phép toán quan hệ ngoài toán hạng được sử dụng là số còn được sử dụng với kiểu dữ liệu char.

* Thứ tự ưu tiên giữa toán tử số học và toán tử quan hệ

Toán tử số học

Toán tử quan hệ

Ví dụ 3:
 $4 > 10$

☐ giá trị 0 (sai)

- 4 >= 4 giá trị 1 (đúng)
- 3 == 5 giá trị 0 (sai)
- 2 <= 1 giá trị 0 (sai)
- 6 != 4 giá trị 1 (đúng)
- 6 - 3 < 4 giá trị 1 (đúng), tương đương (6 - 3) < 4
- 2 * -4 < 3 + 2 giá trị 0 (sai), tương đương (-2 * -4) < (3 + 2)

15.2.3 Phép toán luận lý

- ! : NOT (phép phủ định)
- && : AND (phép và)
- || : OR (phép hoặc)

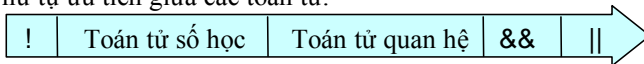
Toán hạng a	Toán hạng b	!a	a && b	a b
Khác 0	Khác 0	0 (sai)	1 (đúng)	1 (đúng)
Khác 0	Bằng 0	0 (sai)	0 (sai)	1 (đúng)
Bằng 0	Khác 0	1 (đúng)	0 (sai)	1 (đúng)
Bằng 0	Bằng 0	1 (đúng)	0 (sai)	0 (sai)

* Thứ tự ưu tiên: ! && ||

Ví dụ 4:

- !(2 <= 1) giá trị 1 (đúng)
- 5 && 10 giá trị 1 (đúng)
- !6 giá trị 0 (sai)
- 1 && 0 giá trị 0 (sai)
- 1 || 0 giá trị 1 (đúng)

* Thứ tự ưu tiên giữa các toán tử:



15.2.4 Phép toán trên bit (bitwise)

- & : và (AND)
- | : hoặc (OR)
- ^ : hoặc loại trừ (XOR)
- >> : dịch phải
- << : dịch trái
- ~ : đảo

Bit a	Bit b	~a	a & b	a b	a ^ b
0	0	1	0	0	0
0	1	1	0	1	1
1	0	0	0	1	1
1	1	0	1	1	0

Ví dụ 5:

- a = 13 đổi ra hệ nhị phân 1101
 - b = 10 đổi ra hệ nhị phân 1010
- | | | |
|--------|--------|---------------------------|
| 1101 | 1101 | 1101 |
| & 1010 | 1010 | ^ 1010 |
| = 1000 | = 1111 | = 0111 |
| = 8 | = 15 | = 7 (dạng thập phân) |

$a = 1235$ 0100 1101 0011
 $b = 465$ 0001 1101 0001
 \quad 0100 1101 0011 0100 1101 0011 0100 1101 0011
 $\&$ 0001 1101 0001 $|$ 0001 1101 0001 \wedge 0001 1101 0001
 $=$ 0000 1101 0001 $=$ 0101 1101 0011 $=$ 0101 0000 0010
 $=$ 209 $=$ 1491 $=$ 1282 (dạng thập phân)

15.2.5 Các phép toán khác

1. Phép toán gán

Phép gán là thay thế giá trị hiện tại của biến bằng một giá trị mới.
 Các phép gán: =, +=, *=, /=, %=, <<=, >>=, &=, |=, ^=.

Ví dụ 6: ta có giá trị $i = 3$

$i = i + 3$ 6
 $i += 3$ 6 $= i + 3$
 $i *= 3$ 9 $= i * 3$

2. Phép toán tăng, giảm: ++, --

Toán tử ++ sẽ cộng thêm 1 vào toán hạng của nó, toán tử -- sẽ trừ đi 1.

Ví dụ 7: ta có giá trị $n = 6$

+ Sau phép tính ++n hoặc n++, ta có $n = 7$.

+ Sau phép tính --n hoặc n--, ta có $n = 5$.

* Sự khác nhau giữa ++n và n++, --n và n--

+ Sau phép tính $x = ++n + 2$, ta có $x = 9$. (n tăng 1 cộng với 2 rồi gán cho x)

+ Sau phép tính $x = n++ + 2$, ta có $x = 8$. (n cộng với 2 gán cho x rồi mới tăng 1)

15.2.6 Độ ưu tiên của các phép toán

Độ ưu tiên	Các phép toán	Trình tự kết hợp
1	() [] ->	Trái sang phải
2	! ~ & * - ++ -- (type) sizeof	Phải sang trái
3	* / %	Trái sang phải
4	+ -	Trái sang phải
5	<< >>	Trái sang phải
6	< <= > >=	Trái sang phải
7	== !=	Trái sang phải
8	&	Trái sang phải
9	^	Trái sang phải
10		Trái sang phải
11	&&	Trái sang phải
12		Trái sang phải
13	? :	Phải sang trái
14	= += -= *= /= %= <<= >>= &= ^=	Phải sang trái
15	,	Trái sang phải

Lưu ý:

- Phép đảo (-) ở dòng 2, phép trừ (-) ở dòng 4
- Phép lấy địa chỉ (&) ở dòng 2, phép AND bit (&) ở dòng 8
- Phép lấy đối tượng con trỏ (*) ở dòng 2, phép nhân (*) ở dòng 3.

15.3 Bài tập

1. Giả sử a, b, c là biến kiểu int với a = 8, b = 3 và c = 5. Xác định giá trị các biểu thức sau:

$a + b + c$	$a \% c * 2$	$a * (a \% b)$
$a / b - c$	$2 * b + 3 * (a - c)$	$a * (b + (c - 4 * 3))$

$a + c / a$		$c * (b / a)$		$5 * a - 6 / b$	
$a \% b$		$(a * b) \% c$		$5 \% b \% c$	

2. Giả sử x, y, z là biến kiểu float với x = 8.8, y = 3.5 và z = 5.2. Xác định giá trị các biểu thức sau:

$x + y + z$		$z / (y + x)$		$x / y - z * y$	
$5 * y + 6 * (x - z)$		$(z / y) + x$		$2.5 * x / z - (y + 6)$	
x / z		$2 * y / 3 * z$		$5 * 6 / ((x + y) / z)$	
$x \% z$		$2 * y / (3 * z)$		$x / y * (6 + ((z - y) + 3.4))$	

3. Cho chương trình C với các khai báo và khởi tạo các biến như sau:

```
int i = 8, j = 5;
float x = 0.005, y = -0.01;
char c = 'c', d = 'd';
```

Hãy xác định giá trị trả về của các biểu thức sau:

$(3 * i - 2 * j) \% (4 * d - c)$		$c < d$	
$2 * ((i / 4) + (6 * (j - 3)) \% (i + j - 4))$		$x >= 0$	
$(i - 7 * j) \% (c + 3 * d) / (x - y)$		$x < y$	
$-(i + j) * -1$		$j != 6$	
$++i$		$c == 99$	
$i++$		$d != 100$	
$i++ + 5$		$5 * (i + j + 1) > 'd'$	
$++i + 5$		$(3 * x + y) == 0$	
$j--$		$2 * x + (y == 0)$	
$--j$		$!(i < j)$	
$j-- + i$		$!(d == 100)$	
$--j - -5$		$!(x < 0)$	
$++x$		$(i > 0) \&\& (j < 6)$	
$y--$		$(i > 0) !! (j < 5)$	
$i >= j$		$(x > y) \&\& (i > 0) \parallel (j < 5)$	

4. Cho chương trình có các khai báo biến và khởi tạo như sau:

```
int i = 8, j = 5, k;
float x = 0.005, y = -0.01, z;
char a, b, c = 'c', d = 'd';
```

Xác định giá trị các biểu thức gán sau:

$k = (i + j * 4)$		$z = i / j$		$i \% = j$	
$x = (x + y * 1.2)$		$a = b = d$		$i += (j - 3)$	
$i = j$		$y -= x$		$k = (j == 5) ? i : j$	
$k = (x + y)$		$x *= 2$		$k = (j > 5) ? i : j$	
$k = c$		$i /= j$		$i += j * = i /= 2$	
$i = j = 1.1$		$i += 2$		$a = (c < d) ? c : d$	
$z = k = x$		$z = (x >= 0) ? x : 0$		$i -= (j > 0) ? j : 0$	
$k = z = x$		$z = (y >= 0) ? y : 0$		$i = (i * 9 * (3 + (8 * j / 3)))$	

Bài 16 :

MỘT SỐ HÀM CHUẨN THƯỜNG DÙNG

16.1 Các hàm chuyển đổi dữ liệu

16.1.1 atof

double atof(const char *s);

☞ Phải khai báo **math.h** hoặc **stdlib.h**

Chuyển đổi 1 chuỗi sang giá trị double.

Ví dụ: float f;
char *str = "12345.67";
f = atof(str);
Kết quả f = 12345.67;

16.1.2 atoi

int atoi(const char *s);

☞ Phải khai báo **stdlib.h**

Chuyển đổi 1 chuỗi sang giá trị int.

Ví dụ: int i;
char *str = "12345.67";
i = atoi(str);
Kết quả i = 12345

16.1.3 itoa

char *itoa(int value, char *string, int radix); ☞ Phải khai báo **stdlib.h**

Chuyển đổi số nguyên value sang chuỗi string theo cơ số radix.

Ví dụ: int number = 12345;
char string[25];
itoa(number, string, 10); //chuyển đổi number sang chuỗi theo cơ số 10
Kết quả string = "12345";
itoa(number, string, 2); //chuyển đổi number sang chuỗi theo cơ số 2
Kết quả string = "11000000111001";

16.1.4 tolower

int tolower(int ch);

☞ Phải khai báo **cctype.h**

Đổi chữ hoa sang chữ thường.

Ví dụ: int len, i;
char *string = "THIS IS A STRING";
len = strlen(string);
for (i = 0; i < len; i++)
string[i] = tolower(string[i]); //đổi từ kí tự trong string thành chữ thường

16.1.5 toupper

int toupper(int ch);

☞ Phải khai báo **cctype.h**

Đổi chữ thường sang chữ hoa.

Ví dụ: int len, i;
char *string = "this is a string";
len = strlen(string);
for (i = 0; i < len; i++) string[i] = toupper(string[i]); //đổi từ kí tự trong string thành chữ thường

char *strcpy(char *dest, const char *src); ☞ Phải khai báo **string.h**

16.2 Chép chuỗi src Các hàm xử lý chuỗi ký tự

16.2.1 strcat

strcat(char *char *dest, const char *src); ☞ Phải khai báo **string.h**

Thêm chuỗi src vào sau chuỗi dest.

16.2.2 strcpy

vào dest.

Ví dụ:

```
char destination[25];
char *blank = " ", *c = "C++", *borland = "Borland";
strcpy(destination, borland); //chép chuỗi borland vào destination
strcat(destination, blank); //thêm chuỗi blank vào sau chuỗi destination
strcat(destination, c); //thêm chuỗi c vào sau chuỗi destination
```

16.2.3 strcmp

int *strcmp(const char *s1, const char *s2);

☞ Phải khai báo **string.h**

So sánh chuỗi s1 với chuỗi s2. Kết quả trả về:

- < 0 nếu s1 < s2
- = 0 nếu s1 = s2
- > 0 nếu s1 > s2

Ví dụ:

```
char *buf1 = "aaa", *buf2 = "bbb", *buf3 = "aaa";
strcmp(buf1, buf2); //kết quả trả về - 1
strcmp(buf1, buf3); //kết quả trả về 0
strcmp(buf2, buf3); //kết quả trả về 1
```

16.2.4 strcmpi

int *strcmp(const char *s1, const char *s2); ☞ Phải khai báo **string.h**

So sánh chuỗi s1 với chuỗi s2 không phân biệt chữ hoa, chữ thường. Kết quả trả về:

- < 0 nếu s1 < s2
- = 0 nếu s1 = s2
- > 0 nếu s1 > s2

Ví dụ:

```
char *buf1 = "aaa", *buf2 = "AAA";
strcmp(buf1, buf2); //kết quả trả về 0
```

16.2.5 strlwr

char *strlwr(char *s); ☞ Phải khai báo **string.h**

Chuyển chuỗi s sang chữ thường

Ví dụ:

```
char *s = "Borland C";
s = strlwr(s); //kết quả s = "borland c"
```

16.2.6strupr

char *strupr(char *s); ☞ Phải khai báo **string.h**

Chuyển chuỗi s sang chữ hoa

Ví dụ:

```
char *s = "Borland C";
```

Formatted: Bullets and Numbering

Formatted: Bullets and Numbering

```
s = strlwr(s);           //kết quả s = "BORLAND C"
```

16.2.7 strlen

```
int strlen(const char *s);    ☞ Phải khai báo string.h
```

Trả về độ dài chuỗi s.

Ví dụ: char *s = "Borland C";

```
int len_s;
len_s = strlen(s);       //kết quả len_s = 9
```

16.3 Các hàm toán học

16.3.1 abs

```
int abs(int x);             ☞ Phải khai báo stdlib.h
```

Cho giá trị tuyệt đối của số nguyên x.

Ví dụ: int num = -123;

```
num = abs(num);         //kết quả num = 123
```

16.3.2 labs

```
long int labs(long int x); ☞ Phải khai báo stdlib.h
```

Cho giá trị tuyệt đối của số nguyên dài x.

Ví dụ: int num = -12345678L;

```
num = labs(num);       //kết quả num = 12345678
```

16.3.3 rand

```
int rand(void);           ☞ Phải khai báo stdlib.h
```

Cho 1 giá trị ngẫu nhiên từ 0 đến 32767

Ví dụ: int num;

```
randomize();           //dùng hàm này để khởi đầu bộ số ngẫu nhiên
num = rand();          //kết quả num = 1 con số trong khoảng 0..32767
```

16.3.4 random

```
int random(int num);      ☞ Phải khai báo stdlib.h
```

Cho 1 giá trị ngẫu nhiên từ 0 đến 32767

Ví dụ: int n;

```
randomize();
n = random(100);       //kết quả n = 1 con số trong khoảng 0..99
```

16.3.5 pow

```
double pow(double x, double y); ☞ Phải khai báo math.h
```

Tính x mũ y

Ví dụ: double x = 2.0, y = 3.0, z;

```
z = pow(x, y);         //kết quả z = 8.0
```

16.3.6 sqrt

```
double sqrt(double x);    ☞ Phải khai báo math.h
```

Tính căn bậc 2 của x.

Ví dụ: double x = 4.0, y;

```
y = sqrt(x);          //kết quả y = 2.0
```


16.4 Các hàm xử lý file

16.4.1 rewind

void rewind(FILE *stream);

☞ Phải khai báo **stdio.h**

Đưa con trỏ về đầu file.

16.4.2 ftell

long ftell(FILE *stream);

☞ Phải khai báo **stdio.h**

Trả về vị trí con trỏ file hiện tại.

16.4.3 fseek

int fseek(FILE *stream, long offset, int whence); ☞ Phải khai báo **stdio.h**

Di chuyển con trỏ file đến vị trí mong muốn

■ **long offset**: chỉ ra số byte kể từ vị trí trước đó đến vị trí bắt đầu đọc

■ **int whence**: chỉ ra điểm xuất phát để tính offset gồm các giá trị sau: SEEK_SET (đầu tập tin), SEEK_CUR (tại vị trí con trỏ hiện hành), SEEK_END (cuối tập tin).

Formatted: Bullets and Numbering



MỤC LỤC

BÀI 1 :	NGÔN NGỮ LẬP TRÌNH & PHƯƠNG PHÁP LẬP TRÌNH	7
1.1	Mục tiêu	7
1.2	Lý thuyết.....	7
1.2.1	Ngôn ngữ lập trình (Programming Language)	7
1.2.1.1	Thuật giải (Algorithm)	7
1.2.1.2	Chương trình (Program)	7
1.2.1.3	Ngôn ngữ lập trình (Programming language)	8
1.2.2	Các bước lập trình	8
1.2.3	Kỹ thuật lập trình	8
1.2.3.1	I-P-O Cycle (Input-Process-Output Cycle) (Quy trình nhập-xử lý-xuất)	8
1.2.3.2	Sử dụng lưu đồ (Flowchart).....	9
BÀI 2 :	LÀM QUEN LẬP TRÌNH C QUA CÁC VÍ DỤ ĐƠN GIẢN	12
2.1	Mục tiêu	12
2.2	Nội dung.....	12
2.2.1	Khởi động và thoát BorlandC.....	12
2.2.1.1	Khởi động	12
2.2.1.2	Thoát.....	13
2.2.2	Các ví dụ đơn giản	13
2.2.2.1	Ví dụ 1.....	13
2.2.2.2	Ví dụ 2.....	15
2.2.2.3	Ví dụ 3.....	16
2.2.2.4	Ví dụ 4.....	16
BÀI 3 :	CÁC THÀNH PHẦN TRONG NGÔN NGỮ C	18
3.1	Mục tiêu	18
3.2	Nội dung.....	18
3.2.1	Từ khóa	18
3.2.2	Tên.....	18
3.2.3	Kiểu dữ liệu	18
3.2.4	Ghi chú.....	19
3.2.5	Khai báo biến	19
3.2.5.1	Tên biến	19
3.2.5.2	Khai báo biến	19
3.2.5.3	Vừa khai báo vừa khởi gán.....	20
3.2.5.4	Phạm vi của biến.....	20
BÀI 4 :	NHẬP / XUẤT DỮ LIỆU	21
4.1	Mục tiêu	21
4.2	Nội dung.....	21

4.2.1	Hàm printf.....	21
4.2.2	Hàm scanf.....	24
4.3	Bài tập.....	25
BÀI 5 :	CẤU TRÚC RỄ NHÁNH CÓ ĐIỀU KIỆN	26
5.1	Mục tiêu	26
5.2	Nội dung.....	26
5.2.1	Lệnh và khối lệnh.....	26
5.2.1.1	Lệnh	26
5.2.1.2	Khối lệnh	26
5.2.2	Lệnh if.....	26
5.2.2.1	Dạng 1 (if thiếu)	26
5.2.2.2	Dạng 2 (if đủ)	30
5.2.2.3	Cấu trúc else if.....	33
5.2.2.4	Cấu trúc if lồng.....	37
5.2.3	Lệnh switch.....	41
5.2.3.1	Cấu trúc switch...case (switch thiếu).....	41
5.2.3.2	Cấu trúc switch...case...default (switch đủ).....	44
5.2.3.3	Cấu trúc switch lồng.....	46
5.3	Bài tập.....	48
5.3.1	Sử dụng lệnh if	48
5.3.2	Sử dụng lệnh switch	49
5.4	Bài tập làm thêm.....	49
BÀI 6 :	CẤU TRÚC VÒNG LẶP	51
6.1	Mục tiêu	51
6.2	Nội dung.....	51
6.2.1	Lệnh for.....	51
6.2.2	Lệnh break.....	56
6.2.3	Lệnh continue.....	56
6.2.4	Lệnh while.....	56
6.2.5	Lệnh do...while	58
6.2.6	Vòng lặp lồng nhau	60
6.2.7	So sánh sự khác nhau của các vòng lặp	61
6.3	Bài tập.....	62
BÀI 7 :	HÀM.....	65
7.1	Mục tiêu	65
7.2	Nội dung.....	65
7.2.1	Các ví dụ về hàm	65
7.2.2	Tham số dạng tham biến và tham trị.....	68

7.2.3	Sử dụng biến toàn cục	69
7.2.4	Dùng dẫn hướng #define	71
7.3	Bài tập.....	71
BÀI 8 : MẢNG VÀ CHUỖI		72
8.1	Mục tiêu	72
8.2	Nội dung.....	72
8.2.1	Mảng.....	72
8.2.1.1	Cách khai báo mảng.....	72
8.2.1.2	Tham chiếu đến từng phần tử mảng	72
8.2.1.3	Nhập dữ liệu cho mảng	73
8.2.1.4	Đọc dữ liệu từ mảng	73
8.2.1.5	Sử dụng biến kiểu khác.....	74
8.2.1.6	Kỹ thuật Sentinel.....	74
8.2.1.7	Khởi tạo mảng.....	75
8.2.1.8	Khởi tạo mảng không bao hàm kích thước.....	76
8.2.1.9	Mảng nhiều chiều	76
8.2.1.10	Tham chiếu đến từng phần tử mảng 2 chiều	76
8.2.1.11	Nhập dữ liệu cho mảng 2 chiều	77
8.2.1.12	Đọc dữ liệu từ mảng 2 chiều	77
8.2.1.13	Sử dụng biến kiểu khác trong mảng 2 chiều.....	78
8.2.1.14	Khởi tạo mảng 2 chiều	78
8.2.1.15	Dùng mảng 1 chiều làm tham số cho hàm	79
8.2.1.16	Dùng mảng 2 chiều làm tham số cho hàm	82
8.2.2	Chuỗi.....	84
8.2.2.1	Cách khai báo chuỗi.....	84
8.2.2.2	Hàm nhập (gets), xuất (puts) chuỗi.....	85
8.2.2.3	Khởi tạo chuỗi.....	86
8.2.2.4	Mảng chuỗi.....	86
8.3	Bài tập.....	87
BÀI 9 : CON TRỎ		90
9.1	Mục tiêu	90
9.2	Nội dung.....	90
9.2.1	Con trỏ?	90
9.2.2	Khái báo biến con trỏ	90
9.2.3	Truyền địa chỉ sang hàm	91
9.2.4	Con trỏ và mảng.....	92
9.2.5	Con trỏ trỏ đến mảng trong hàm	92
9.2.6	Con trỏ và chuỗi.....	93
9.2.7	Khởi tạo mảng con trỏ trỏ đến chuỗi	94
9.2.8	Xử lý con trỏ trỏ đến chuỗi	95
9.2.9	Con trỏ trỏ đến con trỏ.....	97
9.3	Bài tập.....	98

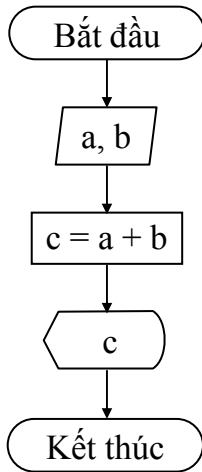
BÀI 10 : CÁC KIỂU DỮ LIỆU TỰ TẠO	99
10.1 Mục tiêu	99
10.2 Nội dung.....	99
10.2.1 Structure	99
10.2.1.1 Khai báo kiểu structure	99
10.2.1.2 Cách khai báo biến có kiểu structure	99
10.2.1.3 Tham chiếu các phần tử trong structure.....	99
10.2.1.4 Khởi tạo structure	101
10.2.1.5 Structure lồng nhau.....	102
10.2.1.6 Truyền structure sang hàm	103
10.2.2 Enum	105
10.2.2.1 Định nghĩa kiểu enum	105
10.2.2.2 Cách khai báo biến có kiểu enum	106
10.2.2.3 Sử dụng enum trong chương trình	106
10.3 Bài tập.....	108
BÀI 11 : TẬP TIN	109
11.1 Mục tiêu	109
11.2 Nội dung.....	109
11.2.1 Ví dụ ghi, đọc số nguyên.....	109
11.2.2 Ghi, đọc mảng	110
11.2.3 Ghi, đọc structure	111
11.2.4 Các mode khác để mở tập tin	112
11.2.5 Một số hàm thao tác trên file khác.....	112
11.3 Bài tập.....	113
BÀI 12 : ĐỆ QUY	114
12.1 Mục tiêu	114
12.2 Nội dung.....	114
12.3 Bài tập.....	117
BÀI 13 : TRÌNH SOẠN THẢO CỦA BORLAND C	118
13.1 Mở tập tin soạn thảo mới.....	118
13.2 Lưu tập tin.....	118
13.2.1 Nếu là tập tin soạn thảo mới chưa lưu	118
13.2.2 Nếu là tập tin đã lưu ít nhất 1 lần hoặc được mở bằng lệnh Open:	118
13.3 Mở tập tin	119
13.4 Các phím, tổ hợp phím thường dùng.....	119
13.4.1 Các phím di chuyển con trỏ	119

13.4.2 Các phím thao tác trên khối.....	120
13.4.3 Các thao tác xóa	120
13.4.4 Các thao tác copy, di chuyển.....	120
13.4.5 Các thao tác khác.....	120
13.5 Ghi một khối ra đĩa	121
13.6 Chèn nội dung file từ đĩa vào vị trí con trỏ	121
13.7 Tìm kiếm văn bản trong nội dung soạn thảo	121
13.8 Tìm và thay thế văn bản trong nội dung soạn thảo.....	121
13.9 Sửa lỗi cú pháp.....	122
13.10 Chạy từng bước	122
13.11 Sử dụng Help (Giúp đỡ).....	122
BÀI 14 : CÁC HỆ ĐẾM	124
14.1 Khái niệm	124
14.2 Quy tắc.....	124
14.3 Chuyển đổi giữa các hệ	125
14.3.1 Chuyển đổi giữa hệ 2 và hệ 10	125
14.3.2 Chuyển đổi giữa hệ 8 và hệ 10	126
14.3.3 Chuyển đổi giữa hệ 16 và hệ 10	126
14.3.4 Chuyển đổi giữa hệ 2 và hệ 16	127
BÀI 15 : BIỂU THỨC VÀ PHÉP TOÁN	128
15.1 Biểu thức.....	128
15.2 Phép toán.....	128
15.2.1 Phép toán số học.....	128
15.2.2 Phép quan hệ	128
15.2.3 Phép toán luận lý.....	129
15.2.4 Phép toán trên bit (bitwise).....	129
15.2.5 Các phép toán khác.....	130
15.2.6 Độ ưu tiên của các phép toán.....	130
15.3 Bài tập.....	130
BÀI 16 : MỘT SỐ HÀM CHUẨN THƯỜNG DÙNG.....	132
16.1 Các hàm chuyển đổi dữ liệu	132
16.1.1 atof.....	132
16.1.2 atoi.....	132
16.1.3 itoa	132
16.1.4 tolower.....	132

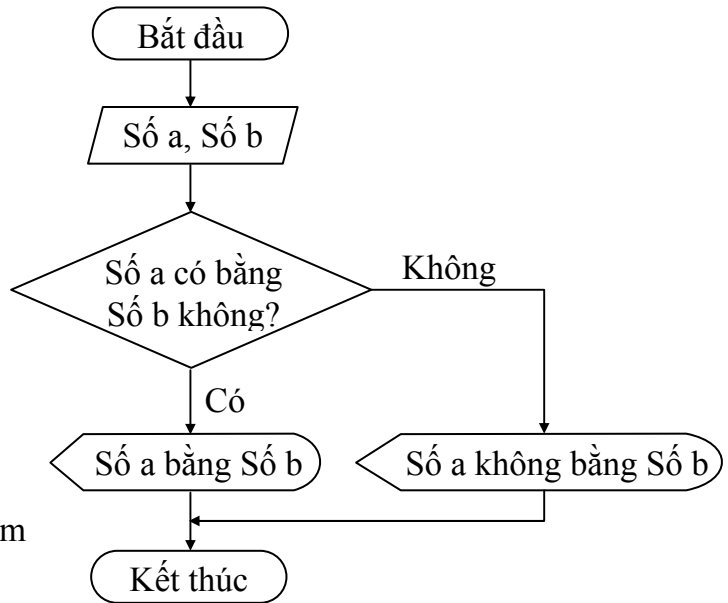
16.1.5 toupper.....	132
16.2 Các hàm xử lý chuỗi ký tự.....	133
16.2.1 strcat.....	133
16.2.2 strcpy.....	133
16.2.3 strcmp.....	133
16.2.4 strcmpi.....	133
16.2.5 strlwr.....	133
16.2.6strupr.....	133
16.2.7 strlen.....	134
16.3 Các hàm toán học.....	134
16.3.1 abs.....	134
16.3.2 labs.....	134
16.3.3 rand.....	134
16.3.4 random.....	134
16.3.5 pow.....	134
16.3.6 sqrt.....	134
16.4 Các hàm xử lý file.....	135
16.4.1 rewind.....	135
16.4.2 ftell.....	135
16.4.3 fseek.....	135



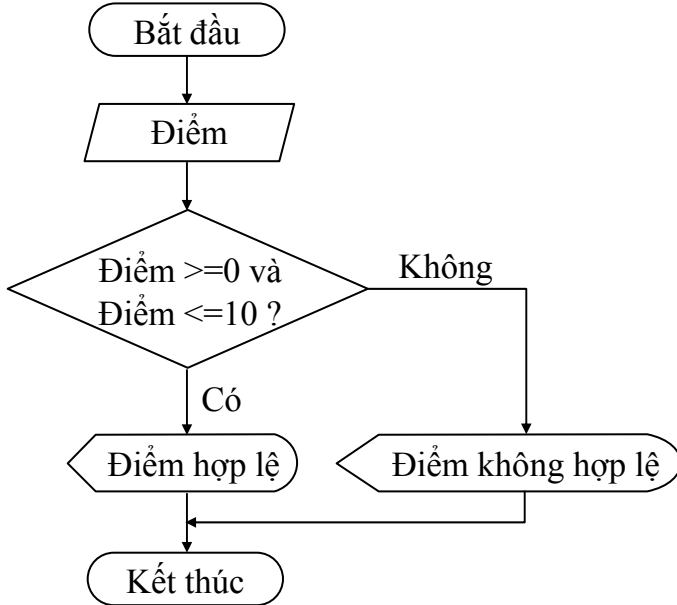
Ví dụ 9: Cộng 2 số



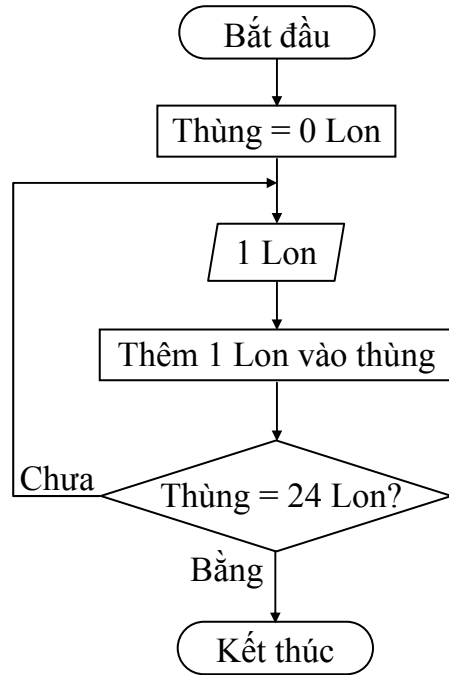
Ví dụ 10: so sánh 2 số



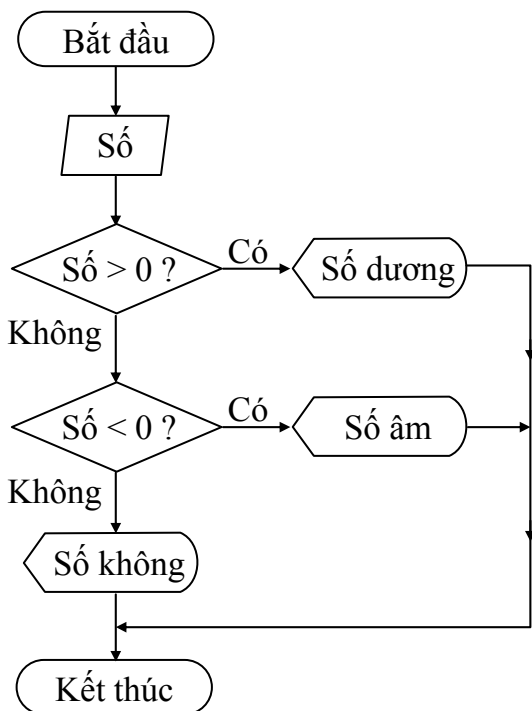
Ví dụ 11: Kiểm tra tính hợp lệ của điểm



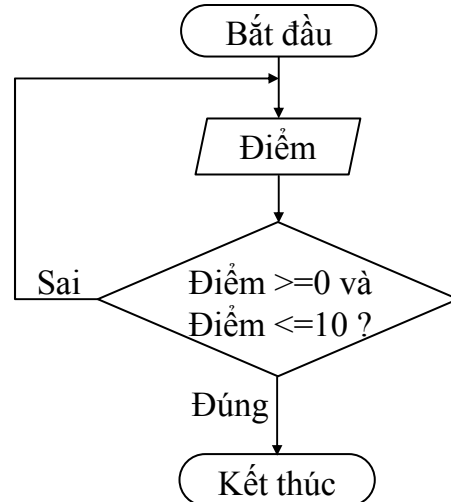
Ví dụ 12: Xếp lon vào thùng



Ví dụ 13: Kiểm tra loại số



Ví dụ 14: Kiểm tra tính hợp lệ của điểm



 Bài tập

Vẽ lưu đồ cho các chương trình sau:

1. Đổi từ tiền VND sang tiền USD.
2. Tính điểm trung bình của học sinh gồm các môn Toán, Lý, Hóa.
3. Giải phương trình bậc 2: $ax^2 + bx + c = 0$
4. Đổi từ độ sang radian và đổi từ radian sang độ
(công thức $\alpha/\pi = a/180$, với α : radian, a: độ)
5. Kiểm tra 2 số a, b giống nhau hay khác nhau.





Bạn nhập đoạn chương trình trên vào máy. Dịch, chạy và quan sát kết quả.

Ctrl – F9: Dịch và chạy chương trình. Alt – F5: Xem màn hình kết quả.



Sau khi bạn nhập xong đoạn chương trình vào máy. Bạn Ấn và giữ phím Ctrl, gõ F9 để dịch và chạy chương trình. Khi đó bạn thấy chương trình chớp rất nhanh và không thấy kết quả gì cả. Bạn Ấn và giữ phím Alt, gõ F5 để xem kết quả, khi xem xong, bạn ấn phím bất kỳ để quay về màn hình soạn thảo chương trình.



Bây giờ bạn sửa lại dòng thứ 6 bằng câu lệnh `printf("Bai hoc C dau tien.\n");`, sau đó dịch và chạy lại chương trình, quan sát kết quả.

👉 Kết quả in ra màn hình

Bai hoc C dau tien.

Ở dòng bạn vừa sửa có thêm `\n`, `\n` là ký hiệu xuống dòng sử dụng trong lệnh `printf`. Sau đây là một số ký hiệu khác.

+ Các kí tự điều khiển:

- `\n` : Nhảy xuống dòng kế tiếp canh về cột đầu tiên.
- `\t` : Canh cột tab ngang.
- `\r` : Nhảy về đầu hàng, không xuống hàng.
- `\a` : Tiếng kêu bip.

+ Các kí tự đặc biệt:

- `\\` : In ra dấu `\`
- `\"` : In ra dấu `"`
- `'` : In ra dấu `'`



Bây giờ bạn sửa lại dòng thứ 6 bằng câu lệnh `printf("\tBai hoc C dau tien.\a\n");`, sau đó dịch và chạy lại chương trình, quan sát kết quả.

👉 Kết quả in ra màn hình

Bai hoc C dau tien.

Khi chạy chương trình bạn nghe tiếng bip phát ra từ loa.



Mỗi khi chạy chương trình bạn thấy rất bất tiện trong việc xem kết quả phải ấn tổ hợp phím `Alt – F5`. Để khắc phục tình trạng này bạn sửa lại chương trình như sau:

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	<code>/* Chuong trinh in ra cau bai hoc C dau tien */</code>
2	<code>#include <stdio.h></code>
3	<code>#include <conio.h></code>
4	
5	<code>void main(void)</code>
6	<code>{</code>
7	<code>printf("\t\tBai hoc C \rdau tien.\n");</code>
8	<code>getch();</code>
9	<code>}</code>
	F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

 **Kết quả in ra màn hình**

Nhap vao ban kinh hinh tron: 1

Dien tich hinh tron: 6.28

—

■ Dòng thứ 5: **`#define PI 3.14`**, dùng chỉ thị `define` để định nghĩa hằng số `PI` có giá trị 3.14. Trước `define` phải có dấu `#` và cuối dòng không có dấu chấm phẩy.

■ Dòng thứ 12: **`printf("Dien tich hinh tron: %.2f\n", 2*PI*fR);`**. Hàm này có 2 đối mục. Đối mục thứ nhất là một chuỗi định dạng có chứa chuỗi văn bản ***Dien tich hinh tron:*** và **`%.2f`** (ký hiệu khai báo chuyển đổi dạng thức) cho biết dạng số chấm động sẽ được in ra, trong đó **`.2`** nghĩa là in ra với 2 số lẻ. Đối mục thứ 2 là biểu thức hằng **`2*PI*fR`**;



Bạn nhập đoạn chương trình trên vào máy. Dịch, chạy và quan sát kết quả.



Kết quả in ra màn hình

Bai hoc C dau tien.

```

Ví dụ 2: printf("Ma dinh dang \\\" in ra dau \".\n");

```

Kết quả in ra màn hình

Ma dinh dang \" in ra dau \".

Ví dụ 3: giả sử biến i có giá trị = 5

```

printf("So ban vua nhap la: %d .\n", i);

```

Kết quả in ra màn hình

So ban vua nhap la: 5.

Ví dụ 4: giả sử biến a có giá trị = 7 và b có giá trị = 4

```

printf("Tong cua 2 so %d va %d la %d .\n", a, b, a+b);

```

Kết quả in ra màn hình

Tong cua 2 so 7 va 4 la 11.

Ví dụ 5: sửa lại ví dụ 4

```

printf("Tong cua 2 so %5d va %3d la %1d .\n", a, b, a+b);

```

☞ Kết quả in ra màn hình

Tong cua 2 so <u>7</u> va <u>4</u> la <u>11</u> .
—

→ 2 kí tự (mặc dù định dạng là 1)
 → 3 kí tự
 → 5 kí tự

Ví dụ 6: sửa lại ví dụ 5

```
printf("Tong cua 2 so %-5d va %-3d la %-1d . \n", a, b, a+b);
```

☞ Kết quả in ra màn hình

Tong cua 2 so <u>7</u> va <u>4</u> la <u>11</u> .
—

→ 2 kí tự (mặc dù định dạng là 1)
 → 3 kí tự
 → 5 kí tự

☞ Dấu trừ trước bề rộng trường sẽ kéo kết quả sang trái

Ví dụ 7: sửa lại ví dụ 4

```
printf("Tong cua 2 so %02d va %02d la %04d . \n", a, b, a+b);
```

☞ Kết quả in ra màn hình

Tong cua 2 so <u>07</u> va <u>04</u> la <u>0011</u> .
—

→ thêm 2 số 0 trước -> đủ 4 kí tự
 → thêm 1 số 0 trước -> đủ 2 kí tự
 → thêm 1 số 0 trước -> đủ 2 kí tự

Ví dụ 8: giả sử int a = 6, b = 1234, c = 62

```
printf("%7d%7d%7d.\n", a, b, c);
printf("%7d%7d%7d.\n", 165, 2, 965);
```

☞ Kết quả in ra màn hình

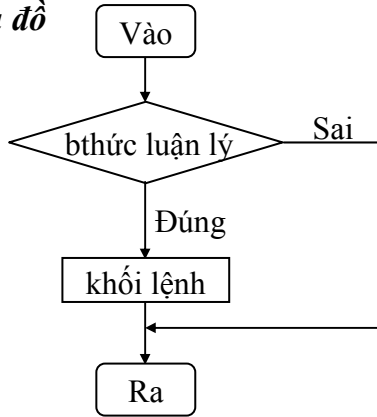
6 1234 62	Số canh về bên phải bề rộng trường.
165 2 965	
—	

```
printf("%-7d%-7d%-7d.\n", a, b, c);
printf("%-7d%-7d%-7d.\n", 165, 2, 965);
```

☞ Kết quả in ra màn hình

6 1234 62	Số canh về bên trái bề rộng trường.
165 2 965	
—	

• Lưu đồ



☞ nếu **biểu thức luận lý** đúng thì thực hiện khối lệnh và thoát khỏi if, ngược lại không làm gì cả và thoát khỏi if.

☞ Nếu **khối lệnh** bao gồm từ 2 lệnh trở lên thì phải đặt trong dấu { }

Diễn giải:

+ Khối lệnh là một lệnh ta viết lệnh if như sau:

```
if (biểu thức luận lý)
    lệnh;
```

+ Khối lệnh bao gồm nhiều lệnh: lệnh 1, lệnh 2..., ta viết lệnh if như sau:

```
if (biểu thức luận lý)
{
    lệnh 1;
    lệnh 2;
    ...
}
```

☞ Không đặt dấu chấm phẩy sau câu lệnh if.

Ví dụ: `if(biểu thức luận lý);`

→ trình biên dịch không báo lỗi nhưng khối lệnh không được thực hiện cho dù điều kiện đúng hay sai.

Ví dụ 3: Viết chương trình nhập vào 2 số nguyên a, b. Tìm và in ra số lớn nhất.

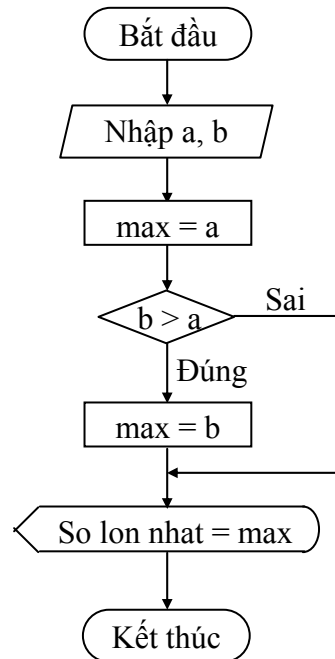
a. Phác họa lời giải

Trước tiên ta cho giá trị *a* là **giá trị lớn nhất bằng cách gán a cho max** (max là biến được khai báo cùng kiểu dữ liệu với a, b). Sau đó so sánh b với a, **nếu b lớn hơn a ta gán b cho max** và cuối cùng ta **được kết quả max là giá trị lớn nhất**.

b. Mô tả quy trình xử lý (giải thuật)

Ngôn ngữ tự nhiên	Ngôn ngữ C
- Khai báo 3 biến a, b, max kiểu số nguyên	- int ia, ib, imax;
- Nhập vào giá trị a	- printf("Nhap vao so a: "); scanf("%d", &ia);
- Nhập vào giá trị b	- printf("Nhap vao so b: "); scanf("%d", &ib);
- Gán a cho max	- imax = ia;
- Nếu b > a thì gán b cho max	- if (ib > ia) imax = ib;
- In ra kết quả max	- printf("So lon nhat = %d.\n", imax);

☞ **Biểu thức luận lý phải đặt trong cặp dấu ().** `if ib > ia` → báo lỗi

c. Mô tả bằng lưu đồ**d. Viết chương trình**

<pre> File Edit Search Run Compile Debug Project Option Window Help /* Chương trình tìm số lớn nhất từ 2 số nguyên a, b */ #include <stdio.h> #include <conio.h> void main(void) { int ia, ib, imax; printf("Nhập vào số a: "); scanf("%d", &ia); printf("Nhập vào số b: "); scanf("%d", &ib); imax = ia; if (ib > ia) imax = ib; printf("Số lớn nhất = %d.\n", imax); getch(); } </pre>
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

☞ Kết quả in ra màn hình

Nhập vào số a : 10 Nhập vào số b : 8 Số lớn nhất = 10.	Cho chạy lại chương trình và thử lại với: a = 7, b = 9 a = 5, b = 5 Quan sát và nhận xét kết quả
--	---

Ví dụ 4: Viết chương trình nhập vào 2 số nguyên a, b. Nếu a lớn hơn b thì hoán đổi giá trị a và b, ngược lại không hoán đổi. In ra giá trị a, b.

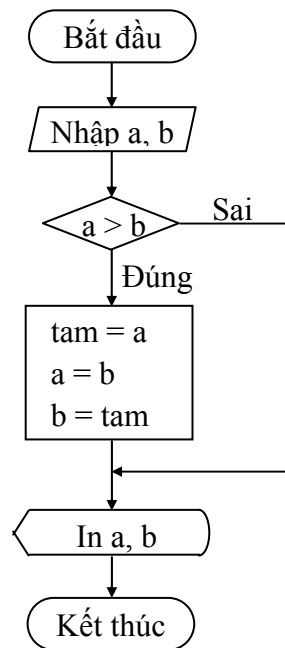
a. Phác họa lời giải

Nếu giá trị a lớn hơn giá trị b , bạn phải hoán chuyển 2 giá trị này cho nhau (nghĩa là a sẽ mang giá trị b và b mang giá trị a) bằng cách đem **giá trị a gởi (gán) cho biến tam** (biến tam được khai báo theo kiểu dữ liệu của a, b), kế đến bạn **gán giá trị b cho a** và cuối cùng bạn **gán giá trị tam cho b** , rồi in ra a, b .

b. Mô tả quy trình thực hiện (giải thuật)

Ngôn ngữ tự nhiên	Ngôn ngữ C
<ul style="list-style-type: none"> - Khai báo 3 biến a, b, tam kiểu số nguyên - Nhập vào giá trị a - Nhập vào giá trị b - Nếu $a > b$ thì <ul style="list-style-type: none"> $tam = a;$ $a = b;$ $b = tam;$ - In ra a, b 	<ul style="list-style-type: none"> - <code>int ia, ib, itam;</code> - <code>printf("Nhap vao so a: ");</code> <code>scanf("%d", &ia);</code> - <code>printf("Nhap vao so b: ");</code> <code>scanf("%d", &ib);</code> - <code>if (ia > ib)</code> { <code>itam = ia;</code> <code>ia = ib;</code> <code>ib = itam;</code> } - <code>printf("%d, %d\n", ia, ib);</code>

c. Mô tả bằng lưu đồ



d. Viết chương trình

```

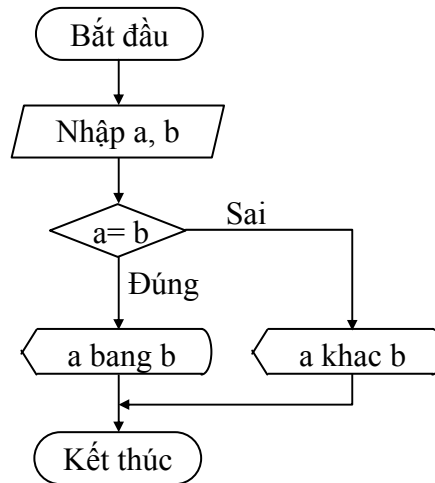
File Edit Search Run Compile Debug Project Option Window Help
/* Chuong trinh hoan vi 2 so a, b neu a > b */

#include <stdio.h>
#include <conio.h>

void main(void)
{
    int ia, ib, itam;
    printf("Nhap vao so a: ");
  
```


<ul style="list-style-type: none"> - Khai báo 2 biến a, b kiểu số nguyên - Nhập vào giá trị a - Nhập vào giá trị b - Nếu a = b thì in ra thông báo "a bằng b" Ngược lại (còn không thì) in ra thông báo "a khác b" 	<pre> - int ia, ib; - printf("Nhap vao so a: "); scanf("%d", &ia); - printf("Nhap vao so b: "); scanf("%d", &ib); - if (ia == ib) printf("a bang b\n"); else printf("a khac b\n"); </pre>
---	---

c. Mô tả bằng lưu đồ



d. Viết chương trình

```

File Edit Search Run Compile Debug Project Option Window Help

/* Chương trình in ra thông báo "a bằng b" nếu a = b, ngược lại in ra "a khác b" */

#include <stdio.h>
#include <conio.h>

void main(void)
{
    int ia, ib;
    printf("Nhap vao so a: ");
    scanf("%d", &ia);
    printf("Nhap vao so b: ");
    scanf("%d", &ib);
    if (ia == ib)
        printf("a bang b\n");
    else
        printf("a khac b\n");
    getch();
}

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu
    
```

👉 Kết quả in ra màn hình

<p>Nhap vao so a : 10 Nhap vao so b : 8 a khac b.</p>	<p>Cho chạy lại chương trình và thử lại với: a = 6, b = 6 a = 1, b = 5</p>
---	--

Quan sát và nhận xét kết quả

☞ Sau else không có dấu chấm phẩy.
 Ví dụ: `else; printf('a khác b\n');`
 → trình biên dịch không báo lỗi, lệnh `printf("a khác b\n");` không thuộc else

Ví dụ 6: Viết chương trình nhập vào kí tự c. Kiểm tra xem nếu kí tự nhập vào là kí tự thường trong khoảng từ 'a' đến 'z' thì đổi sang chữ in hoa và in ra, ngược lại in ra thông báo "Kí tự bạn vừa nhập là: c".

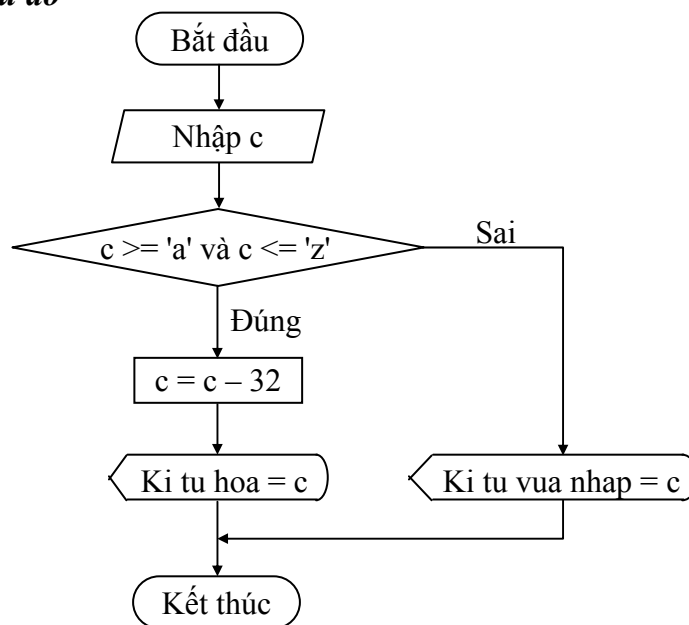
a. Phác họa lời giải

Trước tiên bạn phải kiểm tra xem nếu kí tự c thuộc khoảng 'a' và 'z' thì đổi kí tự c thành chữ in hoa bằng cách lấy kí tự c - 32 rồi gán lại cho chính nó (`c = c - 32`) (vì giữa kí tự thường và in hoa trong bảng mã ASCII cách nhau 32, ví dụ: A trong bảng mã ASCII là 65, B là 66..., còn a là 97, b là 98...), sau khi đổi xong bạn in kí tự c ra. Ngược lại, in câu thông báo "Kí tự bạn vừa nhập là: c".

b. Mô tả quy trình xử lý (giải thuật)

Ngôn ngữ tự nhiên	Ngôn ngữ C
- Khai báo biến c kiểu kí tự - Nhập vào kí tự c - Nếu <code>c >= a</code> và <code>c <= z</code> thì <code>c = c - 32</code> in c ra màn hình Ngược lại in ra thông báo " Kí tự bạn vừa nhập là: c "	<pre> - char c; - printf("Nhap vao 1 ki tu: "); scanf("%c", &c); - if (c >= 'a' && c <= 'z') { c = c - 32; printf("Ki tu hoa la: %c.\n", c); }; else printf("Ki tu ban vua nhap la: %c.\n", c); </pre>

c. Mô tả bằng lưu đồ



d. Viết chương trình

File Edit Search Run Compile Debug Project Option Window Help

Ví dụ 7: Viết chương trình nhập vào 2 số nguyên a, b. In ra thông báo "a lớn hơn b" nếu $a > b$, in ra thông báo "a nhỏ hơn b" nếu $a < b$, in ra thông báo "a bằng b" nếu $a = b$.

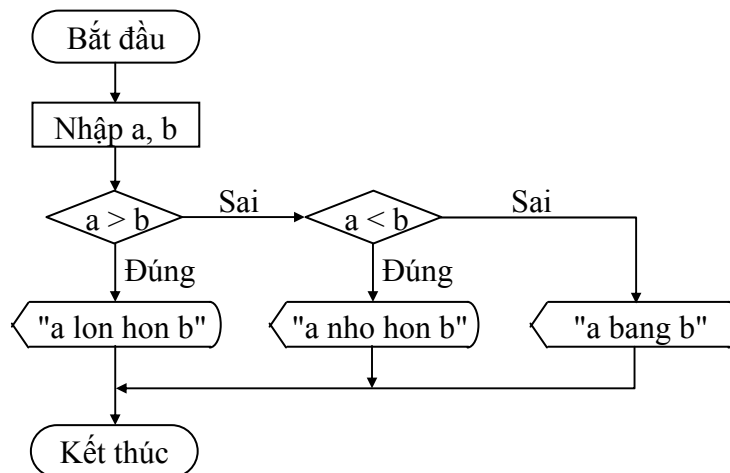
a. Phác họa lời giải

Trước tiên so sánh a với b. Nếu $a > b$ thì in ra thông báo "a lớn hơn b", ngược lại nếu $a < b$ thì in ra thông báo "a nhỏ hơn b", ngược với 2 trường hợp trên thì in ra thông báo "a bằng b".

b. Mô tả quy trình thực hiện (giải thuật)

Ngôn ngữ tự nhiên	Ngôn ngữ C
- Khai báo 2 biến a, b kiểu số nguyên - Nhập vào giá trị a - Nhập vào giá trị b - Nếu $a > b$ thì in ra thông báo "a lớn hơn b" Ngược lại Nếu $a < b$ thì in ra thông báo "a nhỏ hơn b" Ngược lại thì in ra thông báo "a bằng b"	<pre> - int ia, ib; - printf("Nhap vao so a: "); scanf("%d", &ia); - printf("Nhap vao so b: "); scanf("%d", &ib); - if (ia > ib) printf("a lon hon b.\n"); else if (ia < ib) printf("a nho hon b.\n"); else printf("a bang b.\n"); </pre>

c. Mô tả bằng lưu đồ



d. Viết chương trình

```

File Edit Search Run Compile Debug Project Option Window Help
/* Chuong trinh nhap vao 2 so nguyen a, b. In ra thong bao a > b, a < b, a = b */
#include <stdio.h>
#include <conio.h>

void main(void)

```

```

{
  int ia, ib;
  printf("Nhap vao so a: ");
  scanf("%d", &ia);
  printf("Nhap vao so b: ");
  scanf("%d", &ib);
  if (ia>ib)
    printf("a lon hon b.\n");
  else if (ia<ib)
    printf("a nho hon b.\n");
  else
    printf("a bang b.\n");
  getch();
}
    
```

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

☞ Kết quả in ra màn hình

Nhap vao so a : 5 Nhap vao so b : 7 a nho hon b	Cho chạy lại chương trình và thử lại với: a = 8, b = 4 a = 2, b = 2 Quan sát và nhận xét kết quả
---	---

Ví dụ 8: Viết chương trình nhập vào kí tự c. Kiểm tra xem nếu kí tự nhập vào là kí tự thường trong khoảng từ 'a' đến 'z' thì đổi sang chữ in hoa và in ra, nếu kí tự in hoa trong khoảng A đến Z thì đổi sang chữ thường và in ra, nếu kí tự là số từ 0 đến 9 thì in ra câu "Kí tự bạn vừa nhập là số ...(in ra kí tự c)", còn lại không phải 3 trường hợp trên in ra thông báo "Bạn đã nhập kí tự ...(in ra kí tự c)".

a. Phác họa lời giải

Nhập kí tự c vào, kiểm tra xem nếu kí tự c thuộc khoảng 'a' và 'z' đổi kí tự c thành chữ in hoa bằng cách lấy kí tự c – 32 rồi gán lại cho chính nó (c = c – 32) (vì giữa kí tự thường và in hoa trong bảng mã ASCII cách nhau 32, ví dụ: A trong bảng mã ASCII là 65, B là 66..., còn a là 97, b là 98...), sau khi đổi xong bạn in kí tự c ra. Ngược lại Nếu kí tự c thuộc khoảng 'A' và 'Z', đổi kí tự c thành chữ thường (theo cách ngược lại) và in ra. Ngược lại Nếu kí tự c thuộc khoảng '0' và '9' thì in ra thông báo "Kí tự bạn vừa nhập là số...". Ngược lại, in câu thông báo "Bạn đã nhập kí tự...".

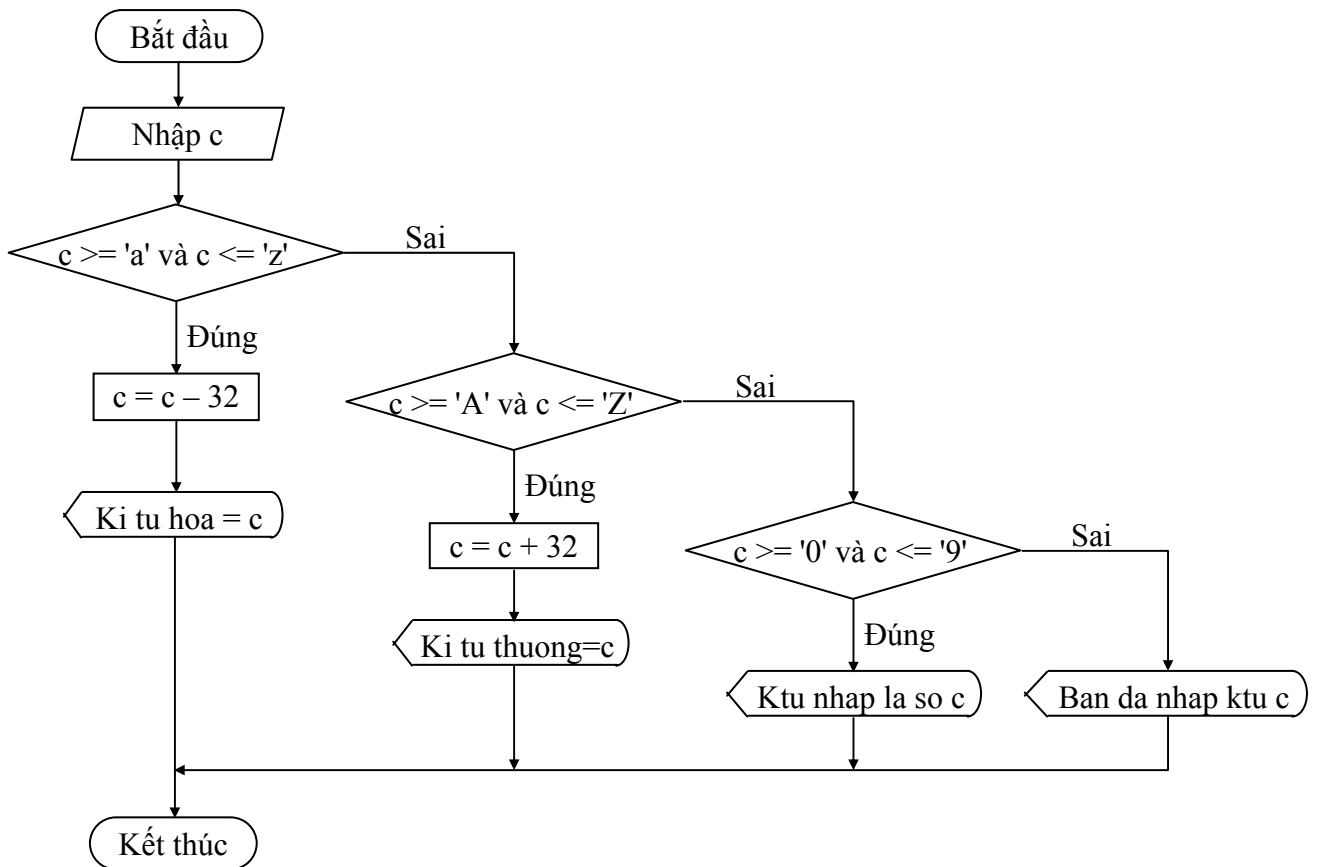
b. Mô tả quy trình xử lý (giải thuật)

Ngôn ngữ tự nhiên	Ngôn ngữ C
- Khai báo biến c kiểu kí tự	- char c;
- Nhập vào kí tự c	- printf("Nhap vao 1 ki tu: "); scanf("%c", &c);
- Nếu c >= a và c <= z thì c = c – 32 in c ra màn hình	- if (c >= 'a' && c <= 'z') { c = c – 32; printf("Ki tu hoa la: %c.\n", c); };
Ngược lại Nếu c >= A và c <= Z thì c = c + 32 in c ra màn hình	else if(c >= 'A' && c <= 'Z') { c = c + 32; printf("Ki tu thuong la: %c.\n", c);

Ngược lại Nếu $c \geq 0$ và $c \leq 9$ thì in thông báo "Kí tự bạn vừa nhập là số c" Ngược lại thì in thông báo "Bạn đã nhập kí tự c"	<pre>}; else if(c >= '0' && c <= '9') printf("Kí tự Ban vua nhap la so %c.\n", c); else printf("Ban da nhap ki tu %c.\n", c);</pre>
--	---

☞ Cũng như if, không đặt dấu chấm phẩy sau câu lệnh else if.
 Ví dụ: `else if(c >= 'A' && c <= 'Z');`
 → trình biên dịch không báo lỗi nhưng khối lệnh sau else if không được thực hiện.

c. Mô tả bằng lưu đồ



e. Viết chương trình

```

File Edit Search Run Compile Debug Project Option Window Help

/* Chương trình nhập vào kí tự c. Đòi ra hoa, thường */

#include <stdio.h>
#include <conio.h>

void main(void)
{
    char c;
    printf("Nhập vào 1 kí tự: ");
    scanf("%c", &c);
    if (c >= 'a' && c <= 'z')           //hoac if(c >= 97 && c <= 122)
    {
        c = c - 32;                     //doi thanh chu in hoa
    }
}
    
```

```

if (n > 0)
{
    if (a > b)
        x = a;
}
else
    x = b;
...
    
```

• **Lưu đồ**

Tương tự 3 dạng trên. Nhưng trong mỗi khối lệnh có thể có một (nhiều) cấu trúc if ở 3 dạng trên.

Ví dụ 10: Viết chương trình nhập vào điểm của một học sinh. In ra xếp loại học tập của học sinh đó. (Cách xếp loại. Nếu điểm ≥ 9 , Xuất sắc. Nếu điểm từ 8 đến cận 9, Giỏi. Nếu điểm từ 7 đến cận 8, Khá. Nếu điểm từ 6 đến cận 7, TBKhá. Nếu điểm từ 5 đến cận 6, TBình. Còn lại là Yếu).

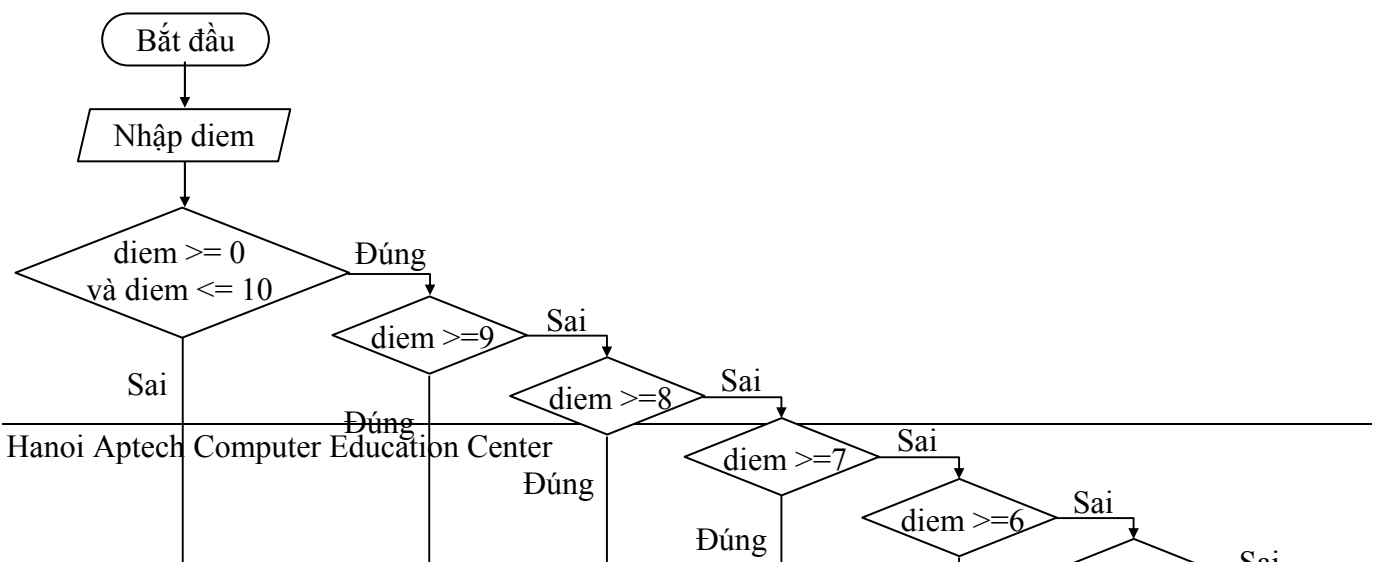
a. Phác họa lời giải

Điểm số nhập vào nếu hợp lệ ($0 \leq \text{điểm} \leq 10$), bạn tiếp tục công việc xếp loại, ngược lại thông báo "Nhập điểm không hợp lệ". Việc xếp loại bạn sử dụng cấu trúc else if.

b. Mô tả quy trình xử lý (giải thuật)

Ngôn ngữ tự nhiên	Ngôn ngữ C
- Khai báo biến diem kiểu số thực - Nhập vào điểm số - Nếu diem ≥ 0 và diem ≤ 10 thì - Nếu diem ≥ 9 thì in ra xếp loại = Xuất sắc Ngược lại Nếu diem ≥ 8 thì in ra xếp loại = Giỏi Ngược lại Nếu diem ≥ 7 thì in ra xếp loại = Khá Ngược lại Nếu diem ≥ 6 thì in ra xếp loại = TBKhá Ngược lại Nếu diem ≥ 5 thì in ra xếp loại = TBình Ngược lại thì in ra xếp loại = Yếu Ngược lại thì in ra "Bạn nhập điểm không hợp lệ"	<pre> - float fdiem; - printf("Nhap vao diem so: "); scanf("%f", &fdiem); - if (fdiem >= 0 && fdiem <= 10) - if (fdiem >= 9) printf("Xep loai = Xuat sac.\n"); else if (fdiem >= 8) printf("Xep loai = Gioi.\n"); else if (fdiem >= 7) printf("Xep loai = Kha.\n"); else if (fdiem >= 6) printf("Xep loai = TBKha.\n"); else if (fdiem >= 5) printf("Xep loai = TBinh.\n"); else printf("Xep loai = Yeu.\n"); else printf("Ban nhap diem khong hop le.\n"); </pre>

c. Mô tả bằng lưu đồ



d. Viết chương trình

File Edit Search Run Compile Debug Project Option Window Help
<pre> /* Chương trình nhập vào 2 số nguyên a, b. In ra thông báo a > b, a < b, a = b */ #include <stdio.h> #include <conio.h> void main(void) { float fdiem; printf("Nhập vào điểm số: "); scanf("%f", &fdiem); if (fdiem >=0 && fdiem <=10) if (fdiem >=9) printf("Xếp loại = Xuất sắc.\n"); else if (fdiem >=8) printf("Xếp loại = Giỏi.\n"); else if (fdiem >=7) printf("Xếp loại = Khá.\n"); else if (fdiem >=6) printf("Xếp loại = TBKhá.\n"); else if (fdiem >=5) printf("Xếp loại = TBình.\n"); else printf("Xếp loại = Yếu.\n"); else //if (fdiem >=0 && fdiem <=10) printf("Nhập điểm không hợp lệ.\n"); getch(); } </pre>
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

☞ Kết quả in ra màn hình

Nhập vào điểm số: 6.5 Xếp loại = TBKhá.	Cho chạy lại chương trình và thử lại với: diem = 4, diem = 9, diem = 7, diem = 12 Quan sát và nhận xét kết quả
--	--

e. Bàn thêm về chương trình

Trong chương trình trên cấu trúc **else if được lồng vào trong cấu trúc dạng 2**, trong cấu trúc else if ta không cần đặt trong khối vì tất cả các if trong cấu trúc này đều có else, nên `else printf("Nhap diem khong hop le.\n")` đương nhiên là thuộc về if (`fdiem >= 0 && fdiem <= 10`). Giả sử trong cấu trúc else if không có dòng `else printf("Xep loai = Yeu.\n")` thì khi đó dòng `else printf("Nhap diem khong hop le.\n")` sẽ thuộc về cấu trúc else if chứ không thuộc về if (`fdiem >= 0 && fdiem <= 10`). Đối với trường hợp đó bạn cần phải đặt cấu trúc else if vào trong {}, thì khi đó dòng `else printf("Nhap diem khong hop le.\n")` sẽ thuộc về if (`fdiem >= 0 && fdiem <= 10`).

Ví dụ 11: Viết chương trình nhập vào 3 số nguyên a, b, c. Tìm và in ra số lớn nhất.

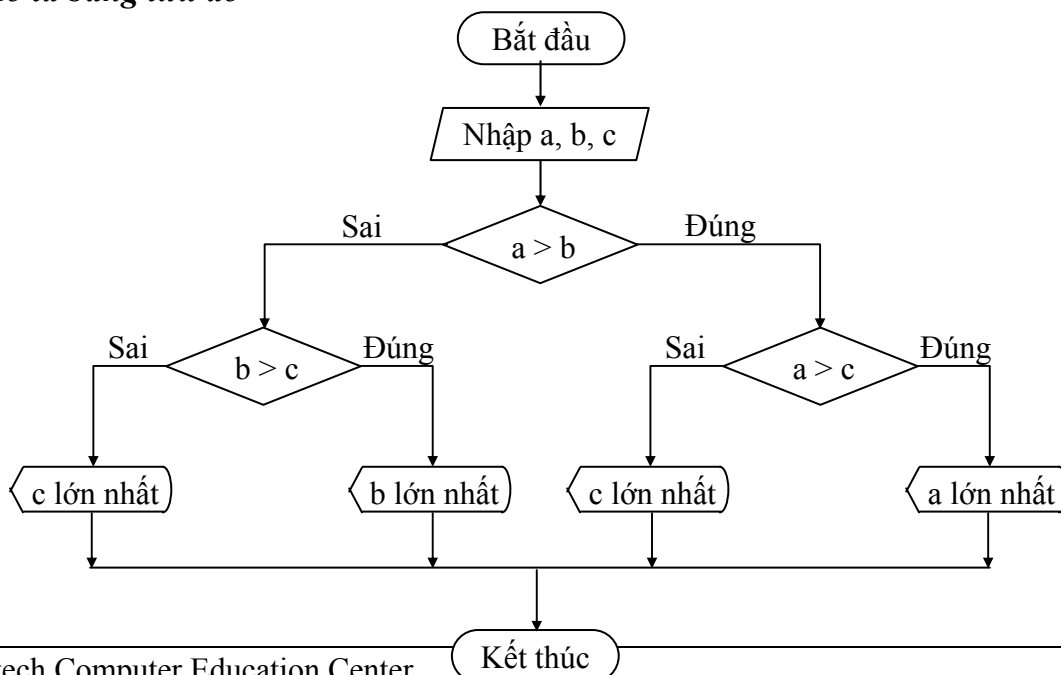
a. Phác họa lời giải

Trước tiên bạn so nếu $a > b$, mà $a > c$ thì a lớn nhất, ngược lại c lớn nhất, còn nếu $a \leq b$, mà $c > b$ thì b lớn nhất, ngược lại c lớn nhất.

b. Mô tả quy trình xử lý (giải thuật)

Ngôn ngữ tự nhiên	Ngôn ngữ C
- Khai báo 3 biến a, b, c kiểu số nguyên - Nhập vào số a - Nhập vào số b - Nhập vào số c - Nếu $a > b$ thì - Nếu $a > c$ thì a lớn nhất Ngược lại thì c lớn nhất Ngược lại - Nếu $b > c$ thì b lớn nhất Ngược lại thì c lớn nhất	- int ia, ib, ic; - printf("Nhap vao so a: "); scanf("%d", &ia); - printf("Nhap vao so b: "); scanf("%d", &ib); - printf("Nhap vao so c: "); scanf("%d", &ic); - if (ia > ib) - if (ia > ic) printf("%d lon nhat.\n", ia); else printf("%d lon nhat.\n", ic); else - if (ib > ic) printf("%d lon nhat.\n", ib); else printf("%d lon nhat.\n", ic);

c. Mô tả bằng lưu đồ



• **Cú pháp lệnh**

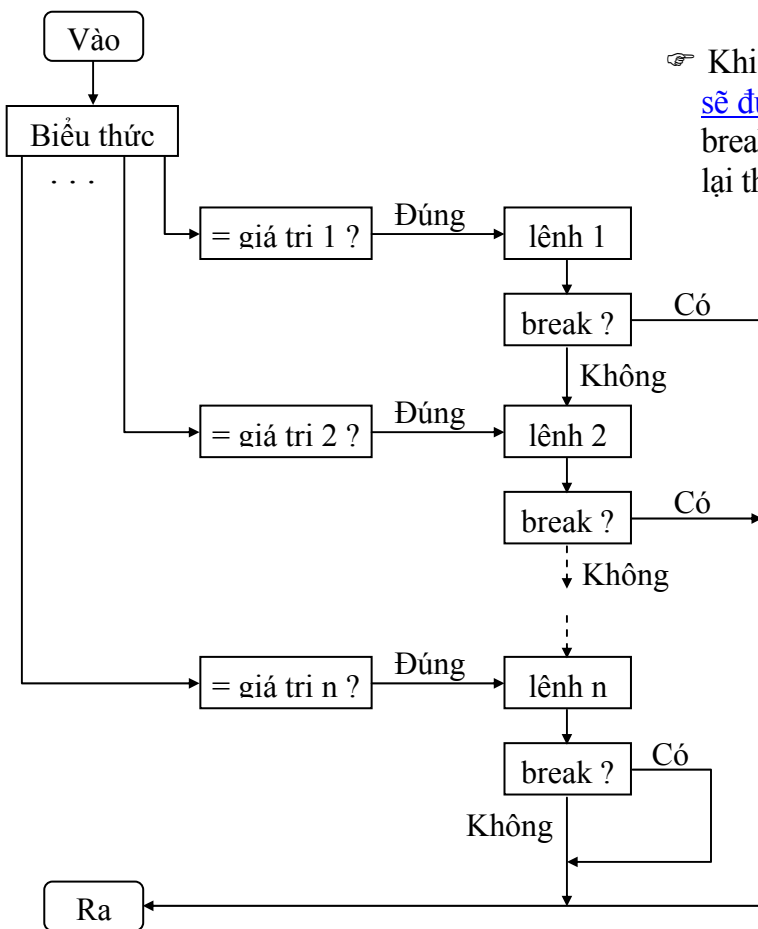
```

switch (biểu thức)
{
    case giá trị 1 : lệnh 1;
                   break;
    case giá trị 2 : lệnh 2;
                   break;
    ...
    case giá trị n : lệnh n;
                   [break;]
}

```

- ☞ từ khóa **switch, case, break** phải viết bằng chữ thường
- ☞ **biểu thức** phải là có kết quả là **giá trị hằng nguyên (char, int, long,...)**
- ☞ **Lệnh 1, 2...n** có thể gồm nhiều lệnh, nhưng không cần đặt trong cặp dấu { }

• **Lưu đồ**



- ☞ Khi **giá trị của biểu thức bằng giá trị i** thì **lệnh i sẽ được thực hiện**. Nếu sau lệnh i không có lệnh break thì sẽ tiếp tục thực hiện lệnh i + 1... Ngược lại thoát khỏi cấu trúc switch.

Ví dụ 12: Viết chương trình nhập vào số 1, 2, 3. In ra tương ứng 1, 2, 3 sao.

a. Viết chương trình

```

File Edit Search Run Compile Debug Project Option Window Help
/* Chương trình nhập vào số 1, 2, 3. In ra số sao tương ứng */
#include <stdio.h>
#include <conio.h>
void main(void)
{
    int i;
    printf("Nhập vào số 1, 2 hoặc 3: ");

```

```
scanf("%d", &i);
switch(i)
{
    case 3: printf("*");
    case 2: printf("*");
    case 1: printf("*");
};
printf("An phim bat ky de ket thuc!\n");
getch();
}
```

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

☞ Kết quả in ra màn hình

Nhap vao so 1, 2 hoặc 3: 2
**

Cho chạy lại chương trình và thử lại với:
i = 1, i = 3, i = 0, i = 4
Quan sát và nhận xét kết quả

b. Bàn thêm về chương trình

Trong chương trình trên khi nhập vào i = 2 lệnh printf("*") ở dòng case 2 được thi hành, nhưng do không có lệnh break sau đó nên lệnh printf("*") ở dòng case 1 tiếp tục được thi hành. Kết quả in ra **.

☞ Không đặt dấu chấm phẩy sau câu lệnh switch.

Ví dụ: switch(i);

→ trình biên dịch không báo lỗi nhưng các lệnh trong switch không được thực hiện.

Ví dụ 13: Viết chương trình nhập vào tháng và in ra quý. (tháng 1 -> quý 1, tháng 10 -> quý 4)

a. Phác họa lời giải

Nhập vào giá trị tháng, kiểm tra xem tháng có hợp lệ (trong khoảng 1 đến 12). Nếu hợp lệ in ra quý tương ứng (1->3: quý 1, 4->6: quý 2, 7->9: quý 3, 10->12: quý 4).

b. Viết chương trình

File Edit Search Run Compile Debug Project Option Window Help

```
/* Chuong trinh nhap vao thang. In ra quy tuong ung */
```

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
void main(void)
```

```
{
    int ithang;
    printf("Nhap vao thang: ");
    scanf("%d", &ithang);
    if (ithang > 0 && ithang <= 12)
        switch(ithang)
        {
            case 1:
            case 2:
            case 3: printf("Quy 1.\n");
                    break;

            case 4:
            case 5:
            case 6: printf("Quy 2.\n");
                    break;

            case 7:
            case 8:
```

Ví dụ 14: Viết lại chương trình ở **Ví dụ 12**

a. Viết chương trình

File Edit Search Run Compile Debug Project Option Window Help
<pre> /* Chuong trinh nhap vao so 1, 2, 3. In ra so sao tuong ung */ #include <stdio.h> #include <conio.h> void main(void) { int i; printf("Nhap vao so 1, 2 hoặc 3: "); scanf("%d", &i); switch(i) { case 3: printf("*"); case 2: printf("*"); case 1: printf("*"); break; default: printf("Ban nhap phai nhap vao so 1, 2 hoac 3.\n"); }; getch(); } </pre>
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

 **Kết quả in ra màn hình**

Nhap vao so 1, 2 hoặc 3: 3 *** _	Cho chạy lại chương trình và thử lại với: i = 1, i = 3, i = 0, i = 4 Quan sát kết quả
--	---

b. Bàn thêm về chương trình

Tương tự 2 dạng trên. Nhưng trong mỗi lệnh có thể có một (nhiều) cấu trúc switch ở 2 dạng trên.

Ví dụ 16: Viết chương trình menu 2 cấp

a. Viết chương trình

File Edit Search Run Compile Debug Project Option Window Help
<pre> /* Chuong trinh menu 2 cap */ #include <stdio.h> #include <conio.h> void main(void) { int imenu, isubmenu; printf("-----\n"); printf(" MAIN MENU \n"); printf("-----\n"); printf("1. File\n"); printf("2. Edit\n"); printf("3. Search\n"); printf("Chon muc tuong ung: "); scanf("%d", &imenu); switch(imenu) { case 1: printf("-----\n"); printf(" MENU FILE \n"); printf("-----\n"); printf("1. New\n"); printf("2. Open\n"); printf("Chon muc tuong ung: "); scanf("%d", &isubmenu); switch(isubmenu) { case 1: printf("Ban da chon chuc nang New File\n"); break; case 2: printf("Ban da chon chuc nang Open File\n"); break; } break; //break cua case 1 – switch(imenu) case 2: printf("Ban da chon chuc nang Edit\n"); break; case 3: printf("Ban da chon chuc nang Search\n"); }; getch(); } </pre>
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

Kết quả in ra màn hình

<pre> ----- MAIN MENU </pre>	Cho chạy lại chương trình và thử lại với: mục chọn chức năng khác
------------------------------	--

18. Viết chương trình nhập vào 3 số nguyên rồi in ra màn hình theo thứ tự tăng dần.

19. Viết chương trình tính tiền điện gồm các khoảng sau:

- Tiền thuê bao điện kê: 1000đ/tháng
- Định mức sử dụng điện cho mỗi hộ là: 50 KW với giá 230đ/KW
- Nếu phần vượt định mức $\leq 50KW$ thì tính giá 480đ/KW
- Nếu $50KW < \text{phần vượt định mức} < 100KW$ thì tính giá 700đ/KW
- Nếu phần vượt định mức $\leq 100KW$ thì tính giá 900đ/KW

Chỉ số mới và cũ được nhập vào từ bàn phím

- In ra màn hình chỉ số cũ, chỉ số mới, tiền trả định mức, tiền trả vượt định mức, tổng tiền phải trả.

+ Nếu biểu thức 2 không có, vòng for được xem là luôn luôn **đúng**. Muốn thoát khỏi vòng lặp for phải dùng một trong 3 lệnh **break**, **goto** hoặc **return**.

+ Với mỗi biểu thức có thể viết thành một dãy biểu thức con phân cách nhau bởi dấu phẩy. Khi đó các biểu thức con được xác định từ trái sang phải. Tính đúng sai của dãy biểu thức con trong biểu thức thứ 2 được xác định bởi biểu thức con cuối cùng.

+ Trong thân for (khối lệnh) có thể chứa một hoặc nhiều cấu trúc điều khiển khác.

+ Khi gặp lệnh **break**, cấu trúc lặp sâu nhất sẽ thoát ra.

+ Trong thân for có thể dùng lệnh **goto** để thoát khỏi vòng lặp đến vị trí mong muốn.

+ Trong thân for có thể sử dụng **return** để trở về một hàm nào đó.


+ Trong thân for có thể sử dụng lệnh continue để chuyển đến đầu vòng lặp (bỏ qua các câu lệnh còn lại trong thân).

Ví dụ 1: Viết chương trình in ra câu "Vi dụ sử dụng vòng lặp for" 3 lần.

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	/* Chương trình in ra câu "Vi dụ sử dụng vòng lặp for" 3 lần */
2	
3	#include <stdio.h>
4	#include <conio.h>
5	
6	#define MSG "Vi dụ sử dụng vòng lặp for.\n"
7	
8	void main(void)
9	{
10	int i;
11	for(i = 1; i<=3; i++) /hoac for(i = 1; i<=3; i+=1)
12	printf("%s", MSG);
13	getch();
14	}
	F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

Kết quả in ra màn hình

Vi dụ sử dụng vòng lặp for. Vi dụ sử dụng vòng lặp for. Vi dụ sử dụng vòng lặp for.	Bạn thay 2 dòng 11 và 12 bằng câu lệnh for(i=1; i<=3; i++, printf("%s", MSG)); Chạy lại chương trình, quan sát và nhận xét kết quả.
---	--

 **Có dấu chấm phẩy sau lệnh for(i=1; i<=3; i++); → các lệnh thuộc vòng lặp for sẽ không được thực hiện.**

Ví dụ 2: Viết chương trình nhập vào 3 số nguyên. Tính và in ra tổng của chúng.

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	/* Chương trình nhập vào 3 số và tính tổng */
2	
3	#include <stdio.h>
4	#include <conio.h>
5	
6	void main(void)
7	{
8	int i, in, is;
9	is = 0;

10	for(i = 1; i<=3; i++)
11	{
12	printf("Nhap vao so thu %d :", i);
13	scanf("%d", &in);
14	is = is + in;
15	}
16	printf("Tong: %d", is);
17	getch();
18	}
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu	

☞ **Kết quả in ra màn hình**

Nhap vao so thu 1: 5 Nhap vao so thu 2: 4 Nhap vao so thu 3: 2 Tong: 11.	Bạn thay các dòng từ 9 đến 15 bằng câu lệnh: for(is=0, i=1; i<=3; printf("Nhap vao so thu %d: ", i), scanf("%d", &in), i++, is=is+in); Chạy lại chương trình, quan sát và nhận xét kết quả.
---	---

☞ Trong vòng lặp for có sử dụng từ 2 lệnh trở lên, nhớ sử dụng cặp ngoặc { } để bọc các lệnh đó lại. Dòng 12, 13, 14 thuộc vòng for dòng 10 do được bọc bởi cặp ngoặc { }. Nếu 3 dòng này không bọc bởi cặp ngoặc { }, thì chỉ dòng 12 thuộc vòng lặp for, còn 2 dòng còn lại không thuộc vòng lặp for.

Ví dụ 3: Viết chương trình nhập vào số nguyên n. Tính tổng các giá trị lẻ từ 0 đến n.

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	/* Chuong trinh nhap vao 3 so va tinh tong */
2	
3	#include <stdio.h>
4	#include <conio.h>
5	
6	void main(void)
7	{
8	int i, in, is = 0;
9	printf("Nhap vao so n: ");
10	scanf("%d", &in);
11	is = 0;
12	for(i = 0; i<=in; i++)
13	{
14	if (i % 2 != 0) //neu i la so le
15	is = is + i; //hoac is += i;
16	}
17	printf("Tong: %d", is);
18	getch();
19	}
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu	

☞ **Kết quả in ra màn hình**

Nhap vao so n : 5 Tong: 9.	Bạn thay các dòng từ 11 đến 16 bằng câu lệnh: for(is=0, i=1; i<=n; is=is+i, i+=2); Chạy lại chương trình, quan sát và nhận xét kết quả.
-------------------------------	---

☞ Bạn có thể viết gộp các lệnh trong thân for vào trong lệnh for. Tuy nhiên, khi lập trình bạn nên viết lệnh for có đủ 3 biểu thức đơn và các lệnh thực hiện trong thân for mỗi lệnh một dòng để sau này có thể đọc lại dễ hiểu, dễ sửa chữa.

Ví dụ 4: Một vài ví dụ thay đổi biến điều khiển vòng lặp.

- Thay đổi biến điều khiển từ 1 đến 100, mỗi lần tăng 1:
for(i = 1; i <= 100; i++)
- Thay đổi biến điều khiển từ 100 đến 1, mỗi lần giảm 1:
for(i = 100; i >= 1; i--)
- Thay đổi biến điều khiển từ 7 đến 77, mỗi lần tăng 7:
for(i = 7; i <= 77; i += 7)
- Thay đổi biến điều khiển từ 20 đến 2, mỗi lần giảm 2:
for(i = 20; i >= 2; i -= 2)

Ví dụ 5: Đọc vào một loạt kí tự trên bàn phím. Kết thúc khi gặp dấu chấm '!'.

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	/* Doc vao 1 loat ktu tren ban phim. Ket thuc khi gap dau cham */
2	
3	#include <stdio.h>
4	
5	#define DAU_CHAM '!
6	
7	void main(void)
8	{
9	char c;
10	for(; (c = getchar()) != DAU_CHAM;)
11	putchar(c);
12	}
	F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

☞ Kết quả in ra màn hình

a	Bạn thay các dòng từ 10 đến 11 bằng câu lệnh: for(; (c = getchar()) != DAU_CHAM; putchar(c)); Chạy lại chương trình, quan sát và nhận xét kết quả.
a	
4	
4	
.	

☞ Vòng lặp for vắng mặt biểu thức 1 và 3.

Ví dụ 6: Đọc vào một loạt kí tự trên bàn phím, đếm số kí tự nhập vào. Kết thúc khi gặp dấu chấm '!'.

Dòng	File Edit Search Run Cmpile Debug Project Option Window Help
1	/* Doc vao 1 loat ktu tren ban phim, dem so ktu nhap vao. Ket thuc khi gap dau cham */
2	
3	#include <stdio.h>
4	#include <conio.h>
5	
6	#define DAU_CHAM '!
7	
8	void main(void)

9	{
10	char c;
11	int idem;
12	for(idem = 0; (c = getchar()) != DAU_CHAM;)
13	idem++;
14	printf("So ki tu: %d.\n", idem);
15	getch();
16	}
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu	

☞ **Kết quả in ra màn hình**

afser. So ki tu: 5.	Bạn thay các dòng từ 12 đến 13 bằng câu lệnh: for(idem = 0; (c = getchar()) != DAU_CHAM; idem++); Chạy lại chương trình, quan sát và nhận xét kết quả.
------------------------	---

☞ **Vòng lặp for vắng mặt biểu thức 3.**

Ví dụ 7: Đọc vào một loạt kí tự trên bàn phím, đếm số kí tự nhập vào. Kết thúc khi gặp dấu chấm '!'. .

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	/* Doc vao 1 loạt ktu tren ban phim, dem so ktu nhap vao. Ket thuc khi gap dau cham */
2	
3	#include <stdio.h>
4	#include <conio.h>
5	
6	#define DAU_CHAM '!'.
7	
8	void main(void)
9	{
10	char c;
11	int idem = 0;
12	for(;;)
13	{
14	c = getchar();
15	if (c == DAU_CHAM) //nhap vao dau cham
16	break; //thoat vong lap
17	idem++;
18	}
19	printf("So ki tu: %d.\n", idem);
20	getch();
21	}
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu	

☞ **Kết quả in ra màn hình**

afser. So ki tu: 5.	Chạy lại chương trình, quan sát và nhận xét kết quả.
------------------------	--

☞ **Vòng lặp for vắng mặt cả ba biểu thức.**

Ví dụ 8: Nhập vào 1 dãy số nguyên từ bàn phím đến khi gặp số 0 thì dừng. In ra tổng các số nguyên dương.

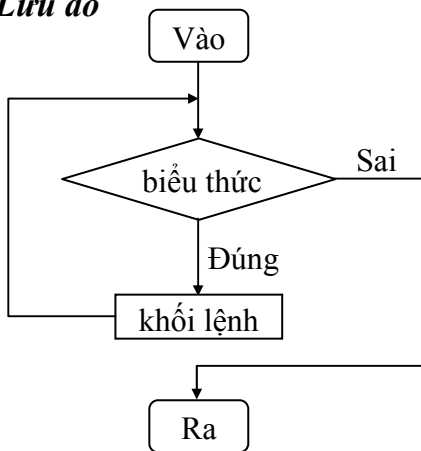
Dòng	File Edit Search Run Compile Debug Project Option Window Help
------	---

**while (biểu thức)
khối lệnh;**

☞ từ khóa **while** phải viết bằng chữ thường

☞ Nếu **khối lệnh** bao gồm từ 2 lệnh trở lên thì phải đặt trong dấu { }

• Lưu đồ



☞ Trước tiên **biểu thức** được kiểm tra
 nếu **sai** thì
 kết thúc vòng lặp while
 (khối lệnh không được thi hành 1 lần nào)
 nếu **đúng**
 thực hiện khối lệnh;
 lặp lại kiểm tra biểu thức

+ Biểu thức: có thể là một biểu thức hoặc nhiều biểu thức con. Nếu là nhiều biểu thức con thì cách nhau bởi dấu phẩy (,) và tính đúng sai của biểu thức được quyết định bởi biểu thức con cuối cùng.

+ Trong thân while (khối lệnh) có thể chứa một hoặc nhiều cấu trúc điều khiển khác.

+ Trong thân while có thể sử dụng lệnh continue để chuyển đến đầu vòng lặp (bỏ qua các câu lệnh còn lại trong thân).

+ Muốn thoát khỏi vòng lặp while tùy ý có thể dùng các lệnh **break, goto, return** như lệnh **for**.

Ví dụ 11: Viết chương trình in ra câu "Vi dụ su dung vong lap while" 3 lần.

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	/* Chuong trinh in ra cau "Vi dụ su dung vong lap while" 3 lan */
2	
3	#include <stdio.h>
4	#include <conio.h>
5	
6	#define MSG "Vi dụ su dung vong lap while.\n"
7	
8	void main(void)
9	{
10	int i = 0;
11	while (i++ < 3)
12	printf("%s", MSG);
13	getch();
14	}
	F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

☞ **Kết quả in ra màn hình**

Vi dụ su dung vong lap while. Vi dụ su dung vong lap while. Vi dụ su dung vong lap while.	Bạn thay 2 dòng 11 và 12 bằng câu lệnh while(printf("%s", MSG), ++i < 3); Chạy lại chương trình và quan sát kết quả.
---	---

Ví dụ 12: Viết chương trình tính tổng các số nguyên từ 1 đến n, với n được nhập vào từ bàn phím.

Kiểm tra biểu thức

Nếu **đúng** thì

lặp lại thực hiện khối lệnh

Nếu **sai** thì

kết thúc vòng lặp

(khối lệnh được thi hành 1 lần)

+ Biểu thức: có thể là một biểu thức hoặc nhiều biểu thức con. Nếu là nhiều biểu thức con thì cách nhau bởi dấu phẩy (,) và tính đúng sai của biểu thức được quyết định bởi biểu thức con cuối cùng.

+ Trong thân do...while (khối lệnh) có thể chứa một hoặc nhiều cấu trúc điều khiển khác.

+ Trong thân do...while có thể sử dụng lệnh continue để chuyển đến đầu vòng lặp (bỏ qua các câu lệnh còn lại trong thân).

+ Muốn thoát khỏi vòng lặp do...while tùy ý có thể dùng các lệnh **break, goto, return**.

Ví dụ 16: Viết chương trình kiểm tra password.

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	/* Chuong trinh kiem tra mat khau */
2	
3	#include <stdio.h>
4	
5	# define PASSWORD 12345
6	
7	void main(void)
8	{
9	int in;
10	do
11	{
12	printf("Nhap vao password: ");
13	scanf("%d", &in);
14	} while (in != PASSWORD)
15	}
	F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

 **Kết quả in ra màn hình**

Nhap vao password: 1123 Nhap vao password: 12346 Nhap vao password: 12345	Bạn thay các dòng từ 10 đến 14 bằng câu lệnh: do{}while(printf("Nhap vao password: "), scanf("%d", &in), in != PASSWORD); Chạy lại chương trình và quan sát kết quả.
---	---

Ví dụ 17: Viết chương trình nhập vào năm hiện tại, năm sinh. In ra tuổi.

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	/* Chuong trinh in tuoi */
2	
3	#include <stdio.h>
4	
5	# define CHUC "Chuc ban vui ve (: >\n"
6	
7	void main(void)
8	{

22. **Viết chương trình tìm các số nguyên gồm 3 chữ số sao cho tích của 3 chữ số bằng tổng 3 chữ số.** Ví dụ: $1*2*3 = 1+2+3$.

23. **Viết chương trình tìm các số nguyên a, b, c, d khác nhau trong khoảng từ 0 tới 10 thỏa mãn điều kiện $a*d*d = b*c*c*c$**

24. **Viết chương trình tính tổ hợp N chập K (với $K \leq N$)**

$$C = \frac{(N-k+1) * (N-k+2) * \dots * N}{1*2*3* \dots * k}$$

Trong đó C là một tích gồm k phần tử với phần tử thứ I là $(N-k+1)/I$. Để viết chương trình này, bạn dùng vòng lặp For với biến điều khiển I từ giá trị đầu là 1 tăng đến giá trị cuối là k kết hợp với việc nhân dồn vào kết quả C .

25. **Viết chương trình giải bài toán cổ điển sau:**

Trăm trâu, trăm cỏ
 Trâu đứng ăn năm
 Trâu nằm ăn ba,
 Ba trâu già ăn một
 Hỏi mỗi loại trâu có bao nhiêu con.

26. **Viết chương trình giải bài toán cổ điển sau:**

Vừa gà vừa chó 36 con
 Bó lại cho tròn, đếm đủ 100 chân
 Hỏi có bao nhiêu gà, bao nhiêu chó

27. **Viết chương trình in ra bảng cửu chương**

28. **Viết chương trình xác định xem một tờ giấy có độ dày 0.1 mm. Phải gấp đôi tờ giấy bao nhiêu lần để nó có độ dày 1m.**

29. **Viết chương trình tìm các số nguyên tố từ 2 đến N , với N được nhập vào.**

30. **Viết chương trình lặp đi lặp lại các công việc sau:**

- Nhập vào một ký tự trên bàn phím.
- Nếu là chữ thường thì in ra chính nó và chữ HOA tương ứng.
- Nếu là chữ HOA thì in ra chính nó và chữ thường tương ứng.
- Nếu là ký số thì in ra chính nó.
- Nếu là một ký tự điều khiển thì kết thúc chương trình

31. **Viết chương trình nhập vào x, n tính:**

$$- \sqrt{x + \sqrt{x + \dots + \sqrt{x}}} \quad (n \text{ dấu căn})$$

$$- 1 + \frac{x}{2} + \dots + \frac{x^n}{n+1}$$

32. **Viết chương trình nhập vào N số nguyên, đếm xem có bao nhiêu số âm, bao nhiêu số dương và bao nhiêu số không.**



Giải thích chương trình

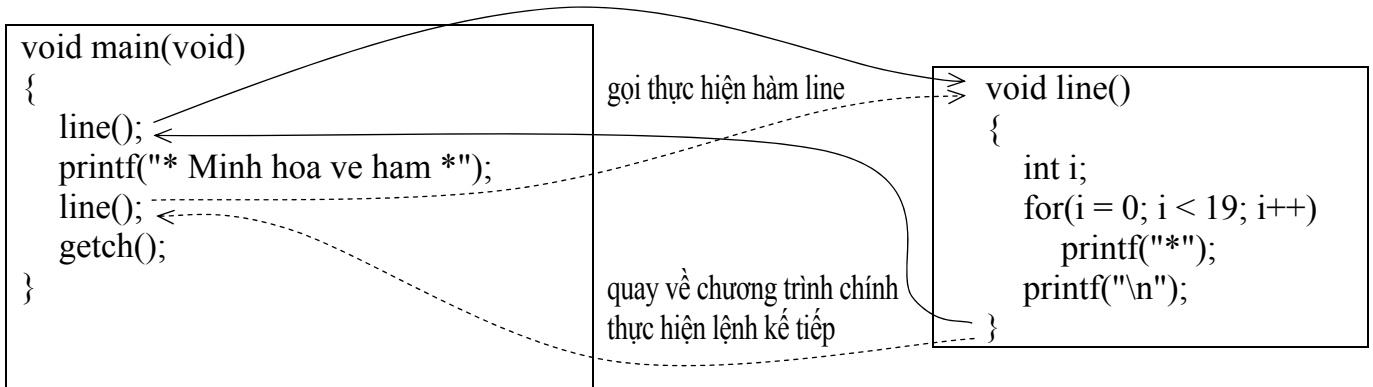
Dòng 8 đến dòng 14: định nghĩa hàm **line**, hàm này không trả về giá trị, thực hiện công việc in ra 19 dấu sao.

Dòng 5: khai báo prototype, sau tên hàm phải có dấu chấm phẩy

Trong hàm line có sử dụng biến i, biến i là biến cục bộ chỉ sử dụng được trong phạm vi hàm line.

Dòng 18 và 20: gọi thực hiện hàm line.

*** Trình tự thực hiện chương trình**



⊗ Không có dấu chấm phẩy sau tên hàm, phải có cặp dấu ngoặc () sau tên hàm nếu hàm không có tham số truyền vào. Phải có dấu chấm phẩy sau tên hàm khai báo prototype. Nên khai báo prototype cho dù hàm được gọi nằm trước hay sau câu lệnh gọi nó.

Ví dụ 2:

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	#include <stdio.h>
2	#include <conio.h>
3	
4	// khai bao prototype
5	int power(int, int);
6	
7	// ham tinh so mu
8	int power(int ix, int in)
9	{
10	int i, ip = 1;
11	for(i = 1; i <= in; i++)
12	ip *= ix;
13	return ip;
14	}
15	
16	void main(void)
17	{
18	printf("2 mu 2 = %d.\n", power(2, 2));
19	printf("2 mu 3 = %d.\n", power(2, 3));
20	getch();
	}
	F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

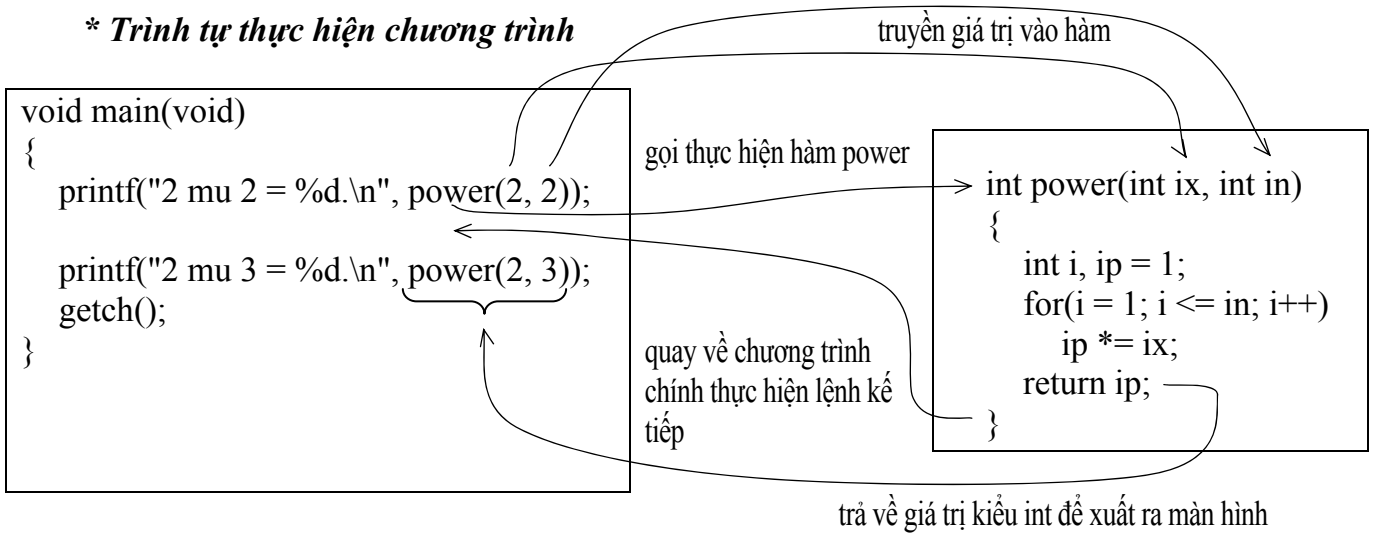
👉 Kết quả in ra màn hình

```
2 mu 2 = 4.
2 mu 3 = 8.
-
```

🔗 Giải thích chương trình

Hàm **power** có hai tham số truyền vào là ix, in có kiểu int và kiểu trả về cũng có kiểu int.
 Dòng 13: return ip: trả về giá trị sau khi tính toán
 Dòng 18: đối mục 2 và 3 có kiểu trả về là int sau khi thực hiện gọi power.
 Hai tham số ix, in của hàm power là dạng truyền tham trị.

*** Trình tự thực hiện chương trình**



👉 Quy tắc đặt tên hàm giống tên biến, hằng... Mỗi đối số cách nhau = dấu phẩy kèm theo kiểu dữ liệu tương ứng.

Ví dụ 3:

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	#include <stdio.h>
2	#include <conio.h>
3	
4	// khai bao prototype
5	void time(int &, int &);
6	
7	// ham doi phut thanh gio:phut
8	void time(int &ig, int &ip)
9	{
10	ig = ip / 60;
11	ip %= 60;
12	}
13	
14	void main(void)
15	{
16	int igio, iphut;

Giải thích chương trình

Chương trình trên gồm 2 hàm **oddeven** và **negative**, 2 hàm này bạn thấy không có tham số để truyền biến inum vào xử lý nhưng vẫn cho kết quả đúng. Do chương trình sử dụng biến **inum** toàn cục (dòng.9) nên biến này có ảnh hưởng đến toàn bộ chương trình mỗi khi gọi và sử dụng nó. Xét tình huống sau: Giả sử trong hàm **negative** ta khai báo biến inum có kiểu **int** như sau:

```
void negative()
{
    int inum;
    ....
}
```

Khi đó chương trình sẽ cho kết quả sai! Do các câu lệnh trong hàm **negative** sử dụng biến **inum** sẽ sử dụng biến **inum** khai báo trong hàm **negative** và lúc này biến **inum** toàn cục không có tác dụng đối với các câu lệnh trong hàm này. Biến **inum** khai báo trong hàm **negative** chỉ có ảnh hưởng trong phạm vi hàm và chu trình sống của nó bắt đầu từ lúc gọi hàm đến khi thực hiện xong.

 **Cẩn thận khi đặt tên biến, xác định rõ phạm vi của biến khi sử dụng để có thể dễ dàng kiểm soát chương trình.**

Ví dụ 6:

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	#include <stdio.h>
2	#include <conio.h>
3	
4	#define PI 3.14
5	
6	// khai bao prototype
7	float area();
8	
9	//khai bao bien toan cuc
10	float frad;
11	
12	void main(void)
13	{
14	printf("Nhap vao ban kinh hinh cau : ");
15	scanf("%f", &frad);
16	printf("Dien tich hinh cau: %10.3f.\n", area());
17	getch();
18	}
19	
20	// ham tinh dien tich hinh cau
21	float area()
22	{
23	return (4*PI*frad*frad);
24	}
	F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

Kết quả in ra màn hình


```

13 {
14     int i, imax;
15     imax = ia[0];           //cho phan tu dau tien la max
16     for (i = 1; i < in; i++)
17         if (imax < ia[i])   //neu so dang xet > max
18             imax = ia[i];   //gan so nay cho max
19     return imax;          //tra ve ket qua so lon nhat
20 }
21
22 void main(void)
23 {
24     int ia[MAX];
25     int i = 0, inum;
26     do
27     {
28         printf("Nhap vao mot so: ");
29         scanf("%d", &ia[i]);
30     } while (ia[i++] != 0);
31     i--;
32     inum = max(ia, i);
33     printf("So lon nhat la: %d.\n", inum);
34     getch();
35 }

```

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

Kết quả in ra màn hình

Nhap vao mot so: 12 Nhap vao mot so: 45 Nhap vao mot so: 3 Nhap vao mot so: 0 So lon nhat la: 45 —	Chạy lại chương trình và thử lại với số liệu khác. Thực hiện một số thay đổi sau: - Di chuyển dòng int a[MAX] ; lên sau dòng số 10 - Sửa dòng int max(int, int) ; thành int max(int) ; - Sửa dòng int max(int a[], int n) ; thành int max(int n) ; - Sửa dòng num = max(a, i) ; thành num = max(i) ; Chạy lại chương trình, quan sát, nhận xét và đánh giá kết quả.
---	---


Giải thích chương trình

Chương trình ban đầu hàm max có hai tham số truyền vào và kết quả trả về là giá trị max có kiểu nguyên, một tham số là mảng 1 chiều kiểu int và một tham số có kiểu int. Với chương trình sau khi sửa hàm max chỉ còn một tham số truyền vào nhưng cho kết quả như nhau. Do sau khi sửa chương trình mảng a[MAX] được khai báo lại là biến toàn cục nên hàm max không cần truyền tham số mảng vào cũng có thể sử dụng được. Tuy vậy, khi lập trình bạn nên viết như chương trình ban đầu là truyền tham số mảng vào (dạng tổng quát) để hàm max có thể thực hiện được trên nhiều mảng khác nhau. Còn với chương trình sửa lại bạn chỉ sử dụng hàm max được với mảng a mà thôi.

Ví dụ 13 : Bạn khai báo các mảng sau ia[MAX], ib[MAX], ic[MAX]. Để tìm giá trị lớn nhất của từng mảng. Bạn chỉ cần gọi hàm

- imax_a = max(ia, i);
- imax_b = max(ib, i);
- imax_c = max(ic, i);

Với chương trình sửa lại bạn không thể tìm được số lớn nhất của mảng b và c.

 **Bạn lưu ý rằng khi truyền mảng sang hàm, không tạo bản sao mảng mới. Vì vậy mảng truyền sang hàm có dạng tham biến. Nghĩa là giá trị của các phần tử trong mảng sẽ bị ảnh hưởng nếu có sự thay đổi trên chúng.**

Ví dụ 14 : Tìm số lớn nhất của 3 mảng a, b, c

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	/* Chương trình tìm số lớn nhất sử dụng hàm */
2	
3	#include <stdio.h>
4	#include <conio.h>
5	
6	#define MAX 20
7	
8	//Khai báo prototype
9	int max(int, int);
10	int input(int);
11	
12	//hàm tìm phần tử lớn nhất trong mảng 1 chiều
13	int max(int ia[], int in)
14	{
15	int i, imax;
16	imax = ia[0]; //cho phần tử đầu tiên là max
17	for (i = 1; i < in; i++)
18	if (max < ia[i]) //nếu số đang xét > max
19	max = ia[i]; //gán số này cho max
20	return imax; //tra về kết quả số lớn nhất
21	}
22	
23	//hàm nhập liệu vào mảng 1 chiều
24	int input(int ia[])
25	{
26	int i = 0;
27	do
28	{
29	printf("Nhập vào một số: ");
30	scanf("%d", &ia[i]);
31	} while (ia[i++] != 0);
32	i--;
33	return i;
34	}
35	
36	void main(void)
37	{
38	int ia[MAX], ib[MAX], ic[MAX];
39	int inum1, inum2, inum3;
40	printf("Nhập liệu cho mảng a: \n");
41	inum1 = max(ia, input(ia));
42	printf("Nhập liệu cho mảng b: \n");
43	inum2 = max(ib, input(ib));
44	printf("Nhập liệu cho mảng c: \n");
45	inum3 = max(ic, input(ic));
46	printf("Số lớn nhất của mảng a: %d, b: %d, c: %d.\n", inum1, inum2, inum3);
47	getch();
48	}
	F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

 **Kết quả in ra màn hình**

```

30 void input(float fa[][MAX])
31 {
32     for (int i = 0; i < in; i++)
33         for (int ij = 0; ij < in; ij++)
34             {
35                 printf("Nhap vao ptu[%d][%d]: ", i, ij);
36                 scanf("%f", &fa[i, j]);
37             }
38 }
39
40 //ham in mang 2 chieu ra man hinh
41 void output(float fa[][MAX])
42 {
43     for (int i = 0; i < in; i++)
44         {
45             for (int ij = 0; ij < n; ij++)
46                 printf("%5.2f", fa[i][ij]);
47             printf("\n");
48         }
49 }
50
51 //ham cong 2 mang 2 chieu
52 void add(float fa[][MAX], float fb[][MAX], float fc[][MAX])
53 {
54     for (int i = 0; i < in; i++)
55         for (int ij = 0; ij < in; ij++)
56             fc[i, ij] = fa[i, ij] + fb[i, ij];
57 }
58
59 void main(void)
60 {
61     float fa[MAX][MAX], fb[MAX][MAX], fc[MAX][MAX];
62     printf("Nhap vao cap ma tran: ");
63     scanf("%d", &in);
64     printf("Nhap lieu ma tran a: \n");
65     input(fa);
66     printf("Nhap lieu ma tran b: \n");
67     input(fb);
68     printf("Nhap lieu ma tran c: \n");
69     input(fc);
70     add(fa, fb, fc);
71     printf("Ma tran a: \n");
72     output(fa);
73     printf("Ma tran b: \n");
74     output(fb);
75     printf("Ma tran c: \n");
76     output(fc);
77     printf("So lon nhat cua ma tran c la: %5.2f.\n", max(fc));
78     getch();
79 }

```

F1 Help **Alt-F8** Next Msg **Alt-F7** Prev Msg **Alt - F9** Compile **F9** Make **F10** Menu

 **Kết quả in ra màn hình**

14. Viết chương trình nhập vào một chuỗi gồm các chữ cái (a -> z, A -> Z). Hãy đếm xem có bao nhiêu nguyên âm a, i, e, o, u.

15. Giả sử số phòng trong một khách sạn được cho bởi hằng số NUM_ROOM. Viết:

- Một khai báo dãy thích hợp để theo dõi phòng nào còn trống.
- Một hàm tìm phòng nào còn trống.
- Viết chương trình đơn giản để quản lý phòng khách sạn theo dạng một trình đơn chọn công việc gồm có 4 mục như sau:
 - Tìm phòng trống.
 - Trả phòng.
 - Liệt kê những phòng còn trống.
 - Liệt kê những phòng đã thuê.
 - Kết thúc.

16. Viết chương trình mô tả văn bản của một bức điện tín. Nhập liệu bao gồm 1 hay nhiều dòng chứa một số từ, mỗi từ cách nhau khoảng trắng. In ra hóa đơn tính tiền với mỗi từ giá 100 đồng, phí trả thêm 50 đồng cho từ dài quá 8 kí tự. Hóa đơn có dạng sau:

Số từ	: 10		
Số từ có kích thước bình thường	: 8 x 100 =	800	dong
Số từ có kích thước > 8 kí tự	: 2 x 150 =	300	dong
Tổng cộng	:	1100	dong

17. Viết chương trình thống kê xem có bao nhiêu người họ "Ly", "Tran"... trong 1 danh sách cho trước. Nếu không có thông báo "Không có người nào thuộc họ".

18. Viết chương trình nhập vào 1 chuỗi, sau đó chép sang chuỗi khác một chuỗi con từ chuỗi ban đầu có số kí tự chỉ định.

Ví dụ: Chuỗi ban đầu "Le Thuy Doan Trang". Nếu số kí tự chỉ định là 2 thì chuỗi đích sẽ là "Le"

19. Viết chương trình nhập vào 1 chuỗi, sau đó chép sang chuỗi khác một chuỗi con từ chuỗi ban đầu với vị trí bắt đầu và số kí tự chỉ định.

Ví dụ: Chuỗi ban đầu "Le Thuy Doan Trang". Nếu vị trí ban đầu là 14 và số kí tự chỉ định là 5 thì chuỗi đích sẽ là "Trang"

20. Viết chương trình nhập vào 1 chuỗi nguồn, ví dụ "Nguyen Minh Long", sau đó nhập vào 1 chuỗi con, ví dụ "Minh", chương trình sẽ xác định vị trí bắt đầu của chuỗi con ở vị trí nào trong chuỗi nguồn. Kết quả in ra màn hình như sau:

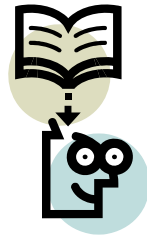
- Chuoi nguon la : Nguyen Minh Long
- Chuoi con la : Minh
- Vi tri bat dau cua chuoi con la : 8

21. Viết chương thực hiện các yêu cầu sau:

- Nhập vào 1 chuỗi bất kỳ, ví dụ : "Nguyen Minh Long
- Muốn xóa từ vị trí nào, ví dụ : 8
- Muốn xóa bao nhiêu kí tự, ví dụ : 5

Kết quả in ra màn hình:

- Chuoi nguon la : Nguyen Minh Long
- Chuoi sau khi xoa : Nguyen Long



```

8 #define MAXLEN 10
9
10 void main(void)
11 {
12     char cname[MAXNUM][MAXLEN]; //mang chuoai
13     char *cptr[MAXNUM];         //mang con tro tro den chuoai
14     char *ctemp;
15     int i, ij, icount = 0;
16
17     //nhap danh sach ten
18     while (icount < MAXNUM)
19     {
20         printf("Nhap vao ten nguoi thu %d: ", icount + 1);
21         gets(cname[icount]);
22         cptr[icount++] = cname[icount]; //con tro den ten
23     }
24
25     //sap xep danh sach theo thu tu tang dan
26     for (i = 0; i < icount - 1; i++)
27         for (ij = i + 1; ij < icount; ij++)
28             if (strcmp(cptr[i], cptr[ij]) > 0)
29             {
30                 ctemp = cptr[i];
31                 cptr[i] = cptr[ij];
32                 cptr[ij] = ctemp;
33             }
34
35     //In danh sach da sap xep
36     printf("Danh sach sau khi sap xep:\n");
37     for (i = 0; i < icount; i++)
38         printf("Ten nguoi thu %d : %s\n", i + 1, cptr[i]);
39     getch();
40 }

```

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

Kết quả in ra màn hình

Nhap vao ten nguoi thu 1: Minh
 Nhap vao ten nguoi thu 2: Lan
 Nhap vao ten nguoi thu 3: Anh
 Nhap vao ten nguoi thu 4: Trang
 Nhap vao ten nguoi thu 5: Quan
 Danh sach sau khi sap xep:
 Ten nguoi thu 1: Anh
 Ten nguoi thu 2: Lan
 Ten nguoi thu 3: Minh
 Ten nguoi thu 4: Quan
 Ten nguoi thu 5: Trang

Chạy lại chương trình và thử nhập với dữ liệu khác.

Quan sát kết quả.

☞ Đối với biến khai báo kiểu con trỏ **nhanvien *nv** thì tham chiếu đến phần tử manv: **nv -> manv.**

Ví dụ 3: Nhập và in danh sách nhân viên.

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	/* Danh sach nhan vien */
2	
3	#include <stdio.h>
4	#include <conio.h>
5	#include <stdlib.h>
6	
7	#define MAX 50
8	
9	void main(void)
10	{
11	struct nhanvien
12	{
13	int manv;
14	char hoten[30];
15	};
16	nhanvien snv[MAX];
17	char ctam[10];
18	int i, in;
19	printf("Nhap vao so nhan vien: ");
20	gets(ctam);
21	in = atoi(ctam);
22	
23	//Nhap danh sach nhan vien
24	for(i = 0; i < in; i++)
25	{
26	printf("Nhap vao ma nhan vien thu %d: ", i + 1);
27	gets(ctam);
28	snv[i].manv = atoi(ctam);
29	printf("Nhap vao ho ten: ");
30	gets(snv[i].hoten);
31	}
32	
33	//in danh sach nhan vien
34	for(i = 0; i < in; i++)
35	printf("%5d %s\n", snv[i].manv, snv[i].hoten);
36	getch();
37	}
	F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

☞ **Kết quả in ra màn hình**

Nhap vao so nhan vien: 2 Nhap vao ma nhan vien thu 1: 123 Nhap vao ho ten: Le Thuy Doan Trang Nhap vao ma nhan vien thu 2: 35	Chạy và thử lại chương trình với dữ liệu khác. Quan sát kết quả.
--	---

```

22 {
23     char ctam[10];
24     for(int i = 0; i < in; i++)
25     {
26         printf("Nhap vao ma nhan vien thu %d: ", i + 1);
27         gets(ctam);
28         snv[i].manv = atoi(ctam);
29         printf("Nhap vao ho ten: ");
30         gets(snv[i].hoten);
31     }
32 }
33
34 //Ham in danh sach ra man hinh
35 void output(nhanvien snv[], int in)
36 {
37     for(i = 0; i < in; i++)
38         printf("%5d %s\n", snv[i].manv, snv[i].hoten);
39 }
40
41 void main(void)
42 {
43     nhanvien snv[MAX];
44     char ctam[10];
45     int i, in;
46     printf("Nhap vao so nhan vien: ");
47     gets(ctam);
48     in = atoi(ctam);
49     input(snv, in);
50     output(snv, in);
51     getch();
52 }

```

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

Kết quả in ra màn hình

<pre> Nhap vao so nhan vien: 3 Nhap vao ma nhan vien thu 1: 123 Nhap vao ho ten: Le Thuy Doan Trang Nhap vao ma nhan vien thu 2: 35 Nhap vao ho ten: Le Nguyen Tuan Anh 123 Le Thuy Doan Trang 35 Le Nguyen Tuan Anh _ </pre>	<p>Chạy và thử lại chương trình với dữ liệu khác. Quan sát kết quả.</p>
---	---

Giải thích chương trình

Ở chương trình này ta phải khai báo struct nhanvien là biến toàn cục, vì khi định nghĩa hàm input và output có sử dụng kiểu dữ liệu struct nhanvien.


```

40         break;
41     case CHU_NHAT:
42         fLuongNgay = fLuongCB * PHU_TROI_CN;
43         break;
44     }
45     fTongLuong += fLuongNgay * igio;
46 } while (ngay != CHU_NHAT);
47 printf("Tong luong tuan = %8.2f dong.\n", fTongLuong);
48 getch();
49 }
50
51 //ham chon ngay ke tiep
52 ngay_tuan ngay_mai(ngay_tuan en)
53 {
54     ngay_tuan engay_ke;
55     switch(en)
56     {
57         case CHU_NHAT : engay_ke = THU_HAI;    break;
58         case THU_HAI   : engay_ke = THU_BA;    break;
59         case THU_BA    : engay_ke = THU_TU;    break;
60         case THU_TU    : engay_ke = THU_NAM;   break;
61         case THU_NAM   : engay_ke = THU_SAU;   break;
62         case THU_SAU   : engay_ke = THU_BAY;   break;
63         case THU_BAY   : engay_ke = CHU_NHAT;  break;
64     }
65     return (engay_ke);
66 }

```

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

Kết quả in ra màn hình

```

Nhap vao luong can ban: 250
Nhap vao so gio lam viec tu Thu hai den Chu nhac:
Nhap vao gio lam viec ngay Thu Hai: 7
Nhap vao gio lam viec ngay Thu Ba: 8
Nhap vao gio lam viec ngay Thu Tu: 6
Nhap vao gio lam viec ngay Thu Nam: 7
Nhap vao gio lam viec ngay Thu Sau: 8
Nhap vao gio lam viec ngay Thu Bay: 7
Nhap vao gio lam viec ngay Chu Nhat: 6
Tong luong tuan = 14625.00 dong.
_

```

Hàm chọn ngày kế tiếp trên khá dài, bạn thay từ dòng 54 đến 65 bằng câu lệnh **return (++en > 6 ? 0 : en);** hoặc **return (++en % 7);**

Chạy lại chương trình, quan sát, nhận xét và đánh giá kết quả với dữ liệu khác.

Giải thích chương trình

Ở chương trình này ta phải khai báo struct nhanvien là biến toàn cục, vì khi định nghĩa hàm input và output có sử dụng kiểu dữ liệu struct nhanvien.

👉 Kết quả in ra màn hình

Nhập vào số n: 5 5! = 120. —	Thử lại chương trình với số liệu khác.
------------------------------------	--

Với $n! = 1 * 2 * 3 * \dots * (n-2) * (n-1) * n$,
 ta viết lại như sau: $(1 * 2 * 3 * \dots * (n-2) * (n-1)) * n = n * (n-1)! \dots = n * (n-1) * (n-2)! \dots$

👉 Ta viết lại hàm giai thừa bằng đệ quy như sau:

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	/* Ham tinh giai thua */
2	
3	long giai thua(int in)
4	{
5	int i;
6	if (in == 0)
7	return (1L);
8	else
9	return (in * giai thua(in - 1));
10	}
	F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

👉 Chạy lại chương trình, quan sát, nhận xét và đánh giá kết quả

👉 Giải thích hoạt động của hàm đệ quy giai thừa

Ví dụ giá trị truyền vào hàm giai thừa qua biến in = 5.

- Thứ tự gọi thực hiện hàm giai thừa

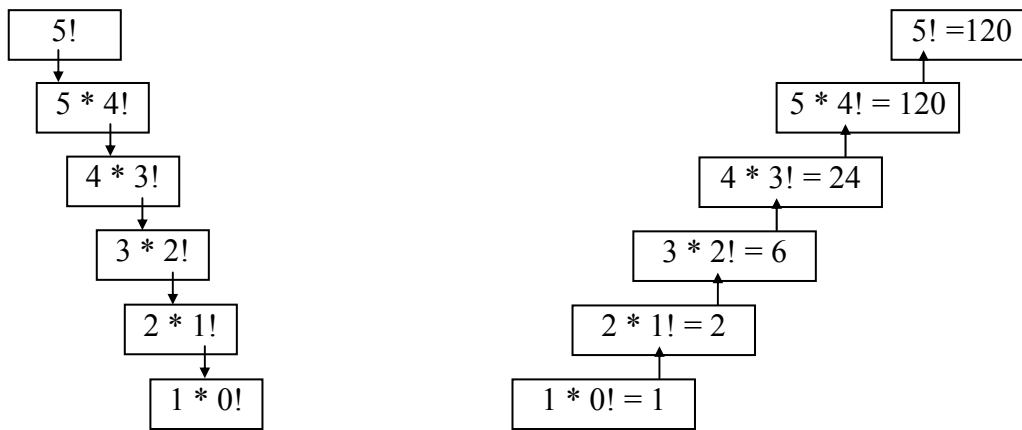
giai thua(in)	return(in * giai thua(in - 1))
5	5 * giai thua(4) = 5 * ?
4	4 * giai thua(3) = 4 * ?
3	3 * giai thua(2) = 3 * ?
2	2 * giai thua(1) = 2 * ?
1	1 * giai thua(0) = 1 * ?

Khi tham số in = 0 thì return về giá trị 1L (giá trị 1 kiểu long). Lúc này các giá trị ? bắt đầu định trị theo thứ tự ngược lại.

- Định trị theo thứ tự ngược lại

giai thua(in)	return(in * giai thua(in - 1))
1	1 * giai thua(0) = 1 * 1 = 1
2	2 * giai thua(1) = 2 * 1 = 2
3	3 * giai thua(2) = 3 * 2 = 6
4	4 * giai thua(3) = 4 * 6 = 24
5	5 * giai thua(4) = 5 * 24 = 120

Kết quả sau cùng ta có $5! = 120$.



Thứ tự gọi đệ quy

Định trị theo thứ tự ngược lại

Ví dụ 2: Dãy số Fibonacci

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ... Bắt đầu bằng 0 và 1, các số tiếp theo bằng tổng hai số đi trước.

Dãy Fibonacci được khai báo đệ quy như sau:

$$\begin{cases} \text{Fibonacci}(0) = 0 \\ \text{Fibonacci}(1) = 1 \\ \text{Fibonacci}(n) = \text{Fibonacci}(n - 1) + \text{Fibonacci}(n - 2) \end{cases}$$

Dòng	File Edit Search Run Compile Debug Project Option Window Help
1	/* Tinh so fibonacci thu n */
2	
3	#include <stdio.h>
4	#include <conio.h>
5	
6	void main(void)
7	{
8	long in;
9	long fibonacci(long);
10	printf("Nhap vao so n: ");
11	scanf("%ld", &in);
12	printf("Fibonacci(%ld) = %ld.\n", in, fibonacci(in));
13	getch();
14	}
15	
16	long fibonacci(long in)
17	{
18	if (in == 0 in == 1)
19	return in;
20	else
21	return fibonacci(in - 1) + fibonacci(in - 2);
22	}
	F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt - F9 Compile F9 Make F10 Menu

Kết quả in ra màn hình

Nhap vao so n: 10 Fibonacci(10) = 55.	Thử lại chương trình với số liệu khác.
--	--

$a / b - c$		$2 * b + 3 * (a - c)$		$a * (b + (c - 4 * 3))$	
$a + c / a$		$c * (b / a)$		$5 * a - 6 / b$	
$a \% b$		$(a * b) \% c$		$5 \% b \% c$	

2. Giả sử x, y, z là biến kiểu float với x = 8.8, y = 3.5 và z = 5.2. Xác định giá trị các biểu thức sau:

$x + y + z$		$z / (y + x)$		$x / y - z * y$	
$5 * y + 6 * (x - z)$		$(z / y) + x$		$2.5 * x / z - (y + 6)$	
x / z		$2 * y / 3 * z$		$5 * 6 / ((x + y) / z)$	
$x \% z$		$2 * y / (3 * z)$		$x / y * (6 + ((z - y) + 3.4))$	

3. Cho chương trình C với các khai báo và khởi tạo các biến như sau:

```
int i = 8, j = 5;
float x = 0.005, y = -0.01;
char c = 'c', d = 'd';
```

Hãy xác định giá trị trả về của các biểu thức sau:

$(3 * i - 2 * j) \% (4 * d - c)$		$c < d$	
$2 * ((i / 4) + (6 * (j - 3)) \% (i + j - 4))$		$x >= 0$	
$(i - 7 * j) \% (c + 3 * d) / (x - y)$		$x < y$	
$-(i + j) * -1$		$j != 6$	
$++i$		$c == 99$	
$i++$		$d != 100$	
$i+++ 5$		$5 * (i + j + 1) > 'd'$	
$++i + 5$		$(3 * x + y) == 0$	
$j--$		$2 * x + (y == 0)$	
$--j$		$!(i < j)$	
$j-- + i$		$!(d == 100)$	
$--j - -5$		$!(x < 0)$	
$++x$		$(i > 0) \&\& (j < 6)$	
$y--$		$(i > 0) !! (j < 5)$	
$i >= j$		$(x > y) \&\& (i > 0) \parallel (j < 5)$	

4. Cho chương trình có các khai báo biến và khởi tạo như sau:

```
int i = 8, j = 5, k;
float x = 0.005, y = -0.01, z;
char a, b, c = 'c', d = 'd';
```

Xác định giá trị các biểu thức gán sau:

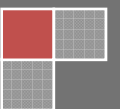
$k = (i + j * 4)$		$z = i / j$		$i \% = j$	
$x = (x + y * 1.2)$		$a = b = d$		$i += (j - 3)$	
$i = j$		$y -= x$		$k = (j == 5) ? i : j$	
$k = (x + y)$		$x *= 2$		$k = (j > 5) ? i : j$	
$k = c$		$i /= j$		$i += j * = i /= 2$	
$i = j = 1.1$		$i += 2$		$a = (c < d) ? c : d$	
$z = k = x$		$z = (x >= 0) ? x : 0$		$i -= (j > 0) ? j : 0$	
$k = z = x$		$z = (y >= 0) ? y : 0$		$i = (i * 9 * (3 + (8 * j / 3)))$	

2007

Lập trình C cho VXL - Cơ bản

Vagam - giotdang

ntuan
BIA
8/15/2007



Lập trình C cho VXL - Cơ bản

I. Giới thiệu

C là một ngôn ngữ khá mạnh và rất nhiều người dùng .Nếu nói số lệnh cơ bản của C thì không nhiều . Nhưng đối với lập trình cho vxl , chúng ta chỉ cần biết số lượng lệnh không nhiều .
Đầu tiên bạn phải làm quen với

Các kiểu toán tử ở C

Các kiểu dữ liệu

Cấu trúc cơ bản của một chương trình

Các cấu trúc điều khiển (chính các tập lệnh)

Cấu trúc điều kiện : if và else

Các cấu trúc lặp

Vòng lặp while

Vòng lặp do while

Vòng lặp for

Lệnh break.

Cấu trúc lựa chọn: switch. case

Biết sử dụng các hàm và chương trình con .

II. Cơ bản C

1. Các chỉ thị trước xử lý của Keil C

// chú thích

```
/****** chú thích*****
```

```
*****
```

```
***** */
```

Đây là dòng chú thích. Tất cả các dòng bắt đầu bằng hai dấu sổ (//) được coi là chú thích mà chúng không có bất kì một ảnh hưởng nào đến hoạt động của chương trình. Chúng có thể được các lập trình viên dùng để giải thích hay bình phẩm bên trong mã nguồn của chương trình. Trong trường hợp này, dòng chú thích là một giải thích ngắn gọn những gì mà chương trình chúng ta làm. Còn trong dấu (/* */) bạn có thể chú thích bao nhiêu dòng tùy thích ,

```
#include <AT89X52.H>  
hoặc  
#include "AT89X52.H"
```

trình biên dịch sẽ gọi file thư viện của 89 ra (cơ bản là 51 cũng như 52)

```
#define bien_thay_the bien
```

Vd

```
#define Congtac P0_6
```

port0.6 được đặt tên là congtac , khi ta gọi tên này trình biên dịch Keil sẽ tự chuyển tới bit quản lý P0_6

Note :cách viết P0_6 phụ thuộc vào từng trình biên dịch , có chương trình thì lại viết là P0.6 , còn keil C viết như cách đầu

```
#define m_left_tien P1_5  
#define m_left_lui P1_4
```

```
#define m_left_forward m_left_tien=0;m_left_lui=1;
```

các bạn chú ý đây là một cách sử dụng marco trong C
khi mình gọi m_left_forward thì chân P1_5 = 0 và P1_4=1

Các viết này gần như cho chúng ta một chương trình con , tuy nhiên không nên quá lạm dụng nó

Một ưu điểm nổi bật của C là các bạn có thể tạo ra các bộ thư viện .

Ví dụ sau là tạo thư viện thuvien.h (đuôi .h bạn có thể tạo bằng cách save as ...*.h ở Keil C).

```
#ifndef _thuvien_H  
#define _thuvien_H
```

```
...//mã chương trình  
#endif
```

2. Các toán tử :

->Toán tử gán (=).

Ex:

```

b = 5;
a = 2 + b;
a = 2 + (b = 5);
a = b = c = 5;

```

-> Các toán tử số học (+, -, *, /, %)

+ cộng
 - trừ
 * nhân
 / chia
 % lấy phần dư (trong phép chia)

-> Các toán tử gán phức hợp (+=, -=, *=, /=, % =, >> =, << =, & =, ^ =, |=)

value += increase; tương đương với value = value + increase;
 a -= 5; tương đương với a = a - 5;
 a /= b; tương đương với a = a / b;
 price *= units + 1; tương đương với price = price * (units + 1);

Tăng và giảm ++ --

```
a++; <=> a+=1; <=> a=a+1;
```

tính chất tiền tố hoặc hậu tố (++a) # (a++) Ex

```

B=3;
B=3;A=++B;
// A is 4, B is 4

```

```

B=3;
A=B++;
// A is 3, B is 4

```

-> Các toán tử quan hệ (==, !=, >, <, >=, <=)

== Bằng
 != Khác
 > Lớn hơn
 < Nhỏ hơn
 > = Lớn hơn hoặc bằng
 < = Nhỏ hơn hoặc bằng

EX

(7 == 5) sẽ trả giá trị false
(6 >= 6) sẽ trả giá trị true

tất nhiên thay vì sử dụng các số, chúng ta có thể sử dụng bất cứ biểu thức nào. Cho a=2, b=3 và c=6

(a*b >= c) sẽ trả giá trị true.
(b+4 < a*c) sẽ trả giá trị false

Chú ý rằng = (một dấu bằng) là hoàn toàn khác với == (hai dấu bằng). (==) nhằm so sánh còn (=) gán giá trị của biểu thức bên phải cho biến ở bên trái .

-> Các toán tử logic (!, &&, ||).

! NOT
&& AND
|| OR

EX:

!(5 == 5) trả về false vì biểu thức bên phải (5 == 5) có giá trị true.
!(6 <= 4) trả về true vì (6 <= 4) có giá trị false.
!true trả về false.
!false trả về true.
((5 == 5) && (3 > 6)) trả về false (true && false).
((5 == 5)

-> Các toán tử thao tác bit (&, |, ^, ~, <<, >>).

& AND Logical AND
| OR Logical OR
^ XOR Logical exclusive OR
~ NOT Đảo ngược bit
<< SHL Dịch bit sang trái
>> SHR Dịch bit sang phải

-> Thứ tự ưu tiên của các toán tử

Thứ tự	Toán tử	Mô tả	Associativity
1	::	scope	Trái

2	() [] -> . sizeof		Trái
3	++ --	tăng/giảm	Phải
	~	Đảo ngược bit	
	!	NOT	
	& *	Toán tử con trỏ	
	(<i>type</i>)	Chuyển đổi kiểu	
	+ -	Dương hoặc âm	
4	* / %	Toán tử số học	Trái
5	+ -	Toán tử số học	Trái
6	<< >>	Dịch bit	Trái
7	< <= > >=	Toán tử quan hệ	Trái
8	== !=	Toán tử quan hệ	Trái
9	& ^	Toán tử thao tác bit	Trái
10	&&	Toán tử logic	Trái
11	? :	Toán tử điều kiện	Phải
12	= += -= *= /= %= >>= <<= &= ^= =	Toán tử gán	Phải
13	,	Dấu phẩy	Trái

3. Các kiểu dữ liệu

Các kiểu biến chuẩn

Type	Bits	Bytes	Range
char	8	1	-128 to +127
unsigned char	8	1	0 to 255
enum	16	2	-32,768 to +32,767
short	16	2	-32,768 to +32,767
unsigned short	16	2	0 to 65,535

int	16	2	-32,768 to +32,767
unsigned int	16	2	0 to 65,535
long	32	4	-2,147,483,648 to +2,147,483,647
unsigned long	32	4	0 to 4,294,697,295

Kiểu dữ liệu trong Keil C

Type	Bits	Bytes	Range
bit	1	0	0 to 1
sbit	1	0	0 to 1
sfr	8	1	0 to 255
sf16	16	2	0 to 65,535

4. Cấu trúc cơ bản của 1 chương trình C

```
//Các chỉ thị tiền định
#include <lcd.h> //Gọi thư viện có sẵn cách viết khác "*.h"
#define led1 PORTA.0 //dùng định nghĩa các biến

char bien1,bien2; //cac bien can dung
int a,b;

void chuongtrinhcon(unsigned int b) // chuong trinh con
{
...
}

int ham(void) // chuong trinh con dang ham
{
....
Return(a);
}
```

```

void main(void) //chương trình chính
{
int a; // khai báo biến dạng số nguyên

chuongtrinhcon();

a = ham();

}

```

Chương trình con là nơi các bạn viết các chương trình nhỏ , rất tiện cho các đoạn lệnh gặp lại nhiều lần . Chương trình con có thể có thể gọi ở trong chương trình chính bất kì đâu .

Hàm là chương trình con trả về cho mình một giá trị.

Cách sử dụng hàm và chương trình con các bạn nên tham khảo thêm quyển kĩ thuật lập trình C để hiểu rõ hơn .

5. Các lệnh cơ bản của C

Cấu trúc điều kiện: if và else

->if (condition) statement

```
if (x == 100) x++;
```

nếu x=100 thì tăng x thêm 1

->if (condition) statement1 else statement2

```
if (x == 100) x++;
else x--;
```

Các cấu trúc lặp

Vòng lặp while .

Dạng của nó như sau:

```
while (expression) statement
```

```
while(1) {};
```

Tạo vòng lặp mãi mãi , rất hay dùng trong lập trình VXL .Chương trình chính sẽ được viết trong dấu ngoặc.

Vòng lặp do-while

Dạng thức:

```
do statement while (condition);
```

```
do
{
```

```
x++; // cho nay cac ban co the viet nhieu cau lenh ,
}
while(x>10)
```

tăng giá trị của x cho đến khi $x > 10$

Chức năng của nó là hoàn toàn giống vòng lặp while chỉ trừ có một điều là điều kiện điều khiển vòng lặp được tính toán sau khi statement được thực hiện, vì vậy statement sẽ được thực hiện ít nhất một lần ngay cả khi condition không bao giờ được thoả mãn. Như vd trên kể cả $x > 10$ thì nó vẫn tăng giá trị 1 lần trước khi thoát

nếu $x=100$ thì tăng x thêm 1 còn không thì giảm x.

Vòng lặp for .

Dạng thức:

for (initialization; condition; increase) statement;
 và chức năng chính của nó là lặp lại statement chừng nào condition còn mang giá trị đúng, như trong vòng lặp while. Nhưng thêm vào đó, for cung cấp chỗ dành cho lệnh khởi tạo và lệnh tăng. Vì vậy vòng lặp này được thiết kế đặc biệt lặp lại một hành động với một số lần xác định. Cách thức hoạt động của nó như sau:

- 1, initialization được thực hiện. Nói chung nó đặt một giá trị ban đầu cho biến điều khiển. Lệnh này được thực hiện chỉ một lần.
- 2, condition được kiểm tra, nếu nó là đúng vòng lặp tiếp tục còn nếu không vòng lặp kết thúc và statement được bỏ qua.
- 3, statement được thực hiện. Nó có thể là một lệnh đơn hoặc là một khối lệnh được bao trong một cặp ngoặc nhọn.
- 4, Cuối cùng, increase được thực hiện để tăng biến điều khiển và vòng lặp quay trở lại bước 2.

Phần khởi tạo và lệnh tăng không bắt buộc phải có. Chúng có thể được bỏ qua nhưng vẫn phải có dấu chấm phẩy ngăn cách giữa các phần. Vì vậy, chúng ta có thể viết for (; $n < 10$;) hoặc for (; $n < 10$; $n++$).

Bằng cách sử dụng dấu phẩy, chúng ta có thể dùng nhiều lệnh trong bất kì trường nào trong vòng for, như là trong phần khởi tạo. Ví dụ chúng ta có thể khởi tạo một lúc nhiều biến trong vòng lặp:

```
for ( n=0, i=100 ; n!=i ; n++, i-- )
{
// cái gì ở đây cũng được...
}
```

Ví dụ điển hình nhất trong lập trình VXL

```
void delayms(int n)
{
```

```
int i,j; // khai bao bien chi trong chuong trinh con
for (i=0;i<n;i++)
for (j=0;j<1500;j++) { } // tham so j tuy thach anh toc do vxl ma cac
```

```

    //ban thay doi cho phu hop
}

```

Các lệnh rẽ nhánh và lệnh nhảy

Lệnh break.

Sử dụng break chúng ta có thể thoát khỏi vòng lặp ngay cả khi điều kiện để nó kết thúc chưa được thỏa mãn. Lệnh này có thể được dùng để kết thúc một vòng lặp không xác định hay buộc nó phải kết thúc giữa chừng thay vì kết thúc một cách bình thường. Ví dụ, chúng ta sẽ dừng việc đếm ngược trước khi nó kết thúc:

Lệnh continue.

Lệnh continue làm cho chương trình bỏ qua phần còn lại của vòng lặp và nhảy sang lần lặp tiếp theo. Ví dụ chúng ta sẽ bỏ qua số 5 trong phần đếm ngược:

Lệnh goto.

Lệnh này cho phép nhảy vô điều kiện tới bất kì điểm nào trong chương trình. Nói chung bạn nên tránh dùng nó trong chương trình C++. Tuy nhiên chúng ta vẫn có một ví dụ dùng lệnh goto để đếm ngược:

Hàm exit.

Mục đích của exit là kết thúc chương trình và trả về một mã xác định. Dạng thức của nó như sau void exit (int exit code);

exit code được dùng bởi một số hệ điều hành hoặc có thể được dùng bởi các chương trình gọi.

Theo quy ước, mã trả về 0 có nghĩa là chương trình kết thúc bình thường còn các giá trị khác 0 có nghĩa là có lỗi.

các lệnh trên mình chủ yếu chỉ dùng lệnh break để thoát khỏi vòng lặp . Các lệnh khác thường rất ít được sử dụng

Đây là 1 đoạn code nhỏ mình trích ra từ chương trình của mình

```

while(1)
{
lcd_gotoxy(5,0);lcd_putsf(" Run Thuan ");
thuan();
if(!enter) { lcd_clear();
lcd_putsf("DA DUNG ");
stop();break;}
}

```

Cấu trúc lựa chọn: switch.

Cú pháp của lệnh switch hơi đặc biệt một chút. Mục đích của nó là kiểm tra một vài giá trị hằng cho một biểu thức, tương tự với những gì chúng ta làm ở đầu bài này khi liên kết một vài lệnh if và else if với nhau. Dạng thức của nó như sau:

Code:

```
switch (expression) {
```

```

case constant1:
    block of instructions 1
    break;
case constant2:
    block of instructions 2
    break;
.
.
.
default:
    default block of instructions
}

```

Nó hoạt động theo cách sau: switch tính biểu thức và kiểm tra xem nó có bằng constant1 hay không, nếu đúng thì nó thực hiện block of instructions 1 cho đến khi tìm thấy từ khoá break, sau đó nhảy đến phần cuối của cấu trúc lựa chọn switch.

Còn nếu không, switch sẽ kiểm tra xem biểu thức có bằng constant2 hay không. Nếu đúng nó sẽ thực hiện block of instructions 2 cho đến khi tìm thấy từ khoá break.

Cuối cùng, nếu giá trị biểu thức không bằng bất kì hằng nào được chỉ định ở trên (bạn có thể chỉ định bao nhiêu câu lệnh case tùy thích), chương trình sẽ thực hiện các lệnh trong phần default: nếu nó tồn tại vì phần này không bắt buộc phải có.

nếu nút enter được bấm thì chương trình sẽ thoát ra khỏi vòng lặp while .

III. C cho 8051

1. Keil Variable Extensions

data địa chỉ trực tiếp MOV A, 07Fh

idata địa chỉ gián tiếp MOV R0, #080h

MOV A, R0

xdata bộ nhớ ngoài MOVX @DPTR

code bộ nhớ chương trình MOVC @A+DPTR

VD

```
unsigned int data bien = 0; // them data vao khai bao kieu bien
```

Chú ý rằng , bạn có thể không cần khai báo cụ thể , chỉ cần

```
unsigned int checksum = 0;
```

2. Địa chỉ ngắt

Interrupt	Vector address	Interrupt number
External 0	0003h	0

Timer 0	000Bh	1
External 1	0013h	2
Timer 1	001Bh	3
Serial	0023h	4

```
org      00h
ljmp main
org 0003h
ljmp ngat0
org 30h
```

```
main:          // chương trình chính
```

```
...
```

```
Here: sjmp Here // vòng lặp vô tận
```

```
ngat0:
```

```
...
```

```
reti
```

code C

Code:

```
void main          // chương trình chính
{
...
while(1)          //vòng lặp vô tận sau khi thực hiện xong công việc
}
void ngat0(void) interrupt 0 // chương trình ngắt
{
...
}
```

3. Một ví dụ hoàn thiện về lập trình C cho 8051

```
#include<AT89X52.h>
#include<stdio.h>
#define strai3 P0_7
#define strai2 P0_6
#define strai1 P0_5
#define strai0 P0_4
#define sphai0 P0_3
#define sphai1 P0_2
#define sphai2 P0_1
#define sphai3 P0_0
//////////
#define mtraif P1_0
#define mtraib P1_1
#define mphaif P1_2
#define mphaib P1_3
#define dc1f P1_6
#define dc1b P1_7
```



```

#define dc2f          P1_4
#define dc2b          P1_5
//////////////////////////////////////////////////
#define f_an0         P2_0
#define f_an1         P2_1
#define f_an2         P2_2
#define f_an3         P2_3
//////////////////////////////////////////////////
#define dc3f          P2_4
#define dc3b          P2_5
//////////////////////////////////////////////////
#define start         P3_6
#define ctht1_batdau  P3_1
#define ctht2_ketthuc P3_2
#define ctht3         P3_3
#define ctht4         P3_4
#define ctht5         P3_5
#define ctht6         P3_6
//////////////////////////////////////////////////
#define tien          1
#define lui           0
//////////////////////////////////////////////////
#define v_cham_trai   50
#define v_cham_phai   50
#define delta_v_cham_trai 20
#define delta_v_cham_phai 20
#define v_nhanh_trai 100
#define v_nhanh_phai 100

#define v_quay_trai   40
#define v_quay_phai   60

//////////////////////////////////////////////////
//vach trang sensor=1 vach xanh sensor=0
int t=0,i=0,j=0,k=0;
int vtrai=100,vphai=100;
int PWMC=0;
int dem=0;
int dirtrai=tien,dirphai=tien;
int phuongan=0;
//////////////////////////////////////////////////
void dithang(int v_left,int v_right);
void dithangcham(int v_left,int v_right);
void stop(void);
void quayphai(void);
void quaytrai(void);
void ragach(void);
void pwm(void) interrupt 1 ;
void khoitao(void);
void hanhtrinh(void);

void ham(int time,int trai,int phai);
void ham2(void);
//=====
void main(void)
{
    khoitao();
    khoidongthang();
    hanhtrinh();
    ragach();
    stop();
}
//=====
void khoitao(void)
{
    TMOD=0x02;
    TH0=0xE1; TR0=1; IE=0x82;
    P1=0x00;P2=0x00;P3=0x00;
    while(!start) stop();
}
//=====
void khoidongthang(void)

```

```

{
    for(i=0;i<300;i++)
    {
        dirtrai=tien;dirphai=tien;
        vtrai=40; vphai=40;
    }
}
//=====
void khoidongquay(void)
{
    dirtrai=tien;dirphai=tien;
    vtrai=90; vphai=80;
    for(i=1;i<=50;i++);
    vtrai=52; vphai=40;
    for(i=1;i<=580;i++);
    ham(90,20,20);
    do
    {
        dirtrai=tien;dirphai=lui;
        vtrai=33;vphai=10;
    }
    while(!strai3);
    for(i=1;i<=40;i++)
    {
        dirtrai=tien;dirphai=lui;
        vtrai=33; vphai=9;
    }
    do
    {
        dirtrai=lui;dirphai=tien;
        vtrai=15;vphai=32;
    }
    while(!strai0);
    ham(100,20,20);
    dirtrai=tien;dirphai=tien;
}
//=====
void quayphai(void)
{
    int ktra=0;
    dirtrai=tien;dirphai=lui;
    do
    {
        vtrai= v_quay_trai;vphai= v_quay_phai;
        if (sphai3) ktra++;
    }
    while(ktra<20);
    stop();
    for(i=1;i<1000;i++){};
    dirtrai=tien; dirphai=tien;
}
//=====
void quaytrai(void)
{
    int ktra=0;
    dirtrai=lui;dirphai=tien;
    do
    {
        vtrai= 60; vphai= 40;
        if (strai3)ktra++;
    }
    while(ktra<20);
    stop();
    for(i=1;i<1000;i++){};
    dirtrai=tien; dirphai=tien;
}
//+++++
//vach trang sensor=1 vach xanh sensor=0
void stop(void)
{
    vtrai=0; vphai=0;
}

```

```

}
//=====
void dithang(int v_left,int v_right)
{
    if(!(strai0||strai1||strai2||strai3||sphai0||sphai1||sphai2||sphai3)) // di thang
    {
        dirphai=tien;dirtrai=tien;
        vtrai=v_left;vphai=v_right;
    }
    else
        if((strai0&&strai1&&sphai0&&sphai1))
        {
            j=0;
            for(i=1;i<=15;i++)
                if((strai0&&strai1&&sphai0&&sphai1)) j++;
            if(j>=10) { dem=dem+1;t=1 ;}
            else t=0;
            while(t)//cho qua vach trang
            {
                { //di thang
                    dirphai=tien;dirtrai=tien;
                    vtrai=40;vphai=40;
                }
                k=0;
                for(i=1;i<=20;i++)
                    if(!(strai3||strai2||sphai3||sphai2)) k++;
                if(k>15) t=0;
            }
        }
    else if ((sphai2||sphai3)) //re phai lon
    {
        dirtrai=tien;dirphai=tien;
        vtrai=60;vphai=0;}
    else if ((strai0||strai1))
    {
        dirtrai=lui;dirphai=tien;
        vtrai=1;vphai=60;
    }
    else if ((sphai0||sphai1))
    {
        dirtrai=tien;dirphai=lui;
        vtrai=60;vphai=1;
    }
    else if ((strai2||strai3))
    {
        dirtrai=tien;dirphai=tien;
        vtrai=0;vphai=60;}
}
//=====
void ragach(void)
{
    while(!ctht2_ketthuc)
    {
        dirtrai=tien;dirphai=tien;
        vtrai=25;vphai=25;
        dc1f=1;dc1b=0;
    }
    dc1f=0;dc1b=0;
    //-----
    for(t=1;t<=5;t++)
    {
        for(i=1;i<=10000;i++){
            dirtrai=tien;dirphai=tien;
            vtrai=25;vphai=25;}
    }
}

```

```

        for(i=1;i<=10000;i++){
            dirtrai=lui;dirphai=lui;
            vtrai=0;vphai=0;}
        }
//-----
        dirtrai=lui;dirphai=lui;
        vtrai=25;vphai=25;
        for(i=1;i<=2000;i++);
    }

////////////////////////////////////
void pwm(void) interrupt 1
{
    PWMC++;
    if(PWMC==100) PWMC=0;
    if(PWMC<vtrai)
        {
            mtraif=dirtrai;
            mtraib=!dirtrai;
        }
    else
        {
            mtraif=0;
            mtraib=0;
        }
    if(PWMC<vphai)
        {
            mphaif=dirphai;
            mphaib=!dirphai;
        }
    else
        {
            mphaif=0;
            mphaib=0;
        }
}
void ham(int time,int trai,int phai)
{
    int g;

    for(g=0;g<=time;g++)
    {
        vphai=phai;vtrai=trai;
        dirtrai=tien;dirphai=tien;
        dirtrai=lui;dirphai=lui;
    }
    dirtrai=tien;dirphai=tien;
    vphai=0;vtrai=0;
}

```

Chúc các bạn học lập trình C cho vi xử lý thật nhanh nhé . Đọc phần lý thuyết cơ bản sau đó đọc bài ví dụ cuối cùng .
Nếu bạn còn gì chưa hiểu , hãy post lên để nhóm vagam giúp các bạn .

Thân

Lê Ngọc Tuấn – giotdang1985@yahoo.com



Bài tập Lập trình C# căn bản

KIỂU SỐ

Lệnh nhập 1 số từ bàn phím

```
Console.WriteLine("Câu thông báo...");
Int So = Int.Parse(Console.ReadLine());
```

```
Int so01 = 23;
Int so02 = 7;
So01 += so02; → so01 = s01 + 1
Console.WriteLine("so : {0}", so01);
Console.ReadLine();
```

```
-----
Int so = 10;
so + +;      so = so + 1
Console.WriteLine("so : {0}", so);
Console.ReadLine();
```

Lưu ý:

So + +

```
Console.WriteLine("So : {0}, so + + );
                               Console.WriteLine("So :{0}", so);
                               So = so + 1
```

+ + So

```
Console.WriteLine("So : {0}, + + so );
                               So = so + 1
                               Console.WriteLine("So :{0}", so);
```

Int so01 = 23; Int so02 = 7;

Tính: So01 += so02 - -;

So01 = so01 + so02;

So02 = so02 - 1;

KQ: 30/6

Int so01 = 23; Int so02 = 7;

Tính: So01 += - - so02;

So02 = so02 - 1;

So01 = so01 + so02;

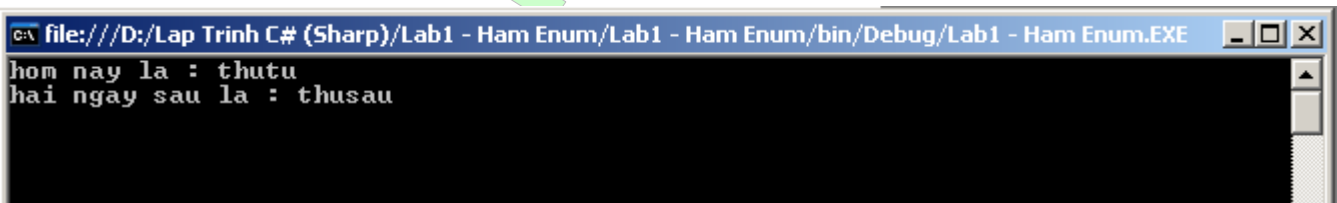
KQ: 29/6

LAB THỰC HÀNH**Lab 1: Hàm enum**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace lab01
{
    class Program
    {
        enum ngaytrongtuan
        {
            thuhai,
            thuba,
            thutu,
            thunam,
            thusau,
            thubay,
            chunhat
        }

        static void Main(string[] args)
        {
            Console.WriteLine("hom nay la : {0}", ngaytrongtuan.thutu);
            Console.WriteLine("hai ngay sau la :
{0}", ngaytrongtuan.thutu + 2);
            Console.ReadLine();
        }
    }
}
```

**Lab 2 – Kiểu số**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace lab02
{
    class Program
    {
        static void Main(string[] args)
        {
            int so = 10;
            so += 5; //so = so + 5 ;
            Console.WriteLine("gia tri cua so la : {0}", so);
            Console.ReadLine();
        }
    }
}
```

```

        int so01 = 23;
        int so02 = 7;
        so01 -= so02;
        so01 += so01; // so01 = so01 + so02
        Console.WriteLine("gia tri cua so01 la : {0}",so01);
        Console.ReadLine();
    }
}
}

```

```

C:\file:///D:/Lap Trinh C# (Sharp)/Lab2 - Kieu so/Lab2 - Kieu so/bin/Debug/Lab2 - Kieu so.EXE
gia tri cua so la : 15
gia tri cua so01 la : 32

```

Lab 3 – Kiểu số

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace lab03
{
    class Program
    {
        static void Main(string[] args)
        {
            int so = 10;
            so++;
            Console.WriteLine("so : {0}",so++); //viet cw truoc roi +
sau
            Console.WriteLine("so : {0}",++so); //+ truoc roi viet cw
sau
            Console.ReadLine();
        }
    }
}

```

```

C:\file:///D:/Lap Trinh C# (Sharp)/Lab3 - Kieuso/Lab3 - Kieuso/bin/Debug/Lab3 - Kieuso.EXE
so : 11
so : 13

```

Lab 4 - Main

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace lab04
{
    class Program
    {

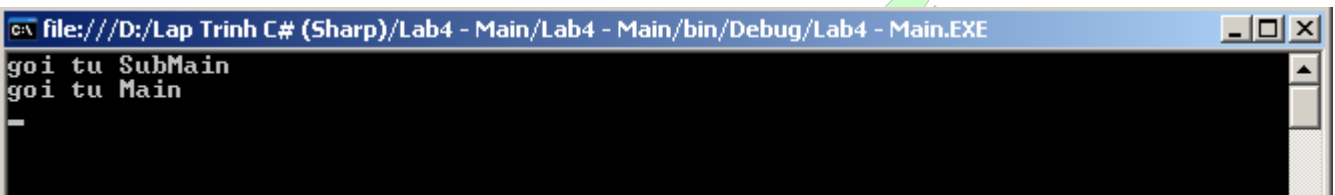
```



```

static void Main(string[] args)
{
    // re nhanh chuong trinh : submain se viet truoc , roi toi
main
    // khi khong goi "Submain" thi chuong trinh k chay
    SubMain();
    Console.WriteLine("goi tu Main");
    Console.ReadLine();
}
static void SubMain()
{
    Console.WriteLine("goi tu SubMain");
}
}
}

```



```

C:\file:///D:/Lap Trinh C# (Sharp)/Lab4 - Main/Lab4 - Main/bin/Debug/Lab4 - Main.EXE
goi tu SubMain
goi tu Main
-

```

Lab 5 - Tính

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace lab05
{
    class Program
    {
        static void Main(string[] args)
        {
            PTbac1();
            TinhCVDTHinhCN();
            TinhCVDTHinhTron();
            DoiKgsangPound();
            DoiPoundsangKg();
            Tinhsogiay();
            Console.WriteLine("-----");
            Console.ReadLine();
        }
        static void TinhCVDTHinhCN()
        {
            Console.WriteLine("-----");
            Console.WriteLine("CHUONG TRINH TINH CHU VI VA DIEN TICH HINH CHU NHAT");
            Console.WriteLine("-----");
            Console.WriteLine("Nhap chieu dai hinh chu nhatt");
            float chieudai = float.Parse(Console.ReadLine());

```

```

        Console.WriteLine("Nhap chieu rong hinh chu nhat");
        float chieurong = float.Parse(Console.ReadLine());
        Console.WriteLine("Chu vi hinh chu nhat la : {0}\nDien
tich hinh chu nhat la : {1}", (chieudai + chieurong) * 2, chieurong *
chieudai);
        Console.ReadLine();
    }
    static void PTbac1()
    {
        Console.WriteLine("-----
-----\nCHUONG TRINH TINH PHUONG TRINH BAC NHAT\n-----
-----");
        Console.WriteLine("Phuong trinh ax + b = 0 ");
        Console.WriteLine("Nhap he so a: ");
        float a = float.Parse(Console.ReadLine());
        Console.WriteLine("Nhap he so b: ");
        float b = float.Parse(Console.ReadLine());
        Console.WriteLine("Nghiem x cua phuong trinh la: ");
        Console.WriteLine("x:{0}", -b / a);
        Console.ReadLine();
    }
    static void TinhCVDTHinhTron()
    {
        Console.WriteLine("-----
-----\nCHUONG TRINH TINH CHU VI VA DIEN TICH HINH TRON\n-----
-----");
        Console.WriteLine("nhap ban kinh duong tron");
        float bankinh = float.Parse(Console.ReadLine());
        const float pi = 3.1416f;
        Console.WriteLine("Chu vi hinh tron la :{0}", bankinh * 2
* pi);
        Console.WriteLine("Dien tich hinh tron la : {0}", bankinh
* bankinh * pi);
        Console.ReadLine();
    }
    static void DoiKgsangPound()
    {
        Console.WriteLine("-----
-----\nCHUONG TRINH DOI KG SANG POUND \n-----
-----");
        Console.WriteLine("Nhap so kg can doi sang pound");
        float kg = float.Parse(Console.ReadLine());
        const float hesokg = 2.2046f;
        Console.WriteLine("So pound se la : {0} P", kg * hesokg);
        Console.ReadLine();
    }
    static void DoiPoundsangKg()
    {
        Console.WriteLine("=====");
        Console.WriteLine("    CHUONG TRINH DOI POUND SANG KG    ");
        Console.WriteLine("=====");
        Console.WriteLine("Nhap so pound can doi sang kg");
        float pound = float.Parse(Console.ReadLine());

```

```

        const float hesopound = 0.4535f;
        Console.WriteLine("So kg la : {0}kg", pound * hesopound);
        Console.ReadLine();
    }
    static void Tinhhsogiay()
    {
        Console.WriteLine("=====");
        Console.WriteLine("        CHUONG TRINH TINH SO GIAY        ");
        Console.WriteLine("=====");
        Console.WriteLine("Nhap vao so gio");
        float sogio = float.Parse(Console.ReadLine());
        Console.WriteLine("Nhap vao so phut");
        float sophut = float.Parse(Console.ReadLine());
        Console.WriteLine("Nhap vao so giay");
        float sogiay = float.Parse(Console.ReadLine());
        Console.WriteLine("So giay la : {0}giay", (sogio * 3600) +
(sophut * 60) + sogiay);
        Console.ReadLine();
    }
}
}
}

```

```

file:///D:/Lap Trinh C# (Sharp)/Lab5 - Tinh/Lab5 - Tinh/bin/Debug/Lab5 - Tinh.EXE
-----
CHUONG TRINH TINH PHUONG TRINH BAC NHAT
-----
Phuong trinh ax + b = 0
Nhap he so a:
2
Nhap he so b:
-6
Nghiem x cua phuong trinh la:
x:3
-----
CHUONG TRINH TINH CHU VI VA DIEN TICH HINH CHU NHAT
-----
Nhap chieu dai hinh chu nhat
3
Nhap chieu rong hinh chu nhat
4
Chu vi hinh chu nhat la : 14
Dien tich hinh chu nhat la : 12
-----
CHUONG TRINH TINH CHU VI VA DIEN TICH HINH TRON
-----
nhap ban kinh duong tron
2.314
Chu vi hinh tron la :14.53932
Dien tich hinh tron la : 16.822
-----
CHUONG TRINH DOI KG SANG POUND
-----
Nhap so kg can doi sang pound
30
So pound se la : 66.138 P
-----
CHUONGTRINH DOI POUND SANG KG
-----
Nhap so pound can doi sang kg
65
So kg la : 29.4775kg

```

Lab6 – Tổng hai số

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Lab6
{
    class Program
    {
        static void Main(string[] args)
        {
            //int so01 = 7;
            Console.Write("Nhap so thu 1: ");
            int so01 = int.Parse(Console.ReadLine());
            //int so02 = 3;
            Console.Write("Nhap so thu 2: ");
            int so02 = int.Parse(Console.ReadLine());
            Console.WriteLine("Tong hai so {0} va {1} la {2}", so01,
so02, so01 + so02);
            Console.ReadLine();
        }
    }
}

```

Lab 7 – Đổi Inch

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Lab7
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.Write("Viet so inch can doi:");
            float soinch = float.Parse(Console.ReadLine());
            const float hesocm = 2.54f;
            Console.WriteLine(" {0}inch = {1}cm", soinch,soinch*
hesocm);
            Console.ReadLine();
        }
    }
}

```

KIỂU LOGIC

Khai báo: Bool

VD: Bool b = true;

1. Phủ định: ký hiệu !

Ví dụ: bool b = true;

b = !b ; → không phủ định

Ghi chú: 2 lần phủ định → không phủ định

2. Phép And (&&) – Or (||):

A	B	A & B	A or B
Đ	Đ	Đ	Đ
Đ	S	S	S
S	Đ	S	S
S	S	S	S

3. Sự kết hợp giữa phép phủ định và And, Or:

Bool a = true;

Bool b = false;

a = a && !b; →

b: false

!b: true

a && !b = true && true → true

Quan sát bảng sau:

A	B	A & B	A or B	!(A & B)	!(A or B)	!A	!B	!A or !B	!A & !B
Đ	Đ	Đ	Đ	S	S	S	S	S	S
Đ	S	S	Đ	Đ	S	S	Đ	Đ	S
S	Đ	S	Đ	Đ	S	Đ	S	Đ	S
S	S	S	Đ	Đ	Đ	Đ	Đ	Đ	Đ



Ta có công thức:

!(A and B) = !A or !B
!(A or B) = !A and !B

!A and B = !(A or !B)

A or !B = !(!A) or !B = !(!A and B)

!(A and !B) = !A or B

4. Phép so sánh trong kiểu số:

- Phép so sánh hay đại lượng so sánh trong kiểu số là 1 phép toán 2 ngôi mã kết quả là kiểu bool.

VD: $5 > 7$ trả ra giá trị False

- Có 6 phép so sánh: $>$, $<$, $==$, $>=$, $<=$, $!=$ ($<>$ hay không bằng)

5. Phép phủ định trong phép so sánh

Ta có: $!>$ là $<=$

$!(!=)$ là $==$

$!(<=)$ là $>$

6. Phép toán 3 ngôi:

(biểu thức bool) ? Giá trị khi đúng : Giá trị khi sai

Lab 8 - Bool

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace BaiLab1
{
    class Program
    {
        static void Main(string[] args)
        {
            bool b = true;
            Console.WriteLine("b:{0}, b", b);
            b = !b;
            Console.WriteLine("!b:{0}", b);
            Console.ReadLine();
        }
    }
}
```

```
c:\ file:///D:/Lap Trinh C# (Sharp)/Lab8 - Bool/Lab8 - Bool/bin/Debug/Lab8 - Bool.EXE
b:True, b
!b:False
```

Lab 9 – Bool a - b

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace BaiLab2
{
    class Program
    {
        static void Main(string[] args)
        {
```

```

        bool a = true;
        bool b = true;
        Console.WriteLine("A\tB\tA and B\t\tA or B");
        Console.WriteLine("{0}\t{1}\t{2}\t\t{3}", a, b, a && b, a
|| b);

        b = false;
        Console.WriteLine("{0}\t{1}\t{2}\t\t{3}", a, b, a && b, a
|| b);

        a = false;
        b = true;
        Console.WriteLine("{0}\t{1}\t{2}\t\t{3}", a, b, a && b, a
|| b);

        b = false;
        Console.WriteLine("{0}\t{1}\t{2}\t\t{3}", a, b, a && b, a
|| b);

        Console.ReadLine();
    }
}
}

```

```

c:\ file:///D:/Lap Trinh C# (Sharp)/Lab9 - Bool a and or b/Lab9 - Bool a and or b/bin/Debug/Lab9 - Bool a ...
A      B      A and B      A or B
True   True   True        True
True   False  False       True
False  True   False       True
False  False  False       False

```

Lab 10 – Bool1

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace BaiLab3
{
    class Program
    {
        static void Main(string[] args)
        {
            int so01 = 25;
            int so02 = 12;
            bool a = so01 > so02;
            bool b = so01 <= so02;
            Console.WriteLine("a:{0} - b: {1}", a, b);
            Console.WriteLine("a:{0} - !a: {1}", a, !a);
            Console.ReadLine();
        }
    }
}

```

```

c:\ file:///D:/Lap Trinh C# (Sharp)/Lab10 - Bool1/Lab10 - Bool1/bin/Debug/Lab10 - Bool1.EXE
a:True - b: False
a:True - !a: False

```

Lab 11 – Bool 2

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace BaiLab4
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Nhap vao so a:");
            int a = int.Parse(Console.ReadLine());
            Console.WriteLine("Nhap vao so b:");
            int b = int.Parse(Console.ReadLine());

            string s = (a >= b) ? "a lon hon hay bang b" : "a nho hon
b";

            Console.WriteLine(s);
            Console.ReadLine();
        }
    }
}

```

Lab 12 - CMND

Nhập vào 1 tuổi. Hãy in ra màn hình tuổi đó có hợp lệ để làm CMND hay không? Biết rằng tuổi làm CMND phải từ 15 tuổi trở lên?

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Lab12
{
    class Program
    {
        static void Main(string[] args)
        {
            TuoihoplelamCMND();
            Cachkhac();
            Console.WriteLine("\nChi tiet: Sangit.sgu@gmail.com\n");
            Console.ReadLine();
        }
        static void TuoihoplelamCMND()
        {
            Console.Write(" Xin cho biet tuoi cua ban: ");

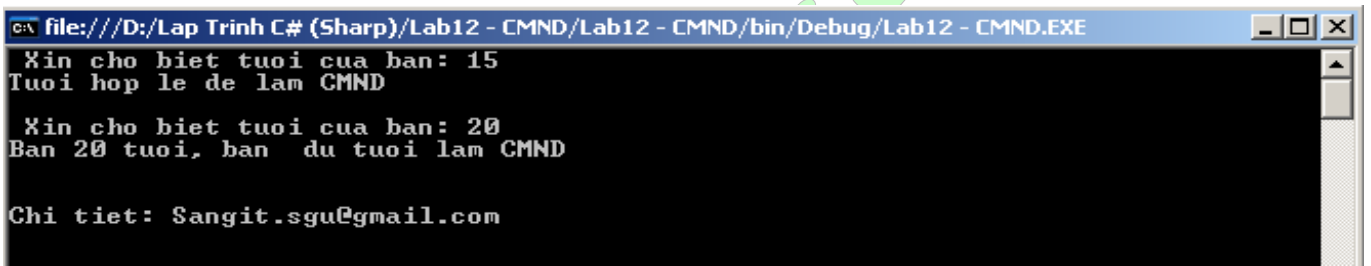
```



```

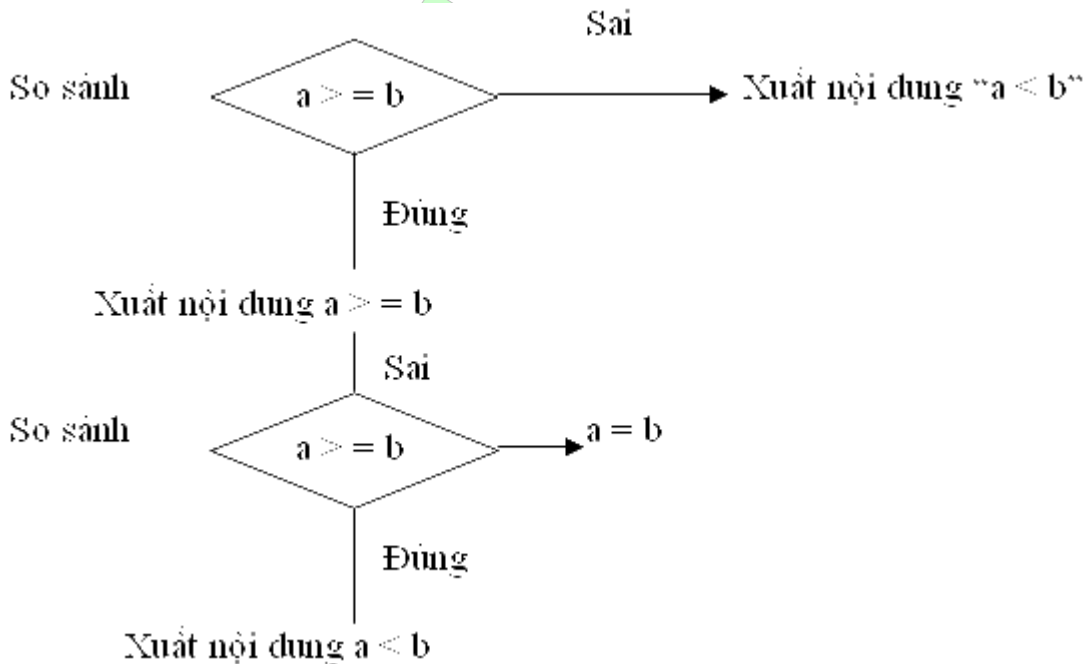
        int tuoi = int.Parse(Console.ReadLine());
        string Kq = tuoi >= 15 ? "Tuoi hop le de lam CMND" :
"Khong du tuoi lam CMND";
        Console.WriteLine(Kq);
        Console.ReadLine();
    }
    static void Cachkhac()
    {
        Console.Write(" Xin cho biet tuoi cua ban: ");
        int tuoi = int.Parse(Console.ReadLine());
        //string Kq = tuoi >= 15 ? "Tuoi hop le de lam CMND" :
"Khong du tuoi lam CMND";
        //Console.WriteLine(Kq);
        Console.WriteLine("Ban {0} tuoi, ban {1} du tuoi lam CMND
", tuoi, tuoi >= 15?"": "khong ");
        Console.ReadLine();
    }
}
}
}

```



Lab13 – So sánh

Nhập vào 2 số a, b. Hãy cho biết a > b, a = b hay a < b



```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace Lab14
{

```

```

class Program
{
    static void Main(string[] args)
    {
        Console.Write("Nhập vào số a: ");
        int a = int.Parse(Console.ReadLine());
        Console.Write("Nhập vào số b: ");
        int b = int.Parse(Console.ReadLine());
        string kq = a >= b ? a > b ? " a > b" : "a = b" : "a < b";
        Console.WriteLine(kq);
        Console.ReadLine();
    }
}

```

Lab 14 – Trả giá trị chuỗi

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Bai14
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.Write("Nhập vào 1 chuỗi: ");
            string s = Console.ReadLine();
            int i = s.Length; //Length tra ra chieu dai chuỗi s
            Console.WriteLine("Chuỗi có chiều dài là: {0} ký tự", i);
            s = s.ToUpper(); //Ham ToUpper tra ra chuỗi Hoa
            Console.WriteLine(s);
            s = s.ToLower(); //Ham ToLower tra ra chuỗi Thuong
            Console.WriteLine(s);
            Console.ReadLine();
        }
    }
}

```

Lab 15 – Cắt, bỏ chuỗi

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Bai8

```

```

{
    class Program
    {
        static void Main(string[] args)
        {
            Console.Write("Nhap vao chuoai s (co chieu dai > 2 ky tu):
");
            string s = Console.ReadLine();
            s = s.Insert(2, "--abcdef--"); //chen chuoai "--
abcdef--" vao chuoai s tai sau vi tri 2
            Console.WriteLine("Chuoai sau khi duoc them vo: {0}", s);
            s = s.Remove(3, 5); //Xoa chuoai trong s tu vi tri
thu 3 va co chieu dai 5 ky tu
            Console.WriteLine("Chuoai sau khi bi xoa: {0}", s);
            string sleft = s.PadLeft(30, '.'); //Them '.' vao ben
traai cho du 30 ky tu
            Console.WriteLine(sleft);
            string sright = s.PadRight(30, '.'); //Them '.' vao ben
phai cho du 30 ky tu
            Console.WriteLine(sright);
            string s1 = "                Nguyen Van Ba                ";
            s1 = s1.Trim(); //Trim la ham cat bo nhung khoang
trang ben trai va ben phai cua chuoai
            Console.WriteLine(s1);
            Console.ReadLine();
        }
    }
}

```

```

C:\ file:///D:/Lap Trinh C# (Sharp)/Lab15 - Cat bo chuoai/Lab15 - Cat bo chuoai/bin/Debug/Lab15 - Cat bo c...
Nhap vao chuoai s <co chieu dai > 2 ky tu>: NGUYEN VAN SANG
Chuoai sau khi duoc them vo: NG--abcdef--UYEN VAN SANG
Chuoai sau khi bi xoa: NG-ef--UYEN VAN SANG
.....NG-ef--UYEN VAN SANG
NG-ef--UYEN VAN SANG.....
Nguyen Van Ba

```

* **Bài tập 1:** Cho a, b, c là ba số bất kỳ. Hãy bỏ dấu ngoặc của các phép so sánh sau:

1. $!(a \leq b) \text{ and } (b < c)$ $a < b \text{ and } (b < c)$
2. $!((a > b) \text{ and } !(b > c))$ $!((a > b) \text{ and } (b \leq c))$ $a \leq b \text{ and } b > c$
3. $(a < b) \text{ or } !(b < c)$ $a < b \text{ or } b < c$
4. $(a > b) \text{ and } !(b \geq c)$ $c > b \text{ and } b < c$
5. $!((a \geq b) \text{ or } (c == b))$ $(a < b) \text{ or } c \neq b$

* **Bài tập 2:** Viết chương trình xuất bảng 10 cột với các trường hợp True/False của A, B.

* **Bài tập 3:**

- 3.1. Nhập vào 1 số nguyên n. Xuất ra số đó là số âm hay dương.
- 3.2. Nhập vào 1 số nguyên n. Xuất ra số đó là số chẵn hay lẻ.
- 3.3. Tuổi lao động được quy định là từ 18 – 65 đối với Nam. Hãy nhập vào tuổi, thông báo tuổi đó có nằm trong độ tuổi lao động hay không?
- 3.4. Viết công thức nhập vào Họ Tên, điểm Toán, Lý, Hóa. Hãy xuất ra tên học sinh được in Hoa và điểm trung bình 1 $(DTB1) = (\text{Điểm T} + \text{L} + \text{H}) / 3$. Tính điểm trung bình 2 $(DTB2) = (\text{điểm T} * 2 + \text{L} + \text{H}) / 4$
- 3.5. Viết công thức nhập vào bậc lương, ngày công, phụ cấp. Tính tiền lãnh = bậc lương * Ngày công + Phụ cấp. Xuất ra màn hình

Tham Khảo

Các thao tác Chuỗi trong C#

I / Giới thiệu :

- Chuỗi trong C# là một kiểu dựng sẵn như int , long ... có đầy đủ tính chất mềm dẻo , mạnh mẽ và dễ dùng

- Để khai báo một đối tượng chuỗi ta sử dụng từ khóa string

1 . Tạo chuỗi mới :

- Cú pháp : **string = " Khai báo và gán một chuỗi "**

2 . Phương thức ToString () :

- Dùng để chuyển đổi một đối tượng bất kỳ sang kiểu chuỗi

Ví dụ : int n = 5 ;

string s = n.ToString();

Bây giờ chuỗi s sẽ có giá trị là 5 . Bằng cách này ta cũng có thể tạo ra một chuỗi mới .

II / Các thao tác chuỗi :

1 . Các hàm thành viên :

- Empty : Biến thành viên tĩnh đại diện cho một chuỗi rỗng
- Compare() : Phương thức tĩnh dùng so sánh hai chuỗi
- Compare Ordinal () : So sánh hai chuỗi không quan tâm đến ngôn ngữ
- Concat () : Tạo chuỗi mới từ nhiều chuỗi
- Copy () : Tạo một bản sao
- Equals () : So sánh hai chuỗi có giống nhau
- Join () : Ghép nối nhiều chuỗi
- Chars () : Indexer của chuỗi
- Length () : Chiều dài chuỗi
- Insert() : Chèn một chuỗi khác vào chuỗi
- SubString () : Lấy một chuỗi con
- ToLower () : Tạo chuỗi chữ thường
- ToUpper () : Tạo chuỗi chữ hoa
- Trim () : Cắt bỏ khoảng trắng hai đầu chuỗi

VD1: Ghép chuỗi :

PHP Code:

```
string a = " Xin ";
string b = " Chào ";
string c = a + " " + b ; // Kết quả Xin Chào
```

VD 2 : Lấy ký tự

PHP Code:

```
string = " VIETPRO " ;
char c = s [ 1 ] ; // kết quả là V
```

VD3 : Lấy chuỗi con :

PHP Code:

```
string s = " VIETPRO ".Substring(4); // kết quả s = "VIET"
string s = " VIETPRO ".Substring(4,3) ; // kết quả s = "PRO"
```

VD 4: Thay thế chuỗi con

PHP Code:

```
string s = " vietpro ".Replace("v", "V") ; // Kết quả s = "Vietpro"
```

1. **Các cấu trúc điều khiển:** C# cung cấp hai cấu trúc điều khiển thực hiện việc lựa chọn điều kiện thực thi chương trình đó là cấu trúc if và switch...case

* **Cấu trúc if...else:** Được mô tả như sau:

```
if (biểu thức điều kiện)
{
// câu lệnh thực thi nếu biểu thức điều kiện đúng
}
[else
{
// câu lệnh thực thi nếu biểu thức điều kiện sai
}]
```

Ví dụ:

```
if (20 % 4 > 0)
{
Console.WriteLine("Số 20 không chia hết cho 4");
}
else
{
Console.WriteLine("Số 20 chia hết cho số 4");
}
```

Cấu trúc switch ... case

```
// switch ... case
switch (Biến điều kiện)
{
case giá trị 1: Câu lệnh thực thi
    break;
case giá trị 2: Câu lệnh thực thi
    break;
case giá trị 3: Câu lệnh thực thi
    break;
case giá trị n: Câu lệnh thực thi
default:
    Câu lệnh thực thi
    break;
}
```

Ví dụ:

```
int x = 20 % 4;
switch (x)
{
    case 1:
        Console.WriteLine("20 chia cho 4 được số dư là 1");
        break;
    case 0:
        Console.WriteLine("20 chia hết cho 4");
        break;
    default:
        Console.WriteLine("Không thuộc tất cả các trường hợp trên");
        break;
}
```

2. Cấu trúc vòng lặp trong lập trình C#

C# cung cấp các cấu trúc vòng lặp chương trình

- While
- Do... while
- For
- Foreach

Sau đây, tôi xin giới thiệu công thức và ví dụ sử dụng các vòng lặp trên

Vòng lặp *While*

Cấu trúc vòng lặp while

while (biểu thức điều kiện)

```
{
// câu lệnh
}
```

Thực thi câu lệnh hoặc một loạt những câu lệnh đến khi điều kiện không được thỏa mãn.

Ví dụ:

```
using System;
class WhileTest
{
public static void Main()
{
int n = 1;
while (n <>
{
Console.WriteLine("Current value of n is {0}", n);
n++;
}
}
}
```

Vòng lặp do

Cấu trúc vòng lặp while

do

{

// câu lệnh

}

While (biểu thức điều kiện)

Thực thi câu lệnh **ít nhất một lần** đến khi điều kiện không được thỏa mãn.

Ví dụ:

```
using System;
public class TestDoWhile
{
    public static void Main ()
    {
        int x;
        int y = 0;
        do
        {
            x = y++;
            Console.WriteLine(x);
        }
        while(y <> 0)
    }
}
```

Vòng lặp *for*

Cấu trúc vòng lặp *for*

```
for ([ phần khởi tạo] ; [biểu thức điều kiện]; [bước lặp])
{
    // thực thi câu lệnh
}
```

Ví dụ:

```
using System;
public class ForLoopTest
{
    public static void Main()
    {
        for (int i = 1; i <= 5; i++)
            Console.WriteLine(i);
    }
}
```


Vòng lặp *foreach*

Câu lệnh lặp *foreach* khá mới với những người đã học ngôn ngữ C, từ khóa này được sử dụng trong ngôn ngữ Visual Basic. Câu lệnh *foreach* cho phép chúng ta lặp qua tất cả các mục trong một mảng hay trong một tập hợp. Cú pháp sử dụng lệnh lặp *foreach* như sau:

```
foreach ( in )
{
// thực hiện thông qua tương ứng với
// từng mục trong mảng hay tập hợp
}
```

Dữ liệu kiểu tập hợp chưa được đề cập tới trong các bài học trước nên bạn chỉ cần quan tâm đến vòng lặp *foreach* sử dụng với mảng. Bạn hãy xem ví dụ sau để hiểu cách sử dụng của vòng lặp *foreach* truy cập đến từng phần tử của mảng.

```
using System;
public class UsingForeach
{
public static int Main()
{
int[ ] intArray = {1,2,3,4,5,6,7,8,9,10};
foreach( int item in intArray)
{
Console.WriteLine("{0} ", item);
}
Console.ReadLine();
return 0;
}
}
```

Kết quả:

0 1 2 3 4 5 6 7 8 9 10

3. Các lệnh *break*, *goto* và *continue*

Câu lệnh nhảy *goto*:

Lệnh nhảy *goto* là một lệnh nhảy đơn giản, cho phép chương trình nhảy vô điều kiện tới một vị trí trong chương trình thông qua tên nhãn. *Goto* giúp chương trình của bạn được linh hoạt hơn nhưng trong nhiều trường hợp nó sẽ làm mất đi cấu trúc thuật toán và gây rối chương trình.

Cách sử dụng lệnh goto:

Tạo một nhãn

goto đến nhãn

Nhãn là một định danh theo sau bởi dấu hai chấm (:). Thường thường một lệnh goto gắn với một điều kiện nào đó.

Ví dụ:

```
public class UsingGoto
{
    public static void Main()
    {
        int i = 0;
        lap: // nhãn
        Console.WriteLine("i:{0}",i);
        i++;
        if ( i <>
        goto lap; // nhảy về nhãn lap
        Console.ReadLine();
    }
}
```

Tương đương với vòng lặp for sau:

```
for (int i = 0; i <>
    Console.WriteLine("i:{0}", i);
```

Câu lệnh nhảy break và continue

Khi đang thực hiện các lệnh trong vòng lặp, có yêu cầu như sau: không thực hiện các lệnh còn lại nữa mà thoát khỏi vòng lặp, hay không thực hiện các công việc còn lại của vòng lặp hiện tại mà nhảy qua vòng lặp tiếp theo. Để đáp ứng yêu cầu trên C# cung cấp hai lệnh nhảy là break và continue để thoát khỏi vòng lặp.

Break khi được sử dụng sẽ đưa chương trình thoát khỏi vòng lặp và tiếp tục thực hiện các lệnh tiếp ngay sau vòng lặp.

Continue ngừng thực hiện các công việc còn lại của vòng lặp hiện thời và quay về đầu vòng lặp để thực hiện bước lặp tiếp theo.

Ví dụ:

```
public class UsingBreak_Continue
{
    public static void Main()
    {
```

```
for (int i = 0; i <>
{
if (i % 2 == 0) continue;
Console.WriteLine("i:{0}", i);
if (i==7) break;
}
Console.ReadLine();
}
}
```

Nếu không có lệnh break và continue vòng lặp sẽ lần lượt in ra các số từ 0 đến 9 nhưng khi gặp I chẵn ($i \% 2 == 0$) thì nó sẽ **continue** – tức là không thực hiện các lệnh tiếp theo mà quay trở lại đầu vòng lặp với giá trị của I được tăng lên 1. Lệnh **break** được thực hiện khi ($i == 7$) nó sẽ thoát khỏi vòng lặp ngay lập tức và cũng kết thúc chương trình và kết quả là chương trình trên chỉ in ra các số lẻ từ 1 đến 7

VÒNG LẶP TRONG C#

1. Vòng lặp do...while...

- Lặp tối thiểu 1 lần
- Số lần lặp không được biết trước
- Lưu ý: điều kiện lặp là phủ định của điều kiện thoát.
- Công thức:

```
do
{
    Lệnh nhập 1;
    Lệnh nhập 2;
    ...
    Lệnh nhập n;
}while (điều kiện lặp);
```

```
Int n;
Bắt đầu:
Console.Write("Nhập số n chẵn");
n = int.Parse(Console.ReadLine());
If(n % 2 != 0)
goto Bắt đầu;
```

```
Int n;
do
{Console.Write("Nhập số n chẵn");
n = int.Parse(Console.ReadLine());
}while(n % 2 != 0);
```

2. Vòng lặp While...

- Có khi không lặp lần nào cả.
- Không biết trước số lần lặp
- Công thức:

```
while (điều kiện lặp)
{
    Lệnh nhập 1;
    Lệnh nhập 2;
    ...
    Lệnh nhập n;
}
```

VD: Nhập vào n là số nguyên. Hãy vẽ ra n dòng tên của ta.

* Do...While...

```
Console.Write("Nhập số lần muốn viết tên");
Int n = int.Parse(Console.ReadLine());
do
{Console.Write("Tên của mình");
n--
}while(n > 0);
```

* While...

```
Console.Write("Nhập số lần muốn viết tên");
Int n = int.Parse(Console.ReadLine());
while(n > 0)
{
    Console.Write("Tên của mình");
    n--;
}
```

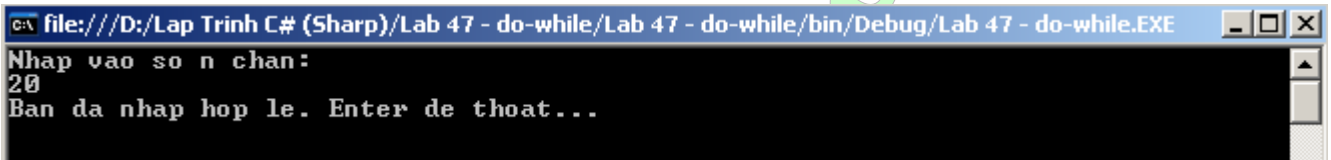
Lab 47 – do...while

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
```

```

namespace Bailab01
{
    class Program
    {
        static void Main(string[] args)
        {
            int n;
            do
            {
                Console.WriteLine("Nhap vao so n chan: ");
                n = int.Parse(Console.ReadLine());
            }
            while (n % 2 != 0);
            Console.WriteLine("Ban da nhap hop le. Enter de
thoat...");
            Console.ReadLine();
        }
    }
}

```

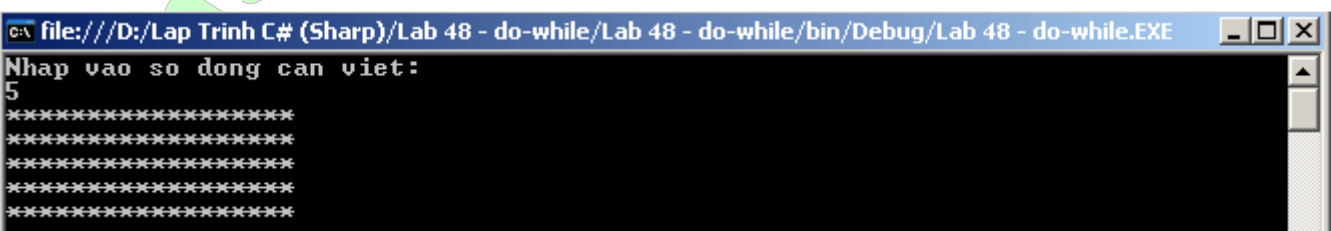

Lab 48 – do...while

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace BaiLab02
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Nhap vao so dong can viet: ");
            int n = int.Parse(Console.ReadLine());
            while (n > 0)
            {
                Console.WriteLine("*****");
                n--;
            }
            Console.ReadLine();
        }
    }
}

```



3. Vòng lặp for...

- Số lần lặp được biết trước
- Phải có biến đếm tham gia vào điều kiện lặp của vòng lặp
- Công thức:

```
for (khởi tạo giá trị biến đếm; điều kiện lặp; tăng/giảm biến đếm
{
  Lệnh lặp 1;
  Lệnh lặp 2;
  ...
  Lệnh lặp n;
}
```

Lab 49 – for

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace BaiLab03
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Liet ke tat ca so chan tu 0 - n.");
            Console.Write("Cho biet n: ");
            int n = int.Parse(Console.ReadLine());
            for (int i = 0; i < n; i+=2)
            {
                Console.Write("{0}\t", i);
            }
            Console.ReadLine();
        }
    }
}
```

```

file:///E:/THUC HANH LAP TRINH/2011.03.15/Bai_lab03/Bai_lab03/bin/Debug/BaiLab03.EXE
Liet ke tat ca so chan tu 0 - n.
Cho biet n: 1000
0      2      4      6      8      10     12     14     16     18
20     22     24     26     28     30     32     34     36     38
40     42     44     46     48     50     52     54     56     58
60     62     64     66     68     70     72     74     76     78
80     82     84     86     88     90     92     94     96     98
100    102    104    106    108    110    112    114    116    118
120    122    124    126    128    130    132    134    136    138
140    142    144    146    148    150    152    154    156    158
160    162    164    166    168    170    172    174    176    178
180    182    184    186    188    190    192    194    196    198
200    202    204    206    208    210    212    214    216    218
220    222    224    226    228    230    232    234    236    238
240    242    244    246    248    250    252    254    256    258
260    262    264    266    268    270    272    274    276    278
280    282    284    286    288    290    292    294    296    298
300    302    304    306    308    310    312    314    316    318
320    322    324    326    328    330    332    334    336    338
340    342    344    346    348    350    352    354    356    358
360    362    364    366    368    370    372    374    376    378
380    382    384    386    388    390    392    394    396    398
400    402    404    406    408    410    412    414    416    418
420    422    424    426    428    430    432    434    436    438
440    442    444    446    448    450    452    454    456    458
460    462    464    466    468    470    472    474    476    478
480    482    484    486    488    490    492    494    496    498
500    502    504    506    508    510    512    514    516    518
520    522    524    526    528    530    532    534    536    538
540    542    544    546    548    550    552    554    556    558
560    562    564    566    568    570    572    574    576    578
580    582    584    586    588    590    592    594    596    598
600    602    604    606    608    610    612    614    616    618
620    622    624    626    628    630    632    634    636    638
640    642    644    646    648    650    652    654    656    658
660    662    664    666    668    670    672    674    676    678
680    682    684    686    688    690    692    694    696    698
700    702    704    706    708    710    712    714    716    718
720    722    724    726    728    730    732    734    736    738
740    742    744    746    748    750    752    754    756    758
760    762    764    766    768    770    772    774    776    778
780    782    784    786    788    790    792    794    796    798
800    802    804    806    808    810    812    814    816    818
820    822    824    826    828    830    832    834    836    838
840    842    844    846    848    850    852    854    856    858
860    862    864    866    868    870    872    874    876    878
880    882    884    886    888    890    892    894    896    898
900    902    904    906    908    910    912    914    916    918
920    922    924    926    928    930    932    934    936    938
940    942    944    946    948    950    952    954    956    958
960    962    964    966    968    970    972    974    976    978
980    982    984    986    988    990    992    994    996    998
    
```

Cách khác

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace BaiLab03
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Liet ke tat ca so chan tu 0 - n.");
            Console.Write("Cho biet n: ");
            int n = int.Parse(Console.ReadLine());
            for (int i = 0; i <= n; i+=2)
            {
                Console.Write("{0}\t", i);
            }
            Console.ReadLine();
        }
    }
}
    
```

```

cs file:///E:/THUC HANH LAP TRINH/2011.03.15/Bai_lab03/Bai_lab03/bin/Debug/Bai_lab03.EXE
Liet ke tat ca so chan tu 0 - n.
Cho biet n: 1000
0      2      4      6      8      10     12     14     16     18
20     22     24     26     28     30     32     34     36     38
40     42     44     46     48     50     52     54     56     58
60     62     64     66     68     70     72     74     76     78
80     82     84     86     88     90     92     94     96     98
100    102    104    106    108    110    112    114    116    118
120    122    124    126    128    130    132    134    136    138
140    142    144    146    148    150    152    154    156    158
160    162    164    166    168    170    172    174    176    178
180    182    184    186    188    190    192    194    196    198
200    202    204    206    208    210    212    214    216    218
220    222    224    226    228    230    232    234    236    238
240    242    244    246    248    250    252    254    256    258
260    262    264    266    268    270    272    274    276    278
280    282    284    286    288    290    292    294    296    298
300    302    304    306    308    310    312    314    316    318
320    322    324    326    328    330    332    334    336    338
340    342    344    346    348    350    352    354    356    358
360    362    364    366    368    370    372    374    376    378
380    382    384    386    388    390    392    394    396    398
400    402    404    406    408    410    412    414    416    418
420    422    424    426    428    430    432    434    436    438
440    442    444    446    448    450    452    454    456    458
460    462    464    466    468    470    472    474    476    478
480    482    484    486    488    490    492    494    496    498
500    502    504    506    508    510    512    514    516    518
520    522    524    526    528    530    532    534    536    538
540    542    544    546    548    550    552    554    556    558
560    562    564    566    568    570    572    574    576    578
580    582    584    586    588    590    592    594    596    598
600    602    604    606    608    610    612    614    616    618
620    622    624    626    628    630    632    634    636    638
640    642    644    646    648    650    652    654    656    658
660    662    664    666    668    670    672    674    676    678
680    682    684    686    688    690    692    694    696    698
700    702    704    706    708    710    712    714    716    718
720    722    724    726    728    730    732    734    736    738
740    742    744    746    748    750    752    754    756    758
760    762    764    766    768    770    772    774    776    778
780    782    784    786    788    790    792    794    796    798
800    802    804    806    808    810    812    814    816    818
820    822    824    826    828    830    832    834    836    838
840    842    844    846    848    850    852    854    856    858
860    862    864    866    868    870    872    874    876    878
880    882    884    886    888    890    892    894    896    898
900    902    904    906    908    910    912    914    916    918
920    922    924    926    928    930    932    934    936    938
940    942    944    946    948    950    952    954    956    958
960    962    964    966    968    970    972    974    976    978
980    982    984    986    988    990    992    994    996    998
1000
    
```

Lab 50 – Bầu hồ lô

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace BaiLab04
{
    class Program
    {
        static void Main(string[] args)
        {
            double ToaDoThuc, ToaDoAo;
            double TamThuc, TamAo, TamThuc2, Goc;
            int BienDem;
            for (ToaDoAo = 1.2; ToaDoAo >= -1.2; ToaDoAo -= 0.05)
            {
                for (ToaDoThuc = -0.6; ToaDoThuc <= 1.77; ToaDoThuc +=
0.03)
                {
                    BienDem = 0;
                    TamThuc = ToaDoThuc;
                    TamAo = ToaDoAo;
                    Goc = (ToaDoThuc * ToaDoThuc) + (ToaDoAo * ToaDoAo);
                    while ((Goc < 4) && (BienDem < 40))
                    {
                        TamThuc = TamThuc * 0.95;
                        TamAo = TamAo * 0.95;
                        Goc = (TamThuc * TamThuc) + (TamAo * TamAo);
                        BienDem++;
                    }
                }
            }
        }
    }
}
    
```



```
ToaDoThuc;
TamThuc2 = (TamThuc * TamThuc) - (TamAo * TamAo) -
TamAo = (2 * TamThuc * TamAo) - ToaDoAo;
TamThuc = TamThuc2;
Goc = (TamThuc * TamThuc) + (TamAo * TamAo);
BienDem++;
}
switch (BienDem % 4)
{
    case 0:
        Console.WriteLine(".");
        break;
    case 1:
        Console.WriteLine("o");
        break;
    case 2:
        Console.WriteLine("O");
        break;
    case 3:
        Console.WriteLine("@");
        break;
}
}
Console.WriteLine("\n");
}
Console.ReadKey();
}
}
```



Lab 51 – Tổng những số chẵn

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
```

```
namespace Bailab05
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Tinh tong nhung so chan tu 0 - n");
            Console.Write("Nhap vao so n: ");
            int n = int.Parse(Console.ReadLine());
            int Sum = 0;
            for (int i = 0; i <= n; i += 2)
            {
                Console.Write("{0}\t", i);
                Sum += i;
            }
            Console.WriteLine();
        }
    }
}
```

```

        Console.WriteLine("Tong chan tu [0 - n] la: {0}", Sum);
        Console.ReadLine();
    }
}

```

```

file:///E:/THUC HANH LAP TRINH/2011.03.15/Bailab05/Bailab05/bin/Debug/Bailab05.EXE
Tinh tong nhung so chan tu 0 - n
Nhap vao so n: 18
0      2      4      6      8      10     12     14     16     18
Tong chan tu [0 - n] la: 90

```

Sangit.design.officelive.com

Lab 52 – Tam giác

Nhập vào 2 số hàng và cột. Hãy vẽ ra hình tam giác tương ứng bằng dấu *

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Tam_giac
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Nhap vao so hang Tam giac");
            int hang = int.Parse(Console.ReadLine());
            Console.WriteLine("Tam giac dac ruot");
            for (int i = 0; i < hang; i++)
            {
                for (int j = 0; j < hang - i - 1; j++)
                {
                    Console.Write("  ");
                }
                for (int j = 0; j < 2 * i + 1; j++)
                {
                    Console.Write("* ");
                }
                Console.WriteLine();
            }
            Console.WriteLine("Tam giac rong ruot");
            for (int i = 0; i < hang; i++)
            {
                for (int j = 0; j < hang - i - 1; j++)
                {
                    Console.Write("  ");
                }
                for (int j = 0; j < 2 * i + 1; j++)
                {
                    if (i == 0 || i == (hang - 1) ||

```

```

        j == 0 || j == 2 * i)
        Console.Write("* ");
        else Console.Write(" ");
    }
    Console.WriteLine();
}
}

Console.ReadLine();
}
}
}

```



Lab 53 – Hình chữ nhật

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Hinh_chu_nhat_rong
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.Write("Cho biet so hang: ");
            int hang = int.Parse(Console.ReadLine());
            Console.Write("Cho biet so cot: ");
            int cot = int.Parse(Console.ReadLine());
            Console.WriteLine("Hinh chu nhat dac ruot");
            for (int i = 0; i < hang; i++)
            {
                for (int j = 0; j < cot; j++)

```

```

    {
        Console.Write("* ");
    }
    Console.WriteLine();
}
Console.WriteLine("Hình chu nhật rong ruot");
for (int i = 0; i < hang; i++)
{
    for (int j = 0; j < cot; j++)
    {
        if (i == 0 || i == (hang - 1) ||
            j == 0 || j == (cot - 1))
            Console.Write("* ");
        else Console.Write(" ");
    }
    Console.WriteLine();
}
Console.ReadLine();
}
}
}

```

```

file:///E:/THUC HANH LAP TRINH/2011.03.25/Lab - Hinh chu nhac/Lab - Hinh chu nhac/bin/...
Cho biet so hang: 10
Cho biet so cot: 20
Hinh chu nhac dac ruot
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
Hinh chu nhac rong ruot
* * * * *
*
*
*
*
*
*
*
*
*
* * * * *

```

CẤU TRÚC ĐIỀU KIỆN CHƯƠNG TRÌNH

If...else (điều kiện):

```

if
{
    Các lệnh thực hiện khi đúng
}
else
{
    Các lệnh thực hiện khi điều kiện sai
}

```

Lab 16 – Mệnh đề if...else

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Lab16
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Nhập vào số a: ");
            int a = int.Parse(Console.ReadLine());
            Console.WriteLine("Nhập vào cơ số b: ");
            int b = int.Parse(Console.ReadLine());
            //string ketqua = a >= b ? "a >= b" : "a < b";
            string ketqua;
            if (a >= b)
            {
                ketqua = "a >= b";
            }
            else
            {
                ketqua = "a < b";
            }
            Console.WriteLine(ketqua);
            Console.ReadLine();
        }
    }
}

```

```

c:\ file:///D:/Lap Trinh C# (Sharp)/Lab16 - If else/Lab16 - If else/bin/Debug/Lab16 - If else.EXE
Nhập vào số a:
3
Nhập vào cơ số b:
8
a < b

```

Lab 17 – If...else

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

```

```

namespace BAILAB2
{
    class Program
    {
        static void Main(string[] args)

        {
            Console.WriteLine("Nhập vào số a: ");
            int a = int.Parse(Console.ReadLine());
            Console.WriteLine("Nhập vào số b: ");
            int b = int.Parse(Console.ReadLine());
            //string ketqua = a >= b ? "a > b" : a == b? "a == b": "a <
b";

            string ketqua;
            if (a > b)
            {
                ketqua = "a > b";
            }
            else
            {
                if (a == b)
                {
                    ketqua = "a == b";
                }
                else
                {
                    ketqua = "a < b";
                }
            }
            Console.WriteLine(ketqua);
            Console.ReadLine();
        }
    }
}

```

```

c:\ file:///D:/Lap Trinh C# (Sharp)/Lab17 - if else/Lab17 - if else/bin/Debug/Lab17 - if else.EXE
Nhập vào số a:
3
Nhập vào số b:
8
a < b

```

Lab 18 – Chẵn lẻ

Hãy nhập vào số a,

Nếu a chẵn thì cho $a = a/2$

Nếu a lẻ thì cho $a = a*2$

Viết công thức nhập số a và viết kết quả a sau khi xử lý.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

```

```

namespace Lab18
{
    class Program
    {
        static void Main(string[] args)

```

```

    {
        Console.WriteLine("Nhập vào số a: ");
        int a = int.Parse(Console.ReadLine());
        if (a % 2 == 0)
        {
            a /= 2; //a = a/2
        }
        else
        {
            a *= 2; //a = a*2
        }
        Console.WriteLine("a = {0}", a);
        Console.ReadLine();
    }
}

```

Lab 19

Nhập vào 1 số a, nếu số a nằm trong phạm vi từ 10 – 99 thì cho a = 2
 Ngược lại cho a = 0. Xuất giá trị a ra màn hình sau khi xử lý.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Lab19
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Nhập vào số a: ");
            int a = int.Parse(Console.ReadLine());
            if (a >= 10 && a <= 99)
            {
                a = 2;
            }
            else
            {
                a = 0;
            }
            Console.WriteLine("a = {0}", a);
            Console.ReadLine();
        }
    }
}

```


Lab 20

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Lab 20
{
    class Program
    {
        static void Main(string[] args)
        {
            int a = 3;           //0 0 0 0 0 0 1 1    21 + 20 = 2 + 1 = 3
            int b = 5;           //0 0 0 0 0 1 0 1    22 + 21 = 4 + 1 = 5
            int c = a & b;       //0 0 0 0 0 0 0 1
            int d = a | b;
            Console.WriteLine("c = {0} - d = {1}", c, d);
            Console.ReadLine();
        }
    }
}

```

**Lab 21**

Nhập vào 1 số a, nếu a là số âm thì a = -1, nếu a dương thì cho a = 1, nếu a = 0 thì giữ nguyên. Xuất a ra màn hình.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Lab21
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Nhap vao so a: ");
            int a = int.Parse(Console.ReadLine());
            if (a < 0)
            {
                a = -1;
            }
            else
            {
                if (a == 0)
                {
                    a = a;
                }
                else
                {
                    a = 1;
                }
            }
        }
    }
}

```

```

    }
    Console.WriteLine("a = {0}", a);
    Console.ReadLine();
}
}
}

```

Lab 22

Nhập vào 1 số a. Hãy kiểm tra nếu $a < 10$ thì cho $a = 10$, $a > 100$ thì cho $a = 100$. Xuất giá trị a.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Lab22
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Nhap vao so a: ");
            int a = int.Parse(Console.ReadLine());
            if (a < 10)
            {
                a = 10;
            }
            if (a > 100)
            {
                a = 100;
            }

            Console.WriteLine("a = {0}", a);
            Console.ReadLine();
        }
    }
}

```

Lab 23

Nhập vào 2 số a và b. Nếu $(a - b) < 0$ mà âm thì xuất ra thông báo " $b > a$ ", nếu $(a - b)$ là dương thì xuất thông báo " $a > b$ ", nếu $(a - b)$ là 0 thì xuất thông báo " $a = b$ "

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace BAILAB8
{
    class Program

```

```

{
    static void Main(string[] args)
    {
        Console.Write("Nhap vao so a: ");
        float a = float.Parse(Console.ReadLine());
        Console.Write("Nhap vao so b: ");
        float b = float.Parse(Console.ReadLine());
        string ketqua;
        if ((a - b) < 0)
        {
            ketqua = "b > a";
        }
        else
        {
            if ((a - b) == 0)
            {
                ketqua = "a = b";
            }
            else
            {
                ketqua = "a > b";
            }
        }
        Console.WriteLine("Vay {0}", ketqua);
        Console.ReadLine();
    }
}

```

```

c:\ file:///D:/Lap Trinh C# (Sharp)/Lab 23/Lab 23/bin/Debug/Lab 23.EXE
Nhap vao so a: 15
Nhap vao so b: 20
Vay b > a

```

Lab 24 – Trò chơi Oản tù xì

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

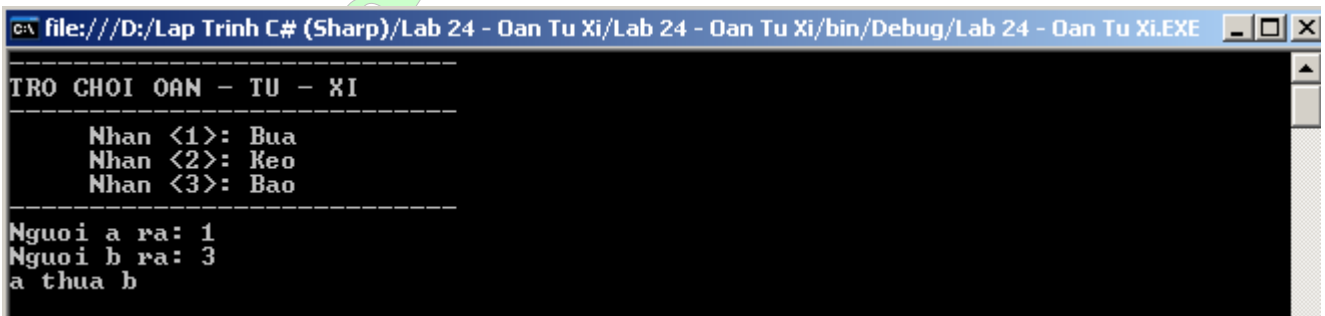
namespace bailab3
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("-----");
            Console.WriteLine("TRO CHOI OAN - TU - XI");
            Console.WriteLine("-----");
            Console.WriteLine("    Nhan <1>: Bua");
            Console.WriteLine("    Nhan <2>: Keo");
            Console.WriteLine("    Nhan <3>: Bao");
            Console.WriteLine("-----");
            Console.Write("Nguoi a ra: ");
            int a = int.Parse(Console.ReadLine());

```

```
Console.WriteLine("Nguoi b ra: ");
int b = int.Parse(Console.ReadLine());
string ketqua;

if (a == 1)
{
    if (b == 1)
    {
        ketqua = "a hoa b";
    }
    else
    {
        if (b == 2)
        {
            ketqua = "a thang b";
        }
        else
        {
            if (b == 3)
            {
                ketqua = "a thua b";
            }
            else
            {
                ketqua = "Nhap b khong hop le";
            }
        }
    }
}
else
{
    if (a == 2)
    {
        if (b == 1)
        {
            ketqua = "a thua b";
        }
        else
        {
            if (b == 2)
            {
                ketqua = "a hoa b";
            }
            else
            {
                if (b == 3)
                {
                    ketqua = "a thang b";
                }
                else
                {
                    ketqua = "Nhap b khong phu hop";
                }
            }
        }
    }
}
```

```
    }  
  }  
else  
{  
    if (a == 3)  
    {  
        if (b == 1)  
        {  
            ketqua = "a thang b";  
        }  
        else  
        {  
            if (b == 2)  
            {  
                ketqua = "a thua b";  
            }  
            else  
            {  
                if (b == 3)  
                {  
                    ketqua = "a hoa b";  
                }  
                else  
                {  
                    ketqua = "Nhap b khong phu hop";  
                }  
            }  
        }  
    }  
    ketqua = "Nhap a khong phu hop";  
}  
Console.WriteLine(ketqua);  
Console.ReadLine();  
}  
}
```



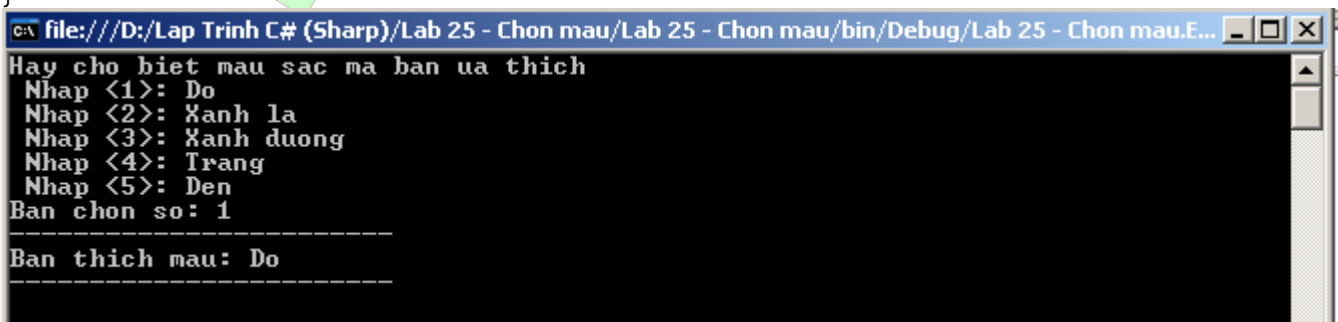
```
C:\ file:///D:/Lap Trinh C# (Sharp)/Lab 24 - Oan Tu Xi/Lab 24 - Oan Tu Xi/bin/Debug/Lab 24 - Oan Tu Xi.EXE  
-----  
TRO CHOI OAN - TU - XI  
-----  
Nhap <1>: Bua  
Nhap <2>: Keo  
Nhap <3>: Bao  
-----  
Nguoi a ra: 1  
Nguoi b ra: 3  
a thua b
```

Hàm switch...case

Lab 25 – Chọn màu

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace BaiLab1
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hay cho biet mau sac ma ban ua thich");
            Console.WriteLine(" Nhap <1>: Do");
            Console.WriteLine(" Nhap <2>: Xanh la");
            Console.WriteLine(" Nhap <3>: Xanh duong");
            Console.WriteLine(" Nhap <4>: Trang");
            Console.WriteLine(" Nhap <5>: Den");
            Console.Write("Ban chon so: ");
            int i = int.Parse(Console.ReadLine());
            string kq;
            switch (i)
            {
                case 1: kq = "Do";
                    break;
                case 2: kq = "Xanh la";
                    break;
                case 3: kq = "Xanh duong";
                    break;
                case 4: kq = "Trang";
                    break;
                case 5: kq = "Den";
                    break;
                default:
                    kq = "Khong ro";
                    break;
            }
            Console.WriteLine("Ban thich mau: {0}", kq);
            Console.ReadLine();
        }
    }
}
```



```
file:///D:/Lap Trinh C# (Sharp)/Lab 25 - Chon mau/Lab 25 - Chon mau/bin/Debug/Lab 25 - Chon mau.E...
Hay cho biet mau sac ma ban ua thich
Nhap <1>: Do
Nhap <2>: Xanh la
Nhap <3>: Xanh duong
Nhap <4>: Trang
Nhap <5>: Den
Ban chon so: 1
-----
Ban thich mau: Do
-----
```

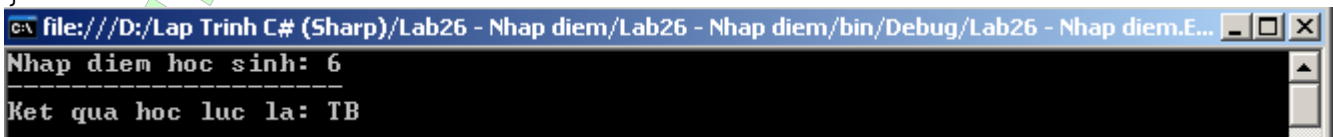
Lab 26 – Nhập điểm

Nhập vào điểm nguyên của học sinh:

Nếu 9, 10 : Giỏi
 Nếu 8, 7 : Khá
 Nếu 6, 5 : T.Bình
 Nếu 4, 3 : Yếu
 Nếu 2, 1, 0 : Kém

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace BaiLab2
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.Write("Nhap diem hoc sinh: ");
            int diem = int.Parse(Console.ReadLine());
            string kq;
            switch (diem)
            {
                case 10:
                case 9: kq = "Gioi";
                    break;
                case 8:
                case 7: kq = "Kha";
                    break;
                case 6:
                case 5: kq = "TB";
                    break;
                case 4:
                case 3: kq = "Yeu";
                    break;
                case 2:
                case 1:
                case 0: kq = "Kem";
                    break;
                default:
                    kq = "Nhap diem khong hop le";
                    break;
            }
            Console.WriteLine("Ket qua hoc luc la: {0}", kq);
            Console.ReadLine();
        }
    }
}
```



```
file:///D:/Lap Trinh C# (Sharp)/Lab26 - Nhap diem/Lab26 - Nhap diem/bin/Debug/Lab26 - Nhap diem.E...
Nhap diem hoc sinh: 6
Ket qua hoc luc la: TB
```

Lab 27 – Nhập số trong khoảng

Nhập vào 1 số a trong khoảng [10 - 20]

Nếu người dùng nhập sai, bắt người dùng nhập lại.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace BaiLab4
{
    class Program
    {
        static void Main(string[] args)
        {
            int a;
            start:
            Console.Write("Nhap vao so a [10 - 20]");
            a = int.Parse(Console.ReadLine());

            if (a < 10 || a > 20)
            {
                Console.WriteLine("Ban phai nhap a nam trong khoang
[10 - 20]. Nhap lai.");
                goto start;
            }
            else
            {
                Console.WriteLine("Ban da nhap hop le");
            }
            Console.ReadLine();
        }
    }
}
```

```
file:///D:/Lap Trinh C# (Sharp)/Lab27 - Nhap so trong khoang/Lab27 - Nhap so trong khoang/bin/Deb...
Nhap vao so a [10 - 20]: 9
Ban phai nhap a nam trong khoang [10 - 20]: . Nhap lai.
Nhap vao so a [10 - 20]: 15
Ban da nhap hop le
```

Lab 28 – Nhập số chẵn

Nhập vào số a chẵn, nếu không đúng thì bắt người dùng nhập lại

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace baitap__nhap_so_chan_
{
    class Program
    {
        static void Main(string[] args)
        {
            int a;
            start:
            Console.WriteLine("nhap vao 1 so chan :");
```



```

        a = int.Parse(Console.ReadLine());
        if ( a % 2 != 0)
        {
            Console.WriteLine("ban hay nhap vao so chan.nhap
lai");
            goto start;
        }

        else
        {
            Console.WriteLine("ban da nhap so hop le");
        }

        Console.ReadLine();
    }
}
}

```

```

c:\ file:///D:/Lap Trinh C# (Sharp)/Lab 28 - Nhap so chan/Lab 28 - Nhap so chan/bin/Debug/Lab 28 - Nhap...
nhap vao 1 so chan:
15
Khong phai la so chan, ban hay nhap vao so chan. Nhap lai:
nhap vao 1 so chan:
10
Ban da nhap so hop le

```

Lab 29 – Nhập số, Học lực

Nhập số a là điểm của học sinh, nếu nhập ngoài khoảng [0 – 10] thì bắt người dùng nhập lại. Sau đó dùng switch...case để thông báo học lực của học sinh.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Lab29
{
    class Program
    {
        static void Main(string[] args)
        {
            int a;
            string kq = "";
            start:
            Console.WriteLine("hay nhap diem cua hoc sinh :");
            a = int.Parse(Console.ReadLine());
            if (a < 0 || a > 10)
            {
                Console.WriteLine("ban nhap diem chua dung.nhap lai");
                goto start;
            }
            else
            {
                switch (a)
                {
                    case 0:

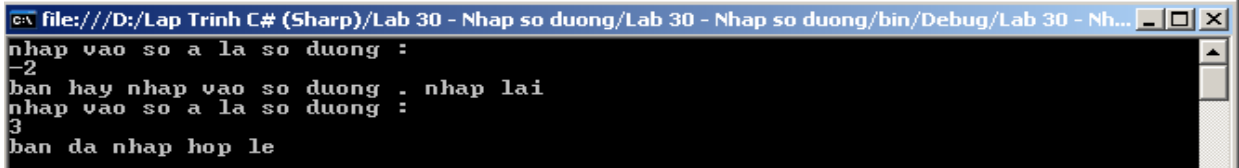
```



```

    }
    else Console.WriteLine("ban da nhap hop le");
    Console.ReadLine();
}
}
}

```



```

c:\ file:///D:/Lap Trinh C# (Sharp)/Lab 30 - Nhap so duong/Lab 30 - Nhap so duong/bin/Debug/Lab 30 - Nh...
nhap vao so a la so duong :
-2
ban hay nhap vao so duong : nhap lai
nhap vao so a la so duong :
3
ban da nhap hop le

```

Kiểm tra 10 phút

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

```

```

namespace BAIKIEMTRA1

```

```

{
    class Program
    {
        static void Main(string[] args)
        {
            int a;
            start:
            do
            {
                Console.WriteLine("Nhap vào so a: ");
                a = int.Parse(Console.ReadLine());
            }
            while (a % 2 != 1);
            Console.WriteLine("Ban nhap so hop le");
            goto start;
            Console.ReadLine();
        }
    }
}

```



```

c:\ file:///D:/Lap Trinh C# (Sharp)/KiemTra10p/KiemTra10p/bin/Debug/KiemTra10p.EXE
Nhap vào so a:
2
Nhap vào so a:
3
Ban nhap so hop le
Nhap vào so a:

```

Các hàm, thủ tục trong C#

1. Hàm:

- Giống như thủ tục nhưng có giá trị trả về. Do đó, "Void" trong thủ tục phải được thay bằng 1 kiểu nào đó gọi là kiểu của biến trả về. Trong công thức của hàm, phải có dòng "Return" là trả giá trị về của hàm.

2. Tham số và các cách truyền tham số trong hàm và thủ tục. Có 3 dạng truyền:

2.1. Truyền bình thường (tham trị):

Không làm thay đổi giá trị tham số sau khi ra khỏi thủ tục và hàm. Mặc dù trong thủ tục/Hàm đã làm thay đổi giá trị tham số đó.

2.2. Truyền bằng tham chiếu (Reference) (lab31)

Các tham số truyền bằng tham chiếu sẽ thay đổi khi ra ngoài thủ tục hàm nếu trong thủ tục/Hàm chúng ta thay đổi giá trị của tham số đó.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Lab2
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Nhập vào số a: ");
            int a = int.Parse(Console.ReadLine());
            Console.WriteLine("Nhập vào số b: ");
            int b = int.Parse(Console.ReadLine());
            HoanVi(ref a, ref b);
            Console.WriteLine("a = {0} - b = {1}", a, b);
            Console.ReadLine();
        }
        static void HoanVi(ref int a, ref int b)
        {
            int temp = a;
            a = b;
            b = temp;
        }
    }
}
```

```
C:\ file:///D:/Lap Trinh C# (Sharp)/Lab 31 - Truyen bang tham so/Lab 31 - Truyen bang tham so/bin/Deb...
Nhập vào số a:
12
Nhập vào số b:
20
a = 20 - b = 12
```

2.3. Truyền bằng từ khóa "out (lab 32)

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Lab2
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Nhập vào số a: ");
            int a = int.Parse(Console.ReadLine());
            Console.WriteLine("Nhập vào số b: ");
            int b = int.Parse(Console.ReadLine());
            int c;
            Tong(a, b, out c);
            Console.WriteLine("a = {0} - b = {1}", c);
            Console.ReadLine();
        }
        static void HoanVi(ref int a, ref int b)
        {
            int temp = a;
            a = b;
            b = temp;
        }

        static void Tong(int a, int b, out int c)
        {
            c = a + b;
        }
    }
}
```

Lab 33 – Tổng 3 chữ số

Nhập vào 1 số có 3 chữ số. Hãy tính tổng các chữ số của số đó. Ví dụ: 123 cho Tổng 6

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace BaiTapvenha
{
    class Program
    {
        static void Main(string[] args)
        {
            int i;
            batdau:
            Console.Write("Nhập số có 3 chữ số:");
            i = int.Parse(Console.ReadLine());
            if (i < 100 || i > 999)
                goto batdau;
        }
    }
}
```

```

    {
        goto batdau;
    }
else
{
    int a = i;
    int donvi = a % 10;
    a /= 10;
    int hangchuc = a % 10;
    int hangdonvi = a / 10;

    Console.WriteLine("Tong 3 chu so:{0} = {1}", i, donvi
+ hangchuc + hangdonvi);
    Console.ReadLine();
}
}
}
}
}

```

```

c:\ file:///F:/bai tap ve ha/BaiTapvenha/BaiTapvenha/bin/Debug/BaiTapvenha.EXE
Nhap so co 3 chu so:256
Tong 3 chu so:256 = 13

```

Cách khác Lab 33

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Lab4
{
    class Program
    {
        static void Main(string[] args)
        {
            int a = ThuVien.NhapsoInt("Nhap vao so co 3 chu so: ",
999, 100);
            int i = a;
            int DonVi = i % 10;
            i /= 10;
            int HangChuc = i % 10;
            int HangTram = i / 10;
            Console.WriteLine("Tong 3 chu so cua {0} = {1}", a,
HangTram + HangChuc + DonVi);
            Console.ReadLine();
        }
    }
}

```

Với:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

```

```

static class ThuVien
{
    public static int nhapsoint(string thongbao)
    {
        Console.Write(thongbao);
        return int.Parse(Console.ReadLine());
    }
    public static int nhapsoint(string thongbao,int max , int min)
    {
        int a;
        do
        {
            Console.Write(thongbao);
            a = int.Parse(Console.ReadLine());
        } while (a < min || a > max);
        return a;
    }
}
}

```

Lab 34 – Tổng thành hàm có tổng 12 (100 – 999)

Viết phần tính tổng thành hàm. Sau đó, liệt kê tất cả những số từ 100 – 999 có tổng là 12

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace BaiTapVeNha_1_
{
    class Program
    {
        static void Main(string[] args)
        {
            int dem = 0;
            int tong;
            for (int j = 100; j <= 999; j++)
            {
                int n = j;
                int donvi = n % 10;
                n /= 10;
                int hangchuc = n % 10;
                int hangdonvi = n / 10;
                tong = donvi + hangchuc + hangdonvi;
                if (tong == 12 )
                {
                    Console.WriteLine("{0}\t", j);
                    j++;
                    dem++;
                }
            }
        }
    }
}

```

```

    }
    else
    {
        Console.WriteLine("");
    }
}
Console.WriteLine("Số 3 chữ số có tổng 12: {0} số", dem);
Console.ReadLine();
}
}
}

```

```

c:\ file:///F:/bai tap ve ha/BaiTapVeNha(1)/bin/Debug/BaiTapVeNha(1).EXE
570
606
615
624
633
642
651
660
705
714
723
732
741
750
804
813
822
831
840
903
912
921
930
Số 3 chữ số có tổng 12: 66 số

```

Lab 35 – Tổng các chữ số chia hết cho 7

Vẫn sử dụng hàm tính tổng các chữ số của 1 số. Liệt kê tất cả những số mà có ký tự cuối cùng là 9. Nhưng tổng các chữ số chia hết cho 7

```

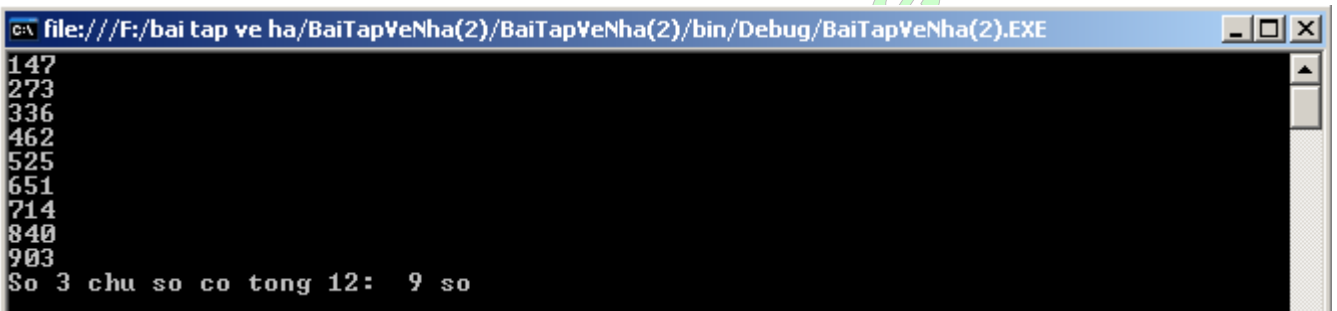
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int dem = 0;
            int tong;
            for (int j = 100; j <= 999; j++)
            {
                int n = j;
                int donvi = n % 10;
                n /= 10;
                int hangchuc = n % 10;
                int hangdonvi = n / 10;
            }
        }
    }
}

```



```
tong = donvi + hangchuc + hangdonvi;  
if (tong == 12 && j % 7 == 0)  
{  
    Console.WriteLine("{0}\t", j);  
    j++;  
    dem++;  
}  
else  
{  
    Console.Write("");  
}  
}  
Console.WriteLine("So 3 chu so co tong 12: {0}so", dem);  
Console.ReadLine();  
}  
}
```



```
C:\ file:///F:/bai tap ve ha/BaiTapVeNha(2)/BaiTapVeNha(2)/bin/Debug/BaiTapVeNha(2).EXE  
147  
273  
336  
462  
525  
651  
714  
840  
903  
So 3 chu so co tong 12: 9 so
```

MẢNG

- Bao gồm những biến cùng kiểu nằm liên tiếp nhau trong bộ nhớ.
- Các thành phần của mảng bao gồm:
 - Chỉ số mảng (bắt đầu từ 0 đến $n - 1$ nếu mảng có n phần tử)
 - Giá trị của mảng là các giá trị ngẫu nhiên chứa trong mỗi ô nhớ của các phần tử trong mảng
- Khai báo

```
int[] a;      a: tên mảng
```

```
int[] b = {3, 6, 9};
```

0	1	2
3	6	9

```
int[] c = new int[ 5 ];
```

0	1	2	3	4
0	0	0	0	0

- Truy cập giá trị

```
Console.WriteLine(b[2]);    → 9
```

```
C[4] = 10
```

Chỉ số	0	1	2	3	4
Giá trị số	0	0	0	0	10

```
int i = 6;
```

```
Console.WriteLine(i + 1);
```

Nhập vào 1 mảng từ bàn phím: (i = i + 1)

Lab 36 – Nhập 1 mảng từ bàn phím

Nhập 1 mảng từ bàn phím

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Lab1
{
    class Program
    {
        static void Main(string[] args)
        {
            //Khai báo mảng a có phần tử
            int[] a = new int[5];

            //Nhập mảng a từ bàn phím
            for (int i = 0; i < a.Length; i++)
            {
                Console.Write("Nhập phần tử a[{0}]: ", i);
                a[i] = int.Parse(Console.ReadLine());
            }
        }
    }
}
```

```

    }

    //Xuat cac gia tri co trong mang a ra man hinh
    for (int i = 0; i < a.Length; i++)
    {

        Console.Write("{0}\t", a[i]);

    }
    Console.ReadLine();
}
}
}

```

```

file:///E:/THUC HANH LAP TRINH/2011.03.22/Lab1/Lab1/bin/Debug/Lab1.EXE
Nhap phan tu a[0]: 1
Nhap phan tu a[1]: 5
Nhap phan tu a[2]: 8
Nhap phan tu a[3]: 9
Nhap phan tu a[4]: 6
1 5 8 9 6

Sangit.design.officelive.com

```

Lab 37 – Khai báo mảng

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Lab2
{
    class Program
    {
        static void Main(string[] args)
        {
            //Khai bao mang a co phan tu
            int[] a = new int[5];

            //Nhap mang a tu ban phim
            for (int i = 0; i < a.Length; i++)
            {
                Console.Write("Nhap phan tu a[{0}]: ", i);
                a[i] = int.Parse(Console.ReadLine());
            }

            //Xuat cac gia tri co trong mang a ra man hinh
            for (int i = 0; i < a.Length; i++)
            {
                Console.Write("{0}\t", a[i]);
            }
        }
    }
}

```

```

        Console.ReadLine();
    }
    static int[] NhapMangInt()
    {
        Console.Write("Cho biet so phan tu Mang: ");
        int n = int.Parse(Console.ReadLine());
        int[] a = new int[n];
        //Nhap mang a tu ban phim. a.Length: la so phan tu co
        trong mang
        for (int i = 0; i < a.Length; i++)
        {
            Console.Write("Nhap phan tu a[{0}]: ", i);
        }
        return a;
    }

    static void XuatMangInt(int[] a)
    {
        //Xuat cac gia tri co trong mang a ra man hinh
        for (int i = 0; i < a.Length; i++)
        {
            Console.Write("{0}\t", a[i]);
        }
    }
}

```

```

C:\ file:///E:/THUC HANH LAP TRINH/2011.03.22/Lab2/Lab2/bin/Debug/Lab2.EXE
Nhap phan tu a[0]: 2
Nhap phan tu a[1]: 5
Nhap phan tu a[2]: 6
Nhap phan tu a[3]: 8
Nhap phan tu a[4]: 5
2 5 6 8 5

```

Lab 38 – Nhập mảng Random

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Lab2
{
    class Program
    {
        static void Main(string[] args)
        {
            int[] a = NhapMangIntRandom();
            XuatMangInt(a);
            Console.ReadLine();
        }
        static int[] NhapMangInt()
        {
            Console.Write("Cho biet so phan tu Mang: ");

```

```

int n = int.Parse(Console.ReadLine());
int[] a = new int[n];
//Nhap mang a tu ban phim. a.Length: la so phan tu co
trong mang
for (int i = 0; i < a.Length; i++)
{
    Console.Write("Nhap phan tu a[{0}]: ", i);
}
return a;
}

static int[] NhapMangIntRandom()
{
    Console.Write("Cho biet so phan tu Mang: ");
    int n = int.Parse(Console.ReadLine());

    int[] a = new int[n];
    //Nhap mang a tu ban phim. a.Length: la so phan tu co
trong mang
    Random rand = new Random(); //Khoi tao bo tao so ngau
nhien
    for (int i = 0; i < a.Length; i++)
    {
        a[i] = rand.Next(100); //Lay mot so ngau nhien < 100
    }
    return a;
}

static void XuatMangInt(int[] a)
{
    //Xuat cac gia tri co trong mang a ra man hinh
    for (int i = 0; i < a.Length; i++)
    {
        Console.Write("{0}\t", a[i]);
    }
}
}
}
}

```

```

file:///E:/THUC HANH LAP TRINH/2011.03.22/Lab2/Lab2/bin/Debug/Lab2.EXE
Cho biet so phan tu Mang: 100
47      66      62      88      18      13      35      11      7      87
61      5       42      20      28      29      93      97      55      28
27      9       57      58      45      7       45      99      49      92
49      37      3       84      65      78      5       89      42      48
78      92      20      92      18      59      9       58      1       55
92      80      19      55      95      37      8       4       42      10
67      35      61      15      37      24      2       58      55      49
23      4       54      6       49      34      89      65      40      86
97      87      98      94      99      69      17      47      88      27
70      1       46      31      81      42      31      4       54      93

```

Lab 39 – Sắp xếp

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Lab3

```

```

{
class Program
{
    static void Main(string[] args)
    {
        int[] a = NhapMangIntRandom();
        Array.Sort(a); //Thu tu sap xep mang a tu nho den lon
        XuatMangInt(a);
        Console.ReadLine();
    }
    static int[] NhapMangInt()
    {
        Console.Write("Cho biet so phan tu Mang: ");
        int n = int.Parse(Console.ReadLine());

        int[] a = new int[n];
        //Nhap mang a tu ban phim. a.Length: la so phan tu co
        trong mang
        for (int i = 0; i < a.Length; i++)
        {
            Console.Write("Nhap phan tu a[{0}]: ", i);
        }
        return a;
    }
    static int[] NhapMangIntRandom()
    {
        Console.Write("Cho biet so phan tu Mang: ");
        int n = int.Parse(Console.ReadLine());

        int[] a = new int[n];
        //Nhap mang a tu ban phim. a.Length: la so phan tu co
        trong mang
        Random rand = new Random(); //Khoi tao bo tao so ngau
        nhien
        for (int i = 0; i < a.Length; i++)
        {
            a[i] = rand.Next(100); //Lay mot so ngau nhien < 100
        }
        return a;
    }
    static void XuatMangInt(int[] a)
    {
        //Xuat cac gia tri co trong mang a ra man hinh
        for (int i = 0; i < a.Length; i++)
        {
            Console.Write("{0}\t", a[i]);
        }
    }
}
}

```

```

file:///E:/THUC HANH LAP TRINH/2011.03.22/Lab3/Lab3/bin/Debug/Lab3.EXE
Cho biet so phan tu Mang: 30
4      9      15     18     21     22     22     28     28     39
43     45     51     53     54     59     60     62     64     65
76     77     78     79     84     86     87     89     89     95

```

Lab 40 – Sắp xếp mảng

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Lab3
{
    class Program
    {
        static void Main(string[] args)
        {
            int[] a = NhapMangIntRandom();
            Array.Sort(a); //Thu tu sap xep mang a tu nho den lon
            Array.Reverse(a);
            XuatMangInt(a);
            Console.ReadLine();
        }

        static int[] NhapMangInt()
        {
            Console.Write("Cho biet so phan tu Mang: ");
            int n = int.Parse(Console.ReadLine());
            int[] a = new int[n];
            //Nhap mang a tu ban phim. a.Length: la so phan tu co
            trong mang
            for (int i = 0; i < a.Length; i++)
            {
                Console.Write("Nhap phan tu a[{0}]: ", i);
            }
            return a;
        }

        static int[] NhapMangIntRandom()
        {
            Console.Write("Cho biet so phan tu Mang: ");
            int n = int.Parse(Console.ReadLine());
            int[] a = new int[n];
            //Nhap mang a tu ban phim. a.Length: la so phan tu co
            trong mang
            Random rand = new Random(); //Khoi tao bo tao so ngau
            nhien
            for (int i = 0; i < a.Length; i++)
            {
                a[i] = rand.Next(100); //Lay mot so ngau nhien < 100
            }
            return a;
        }

        static void XuatMangInt(int[] a)
        {
            //Xuat cac gia tri co trong mang a ra man hinh
            for (int i = 0; i < a.Length; i++)

```

```

    {
        Console.WriteLine("{0}\t", a[i]);
    }
}
}
}

```

```

file:///E:/THUC HANH LAP TRINH/2011.03.22/Lab3/Lab3/bin/Debug/Lab3.EXE
Cho biet so phan tu Mang: 30
96    94    91    90    90    82    80    78    77    71
64    61    61    54    53    53    47    45    45    44
43    41    31    30    23    13    11    7     5     0

```

Lab 41 – Mảng 2 Chiều

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

```

```

namespace Lab4

```

```

{
    class Program
    {
        static void Main(string[] args)
        {
            int[,] a = NhapMang2DInt();
            XuatMang2DInt(a);
            Console.ReadLine();
        }
        static int[,] NhapMang2DInt()
        {
            Console.WriteLine("Cho biet so hang: ");
            int hang = int.Parse(Console.ReadLine());
            Console.WriteLine("Cho biet so cot: ");
            int cot = int.Parse(Console.ReadLine());
            int[,] a = new int[hang, cot];

            //a.GetLength(0) - lay ra so hang co trong mang a
            for (int i = 0; i < a.GetLength(0); i++)
            {
                //a.GetLength(1) - lay ra so cot co trong mang a
                for (int j = 0; j < a.GetLength(1); j++)
                {
                    Console.WriteLine("Nhap phan tu [{0}:{1}]: ", i, j);
                    a[i, j] = int.Parse(Console.ReadLine());
                }
            }
            return a;
        }
        static void XuatMang2DInt(int[,] a)
        {
            for (int i = 0; i < a.GetLength(0); i++)
            {
                for (int j = 0; j < a.GetLength(1); j++)
                {
                    Console.WriteLine("{0}\t", a[i, j]);
                }
            }
        }
    }
}

```



```

C:\ file:///D:/Lap Trinh C# (Sharp)/Lab4/Lab4/bin/Debug/Lab4.EXE
Ban muon nhap bao nhieu phan tu:
10
Moi nhap phan tu thu: 1
5
Moi nhap phan tu thu: 2
8
Moi nhap phan tu thu: 3
9
Moi nhap phan tu thu: 4
4
Moi nhap phan tu thu: 5
8
Moi nhap phan tu thu: 6
1
Moi nhap phan tu thu: 7
3
Moi nhap phan tu thu: 8
5
Moi nhap phan tu thu: 9
1
Moi nhap phan tu thu: 10
15
So luong so hoan hao la: 0
Sap xep lai phan tu tu nho den lon!
1
1
3
4
5
5
8
8
9
15
Sap xep mang tu lon den nho!
15
9
8
8
5
5
4
3
1
1
1
Tong cac phan tu duong la: 59
Phan tu lon nhat la: 15
Phan tu nho nhat la: 5

```

Lab 43 – Tham khảo

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using x = System.Console;

namespace mang1chieu2
{
    class Program
    {
        //Viết chương trình tính tổng bình phương cả các số âm trong
        một mảng các số nguyên?
        static void Main(string[] args)
        {
            int i, n;
            x.WriteLine("\n Nhap vao:n=");

```

```

n = Convert.ToInt32(x.ReadLine());
int[] a = new int[n];
x.WriteLine("\n Nhap cac phan tu cho mang");
for (i = 0; i < a.GetLength(0); i++)
{
    x.Write("a[{0}]=", i);
    a[i] = Convert.ToInt32(x.ReadLine());
}
//in mang
for (i = 0; i < a.GetLength(0); i++)
{
    x.Write(" {0}", a[i]);
}
//tong mang
x.Write("Tong cac phan tu am cua mang:S={0}", tong(a, n));
x.ReadLine();
}
static int tong(int[] a, int n)
{
    int s = 0, i;
    for (i = 0; i < n; i++)
        if (a[i] < 0)
            s += a[i];
    return s;
}
}
}

```

```

c:\ file:///D:/Lap Trinh C# (Sharp)/Lab5-Mang/Lab5-Mang/bin/Debug/Lab5-Mang.EXE
Nhap vao:n=
10
Nhap cac phan tu cho mang
a[0]=2
a[1]=8
a[2]=1
a[3]=-2
a[4]=8
a[5]=3
a[6]=4
a[7]=2
a[8]=-5
a[9]=-9
2 8 1 -2 8 3 4 2 -5 -9Tong cac phan tu am cua mang:S=-16_

```

Lab 44 – Tham khảo

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using x = System.Console;

namespace mang1chieu2
{
    class Program

```

```

{
    //Viết chương trình thực hiện việc đảo mảng một chiều.
    static void Main(string[] args)
    {
        int i, n;
        x.Write("\n Nhập vào:n=");
        n = Convert.ToInt32(x.ReadLine());
        int[] a = new int[n];
        x.WriteLine("Nhập các phần tử cho mảng");
        for (i = 0; i < a.GetLength(0); i++)
        {
            x.Write("a[{0}]=", i);
            a[i] = Convert.ToInt32(x.ReadLine());
        }
        //in mảng
        for (i = 0; i < a.GetLength(0); i++)
        {
            x.Write(" {0}", a[i]);
        }
        //đảo ngược mảng
        x.WriteLine("\nmảng sau khi đảo ngược:");
        daomang(a, n);
        for (i = 0; i < a.GetLength(0); i++)
        {
            x.Write(" {0}", a[i]);
        }
        x.ReadLine();
    }
    static void daomang(int[] a, int n)
    {
        int i, tg;
        for (i = 0; i < n / 2; i++)
        {
            tg = a[i];
            a[i] = a[n - i - 1];
            a[n - i - 1] = tg;
        }
    }
}
}

```

C:\ file:///D:/Lap Trinh C# (Sharp)/Lab6-Mang/Lab6-Mang/bin/Debug/Lab6-Mang.EXE

```

Nhập vào:n=10
Nhập các phần tử cho mảng
a[0]=2
a[1]=5
a[2]=6
a[3]=47
a[4]=7
a[5]=8
a[6]=5
a[7]=2
a[8]=1
a[9]=2
 2 5 6 47 7 8 5 2 1 2
mảng sau khi đảo ngược:
 2 1 2 5 8 7 47 6 5 2_

```

Lab 45 – Tổng các giá trị ở vị trí chẵn

Viết công thức nhập vào 1 mảng a bằng phương pháp tạo ngẫu nhiên. Hãy xuất ra các phần tử trong mảng a và tính tổng số các giá trị ở vị trí chẵn trong mảng a.

Ví dụ

0	1	2	3	4	5
8	10	12	4	7	9

Tổng các giá trị ở vị trí chẵn: 27

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            // Bài Tập 1 nhưng tổng các số đếm từ trị số mảng lẻ
            Console.WriteLine("Cho biết số phần tử mảng: ");
            int n = int.Parse(Console.ReadLine());
            int[] a = new int[n];
            Random rand = new Random();//khởi tạo bộ tạo số ngẫu nhiên
            for (int i = 0; i < a.Length; i++)
            {
                a[i] = rand.Next(100);//xuất ra một giá trị ngẫu nhiên < 100
            }
            xuấtmangint(a);
            int sum = 0;
            for (int i = 1; i < n; i++)
            {
                i++;
                sum = sum + a[i - 1];
            }
            Console.WriteLine();
            Console.WriteLine("Tong cac so chan: {0}", sum);

            Console.ReadLine();
        }
        static void xuấtmangint(int[] a)
        {
            //Xuất các phần tử mảng;
            for (int i = 0; i < a.Length; i++)
            {
                Console.Write("{0}\t", a[i]);
            }
        }
    }
}
```

```
file:///F:/2011.03.22 on pc17/BaiTapVeNha1-1/BaiTapVeNha1-1/bin/Debug/BaiTapVeNha1-1.EXE
Cho biet so phan tu mang: 10
99 71 61 67 9 0 20 52 99 3
Tong cac so chan: 193
```

Lab 46 – Hoán đổi vị trí chẵn lẻ

Viết công thức nhập vào 1 số mảng a từ bàn phím. Hãy hoán đổi 2 giá trị nằm ở vị trí chẵn lẻ chẵn lẻ cạnh nhau. Xuất mảng a ra màn hình

Ví dụ:

0 1 2 3 4 5

8 7 6 1 9 3

Hoán đổi

0 1 2 3 4 5

7 8 1 6 3 9

Lab Mảng Tham Khảo

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace BaiTapVeNha2
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.Write("cho biet so phan tu mang:");
            int n = int.Parse(Console.ReadLine());
            int[] b = new int[n];
            for (int i = 0; i < b.Length; i++)
            {
                Console.WriteLine("nhap so mang[{0}]:", i);
                b[i] = int.Parse(Console.ReadLine());
            }
            for (int i = 0; i <= n/2; i++)
            {
                i++;
                int d = i*1;
                int c = i + 1;
                Console.Write("{0}\t{1}", c, d);
            }
            xuatmangint(b);
            Console.ReadLine();
        }

        static void xuatmangint(int[] b)
        {
            //Xuất các phần tử mảng;
            for (int i = 0; i < b.Length; i++)
            {
                Console.Write("{0}\t", b[i]);
            }
        }
    }
}
```

```

c:\ file:///F:/2011.03.22 on pc17/BaiTapVeNha2/BaiTapVeNha2/bin/Debug/BaiTapVeNha2.EXE
cho biet so phan tu mang:5
nhap so mang[0]:
3
nhap so mang[1]:
6
nhap so mang[2]:
8
nhap so mang[3]:
6
nhap so mang[4]:
9
2      14      33      6      8      6      9      -
    
```

HÀNG ĐỢI (QUEUE) FIFO

Lab 53 - Queue

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Collections;

namespace Lab1
{
    class Program
    {
        static void Main(string[] args)
        {
            //Khoi tao 1 hang doi co kieu int
            Queue<int> q = new Queue<int>();
            //Dua 1 so nguyen vao trong hang doi
            q.Enqueue(10);
            q.Enqueue(20);
            q.Enqueue(30);

            //Lay 1 so nguyen trong hang doi ra
            Console.WriteLine(q.Dequeue());
            //Lay gia tri phan tu dau tien ra xem, khong lay phan tu
ra
            Console.WriteLine(q.Peek());
            Console.WriteLine(q.Dequeue());
            Console.WriteLine(q.Dequeue());

            Console.ReadLine();
        }
    }
}
    
```

```

c:\ file:///E:/THUC HANH LAP TRINH/2011.03.25/Lab1/Lab1/bin/Debug/Lab1.EXE
10
20
20
30
    
```


Bài Tập 1

Tạo 1 hàng đợi có kiểu là Float. Hãy viết theo các cách viết sau

- Thêm vào hàng đợi 3 giá trị bất kỳ
- Lấy từ hàng đợi ra 2 phần tử lần lượt xuất ra màn hình
- Thêm tiếp 2 giá trị bất kỳ
- Thăm dò giá trị của phần tử đầu tiên
- Viết ra màn hình và lấy hết các phần tử còn lại trong hàng đợi (Dùng Stack ---Bài Tập 2)

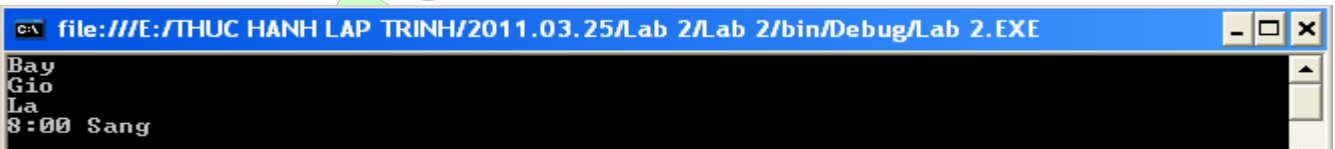
Lab 54 - Queue

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Collections;

namespace Lab_2
{
    class Program
    {
        static void Main(string[] args)
        {
            Queue<string> qs = new Queue<string>();
            qs.Enqueue("Bay");
            qs.Enqueue("Gio");

            qs.Enqueue("La");
            qs.Enqueue("8:00 Sang");

            Console.WriteLine(qs.Dequeue());
            Console.WriteLine(qs.Dequeue());
            Console.WriteLine(qs.Dequeue());
            Console.WriteLine(qs.Dequeue());
            Console.ReadLine();
        }
    }
}
```

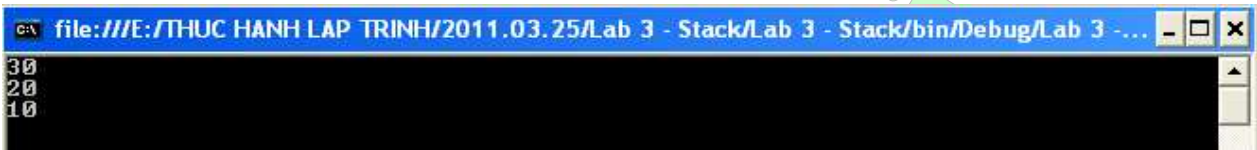
**HÀNG CHÒNG (STACK)****Lab 55 – Stack**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Collections;

namespace Lab_3__Stack
{
    class Program
    {
```

```
static void Main(string[] args)
{
    Stack<int> s = new Stack<int>();
    s.Push(10);
    s.Push(20);
    s.Push(30);

    Console.WriteLine(s.Pop());
    Console.WriteLine(s.Pop());
    Console.WriteLine(s.Pop());
    Console.ReadLine();
}
}
```



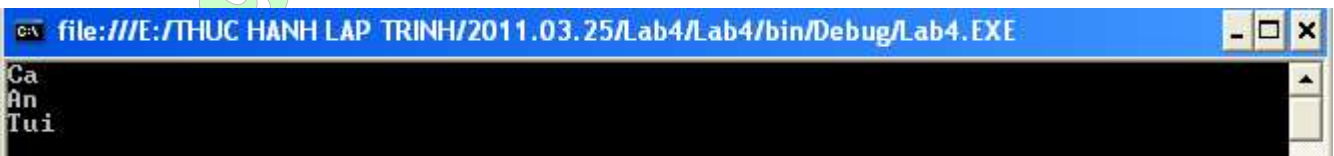
```
C:\> file:///E:/THUC HANH LAP TRINH/2011.03.25/Lab 3 - Stack/Lab 3 - Stack/bin/Debug/Lab 3 -...
30
20
10
```

Lab 56 - Stack

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Lab4
{
    class Program
    {
        static void Main(string[] args)
        {
            Stack<string> ss = new Stack<string>();
            ss.Push("Tui");
            ss.Push("An");
            ss.Push("Ca");

            Console.WriteLine(ss.Pop());
            Console.WriteLine(ss.Pop());
            Console.WriteLine(ss.Pop());
            Console.ReadLine();
        }
    }
}
```



```
C:\> file:///E:/THUC HANH LAP TRINH/2011.03.25/Lab4/Lab4/bin/Debug/Lab4.EXE
Ca
An
Tui
```

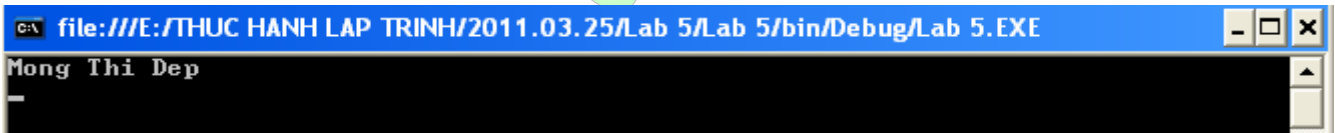
HASHTABLE (KIỂU TỰ ĐIỂN)

Lab 57 - Hashtable

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Collections;

namespace Lab_5
{
    class Program
    {
        static void Main(string[] args)
        {
            Hashtable myDict = new Hashtable();
            myDict.Add("001234", "Nguyen Van Teo");
            myDict.Add("001235", "Nguyen Van Ty");
            myDict.Add("01236", "Le Thi Gai");
            myDict.Add("001237", "Mong Thi Dep");

            Console.WriteLine(myDict["001237"]);
            Console.ReadLine();
        }
    }
}
```



Lab 58 - Hanshtable

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Collections;

namespace Lab_6
{
    class Program
    {
        static void Main(string[] args)
        {
            Hashtable DanhSachHS = new Hashtable();
            bool TiepTuc = false;
            int MaSo;
            string Ten, ans;
            Console.WriteLine("*****");
            Console.WriteLine("        QUAN LY HOC SINH        ");
        }
    }
}
```

```

Console.WriteLine("*****");

Console.WriteLine("\n-----Nhap Thong Tin-----\n");

do
{
    Console.WriteLine("Nhap Ma So HS: ");
    MaSo = int.Parse(Console.ReadLine());
    Console.WriteLine("Nhap Ten Hoc Sinh: ");
    Ten = Console.ReadLine();
    DanhSachHS.Add(MaSo, Ten);
    Console.WriteLine("Ban co muon nhap tiep khong? (Y/N): ");
    ans = Console.ReadLine();
    TiepTuc = "Y".Contains(ans.ToUpper());
} while (TiepTuc);

Console.WriteLine("\n-----Phan Tim Kiem-----\n");

do
{
    Console.WriteLine("Cho biet Ma So Hoc Sinh can tim: ");
    MaSo = int.Parse(Console.ReadLine());
    if (DanhSachHS.Contains(MaSo))
    {
        Console.WriteLine(">>Hoc Sinh tim duoc Ten: {0}",
DanhSachHS[MaSo]);
    }
    else
    {
        Console.WriteLine("Ma So nay khong co trong Danh
Sach Hoc Sinh");
    }

    Console.WriteLine("Ban co muon tim tiep khong? (Y/N): ");
    ans = Console.ReadLine();
    TiepTuc = "Y".Contains(ans.ToUpper());
} while (TiepTuc);
}
}
}

```

```

C:\ file:///E:/THUC HANH LAP TRINH/2011.03.25/Lab 6/Lab 6/bin/Debug/Lab 6.EXE
*****
      QUAN LY HOC SINH
*****
-----Nhap Thong Tin-----

Nhap Ma So HS:
00001
Nhap Ten Hoc Sinh: Nguyen Van Teo
Ban co muon nhap tiep khong? (Y/N): y
Nhap Ma So HS:
00002
Nhap Ten Hoc Sinh: Nguyen Van Ty
Ban co muon nhap tiep khong? (Y/N): y
Nhap Ma So HS:
00003
Nhap Ten Hoc Sinh: Nguyen Van Chuot
Ban co muon nhap tiep khong? (Y/N): n

-----Phan Tim Kiem-----

Cho biet Ma So Hoc Sinh can tim: 5
Ma So nay khong co trong Danh Sach Hoc Sinh
Ban co muon tim tiep khong? (Y/N): y
Cho biet Ma So Hoc Sinh can tim: 00002
>>Hoc Sinh tim duoc Ten: Nguyen Van Ty
Ban co muon tim tiep khong? (Y/N):

```

Lab 60 – Nhập thông tin

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Collections;

namespace BaiLab06
{
    class Program
    {
        static void Main(string[] args)
        {
            Hashtable DanhSachHS = new Hashtable();
            bool TiepTuc = false;
            int MaSo;
            string Ten;
            Console.WriteLine("=====");
            Console.WriteLine("|  CHUONG TRINH QUAN LY HOC SINH  |");
            Console.WriteLine("=====");
            myWrite("          Phan Nhap Thong Tin\n",
ConsoleColor.Yellow);
            myWrite("          -----\n",
ConsoleColor.Yellow);
            do
            {
                myWrite("Nhap Ma So Hoc Sinh: ", ConsoleColor.Cyan);
                MaSo = int.Parse(Console.ReadLine());
                myWrite("Nhap Ten Hoc Sinh: ", ConsoleColor.Cyan);
                Ten = Console.ReadLine();
                DanhSachHS.Add(MaSo, Ten);
                TiepTuc = HoiTiepTuc("Ban co muon nhap tiep ko?");
            } while (TiepTuc);

```

```

myWrite("          Phan Tim Kiem\n", ConsoleColor.Yellow);
myWrite("          -----\n", ConsoleColor.Yellow);
do
{
    myWrite("Cho biet Ma So Hoc Sinh can tim: ",
ConsoleColor.Magenta);
    MaSo = int.Parse(Console.ReadLine());
    if (DanhSachHS.Contains(MaSo))

        {
            myWrite(">> Hoc sinh tim duoc ten: " +
DanhSachHS[MaSo] + "\n", ConsoleColor.Green);
        }
    else
        {
            myWrite("Ma so nay khong co trong Danh Sach Hoc
Sinh!!\n", ConsoleColor.White);
        }
    TiepTuc = HoiTiepTuc("Ban co muon tim tiep ko?");
} while (TiepTuc);
}

static bool HoiTiepTuc(string msg)
{
    string ans;
    myWrite(msg + "(Y/N)", ConsoleColor.Red);
    ans = Console.ReadLine();
    return "Y".Contains(ans.ToUpper());
}

static void myWrite(string msg, ConsoleColor color)
{
    Console.ForegroundColor = color;
    Console.Write(msg);
}
}
}

```

```

C:\ file:///F:/2011.03.25/Bailab06/Bailab06/bin/Debug/Bailab06.EXE
=====
!  CHUONG TRINH QUAN LY HOC SINH  !
=====
      Phan Nhap Thong Tin
-----
Nhap Ma So Hoc Sinh: 01
Nhap Ten Hoc Sinh: Nguyen Van Sang
Ban co muon nhap tiep ko?(Y/N)y
Nhap Ma So Hoc Sinh: 02
Nhap Ten Hoc Sinh: Nguyen Van Ty
Ban co muon nhap tiep ko?(Y/N)y
Nhap Ma So Hoc Sinh: 3
Nhap Ten Hoc Sinh: Nguyen Van Teo
Ban co muon nhap tiep ko?(Y/N)y
Nhap Ma So Hoc Sinh: 4
Nhap Ten Hoc Sinh: Nguyen Van Thuy Lieu
Ban co muon nhap tiep ko?(Y/N)n
      Phan Tim Kiem
-----
Cho biet Ma So Hoc Sinh can tim: 4
>> Hoc sinh tim duoc ten: Nguyen Van Thuy Lieu
Ban co muon tim tiep ko?(Y/N)y
Cho biet Ma So Hoc Sinh can tim: 5
Ma so nay khong co trong Danh Sach Hoc Sinh!!
Ban co muon tim tiep ko?(Y/N)y
Cho biet Ma So Hoc Sinh can tim: 1
>> Hoc sinh tim duoc ten: Nguyen Van Sang
Ban co muon tim tiep ko?(Y/N)

```

KIỂU DỮ LIỆU CÓ CẤU TRÚC STRUCT

Lab 61 Nhập thông tin học sinh

```

struct HocSinh
{
    public int maso;
    public string ten;
    public float diemtoan;
    public float diem ly;
    public float diem hoa;
}
HocSinh[ ] dshs = new HocSinh[50];

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Lab 61
{
    class Program
    {
        struct HocSinh
        {

```

```

public int MaSo;
public string Ten;
public float DiemToan,
           DiemLy,
           DiemHoa,
           DiemTB;
}
static void Main(string[] args)
{
    HocSinh[] dshs = new HocSinh[5];

    //Phan nhap thong tin
    for (int i = 0; i < dshs.Length; i++)
    {
        Console.WriteLine();
        Console.WriteLine("Nhap thong tin hoc sinh thu " + i);
        Console.WriteLine("-----");
        Console.Write("Nhap vao MA SO Hoc Sinh: ");
        dshs[i].MaSo = int.Parse(Console.ReadLine());
        Console.Write("Nhap vao TEN Hoc Sinh: ");
        dshs[i].Ten = Console.ReadLine();
        Console.Write("Nhap vao diem TOAN Hoc Sinh: ");
        dshs[i].DiemToan = float.Parse(Console.ReadLine());
        Console.Write("Nhap vao diem LY Hoc Sinh: ");
        dshs[i].DiemLy = float.Parse(Console.ReadLine());
        Console.Write("Nhap vao diem HOA Hoc Sinh: ");
        dshs[i].DiemHoa = float.Parse(Console.ReadLine());
        dshs[i].DiemTB = (dshs[i].DiemToan + dshs[i].DiemLy +
dshs[i].DiemHoa) / 3;
    }

    Console.WriteLine();
    //Phan xuat thong tin
    for (int i = 0; i < dshs.Length; i++)
    {
        Console.WriteLine();

        Console.WriteLine("Xuat thong tin hoc sinh thu " + i);
        Console.WriteLine("-----");
        Console.WriteLine("■ Ma So Hoc Sinh: " +
dshs[i].MaSo);
        Console.WriteLine("■ Ten Hoc Sinh: " + dshs[i].Ten);
        Console.WriteLine("■ Diem Toan: " + dshs[i].DiemToan);
        Console.WriteLine("■ Diem Ly: " + dshs[i].DiemLy);
        Console.WriteLine("■ Diem Hoa: " + dshs[i].DiemHoa);
        Console.WriteLine("■ Diem TB: " + dshs[i].DiemTB);
    }
    Console.ReadLine();
}
}
}

```



```

file:///E:/29.03.2011/Thong tin hoc sinh/Thong tin hoc sinh/bin/Debug/Thong tin hoc sinh.EXE
*****
Nhap vao ma so Ho Sinh: 123456
Nhap vào Ten Hoc Sinh: Nguyen Van A
Nhap vao diem Toan hoc sinh: 9
Nhap vao diem Ly hoc sinh: 8
Nhap vao diem Hoa hoc sinh: 6

Nhap thong tin hoc sinh thu: 1
*****
Nhap vao ma so Ho Sinh: 123457
Nhap vào Ten Hoc Sinh: Nguyen Van B
Nhap vao diem Toan hoc sinh: 7
Nhap vao diem Ly hoc sinh: 3
Nhap vao diem Hoa hoc sinh: 9

Nhap thong tin hoc sinh thu: 2
*****
Nhap vao ma so Ho Sinh: 123458
Nhap vào Ten Hoc Sinh: Nguyen Van C
Nhap vao diem Toan hoc sinh: 5
Nhap vao diem Ly hoc sinh: 9
Nhap vao diem Hoa hoc sinh: 10

Nhap thong tin hoc sinh thu: 3
*****
Nhap vao ma so Ho Sinh:

```

Lab 52 Cách 2

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Lab 62
{
    class Program
    {
        struct HocSinh
        {
            public int MaSo;
            public string Ten;
            public float DiemToan,
                DiemLy,
                DiemHoa,
                DiemTB;

            public void NhapThongTin()
            {
                Console.Write("Nhap vao MA SO Hoc Sinh: ");
                MaSo = int.Parse(Console.ReadLine());
                Console.Write("Nhap vao TEN Hoc Sinh: ");
                Ten = Console.ReadLine();
                Console.Write("Nhap vao diem TOAN Hoc Sinh: ");
                DiemToan = float.Parse(Console.ReadLine());
                Console.Write("Nhap vao diem LY Hoc Sinh: ");
                DiemLy = float.Parse(Console.ReadLine());

                Console.Write("Nhap vao diem HOA Hoc Sinh: ");
                DiemHoa = float.Parse(Console.ReadLine());
                DiemTB = (DiemToan + DiemLy + DiemHoa) / 3;
            }

            public void XuatThongTin()

```

```

    {
        Console.WriteLine("■ Ma So Hoc Sinh: " + MaSo);
        Console.WriteLine("■ Ten Hoc Sinh: " + Ten);
        Console.WriteLine("■ Diem Toan: " + DiemToan);
        Console.WriteLine("■ Diem Ly: " + DiemLy);
        Console.WriteLine("■ Diem Hoa: " + DiemHoa);
        Console.WriteLine("■ Diem TB: " + DiemTB);
    }
}

static void Main(string[] args)
{
    HocSinh[] dshs = new HocSinh[5];
    //Phan nhap thong tin
    for (int i = 0; i < dshs.Length; i++)
    {
        Console.WriteLine();
        Console.WriteLine("Nhap thong tin hoc sinh thu " + i);
        Console.WriteLine("-----");
        dshs[i].NhapThongTin();
    }

    Console.WriteLine();
    //Phan xuat thong tin
    for (int i = 0; i < dshs.Length; i++)
    {
        Console.WriteLine();
        Console.WriteLine("Xuat thong tin hoc sinh thu " + i);
        Console.WriteLine("-----");
        dshs[i].XuatThongTin();
    }
    Console.ReadLine();
}
}
}

```

```

file:///E:/29.03.2011/Thong tin hoc sinh/Thong tin hoc sinh/bin/Debug/Thong tin hoc sinh.EXE
*****
Nhap vao ma so Ho Sinh: 123456
Nhap vào Ten Hoc Sinh: Nguyen Van A
Nhap vao diem Toan hoc sinh: 9
Nhap vao diem Ly hoc sinh: 8
Nhap vao diem Hoa hoc sinh: 6

Nhap thong tin hoc sinh thu: 1
*****
Nhap vao ma so Ho Sinh: 123457
Nhap vào Ten Hoc Sinh: Nguyen Van B
Nhap vao diem Toan hoc sinh: 7
Nhap vao diem Ly hoc sinh: 3
Nhap vao diem Hoa hoc sinh: 9

Nhap thong tin hoc sinh thu: 2
*****
Nhap vao ma so Ho Sinh: 123458
Nhap vào Ten Hoc Sinh: Nguyen Van C
Nhap vao diem Toan hoc sinh: 5
Nhap vao diem Ly hoc sinh: 9
Nhap vao diem Hoa hoc sinh: 10

Nhap thong tin hoc sinh thu: 3
*****
Nhap vao ma so Ho Sinh:

```

Lab 63 – Nhập thông tin Hàng hóa

Hãy tạo 1 struct hàng hóa có các biến thành viên sau:

Mã hàng	int
Tên Hàng	string
Đơn giá	float
Số lượng	int
Thành tiền	float (SL * DG)

Trong struct hàng hóa, hãy viết 2 Method là nhập thông tin và xuất thông tin

Trong CT chính (main) hãy tạo 1 mảng hàng hóa có 5 phần tử. Dùng vòng lặp để gọi nhập thông tin và xuất thông tin của mỗi hàng hóa có trong dshh

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Hang_hoa
{
    class Program
    {
        struct HangHoa
        {
            public int MaHang,
                SoLuong;
            public string TenHang;
            public float DonGia,
                ThanhTien;
        }
        static void Main(string[] args)
        {
            HangHoa[] dshh = new HangHoa[5];
            //Phan nhap thong tin
            for (int i = 0; i < dshh.Length; i++)
            {
                Console.WriteLine();
                Console.WriteLine("Nhap thong tin hang hoa: " + i);
                Console.WriteLine("*****");
                Console.Write("Nhap Ma Hang: ");
                dshh[i].MaHang = int.Parse(Console.ReadLine());
                Console.Write("Nhap Ten Hang: ");
                dshh[i].TenHang = Console.ReadLine();
                Console.Write("Nhap Don Gia: ");
                dshh[i].DonGia = float.Parse(Console.ReadLine());
                Console.Write("Nhap So Luong: ");
                dshh[i].SoLuong = int.Parse(Console.ReadLine());
                Console.Write("Thanh Tien: ");
            }
            Console.WriteLine();
            //Phan xuat thong tin
            for (int i = 0; i < dshh.Length; i++)
            {
                Console.WriteLine();
                Console.WriteLine("Xuat thong tin hang hoa: " + i);
                Console.WriteLine("*****");
                Console.WriteLine(" ■ Ma Hang: " + dshh[i].MaHang);
            }
        }
    }
}
```

```

        Console.WriteLine(" ■ Ten Hang: " + dshh[i].TenHang);
        Console.WriteLine(" ■ Don Gia: " + dshh[i].DonGia);
        Console.WriteLine(" ■ So Luong: " + dshh[i].SoLuong);
        Console.WriteLine(" ■ Thanh Tien: " +
dshh[i].ThanhTien);
    }
    Console.ReadLine();
}
}
}

```

Lab 64 – Nhập thông tin lao động

Hãy tạo 1 struct Lao Động gồm các biến thành viên sau:

Mã LD	int	
Tên LD	string	
Năm Sinh	int	
Mức lương	float	
Ngày công	int	
Lương lãnh	float	Mức lương * Ngày công
Tuổi	int	Năm (hh) – Năm Sinh

Trong struct Lao Động, hãy viết 2 method là Nhập Thông Tin và Xuất Thông Tin. Trong CT chính (main) hãy tạo 1 mảng dsld gồm có 5 phần tử. Dùng vòng lặp để gọi nhập thông tin và xuất thông tin của mỗi lao động có trong dsld

OBJECT

Khai báo với từ khóa Class

```

Class <Tên lớp>
{
    Các biến thành viên;
    ...
    Các phương thức;
    ...
}

```

2. Các phương thức (Hàm/thủ tục) khởi tạo có tham số và không có tham số

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Lab_65
{
    class HìnhHoc
    {
        //Các biến thành viên
        public string TenLop;

        //Các phương thức khởi tạo
        //Không tham số
        public HìnhHoc()
        {
            TenLop = "Lop chua biet";
        }

        //Có tham số

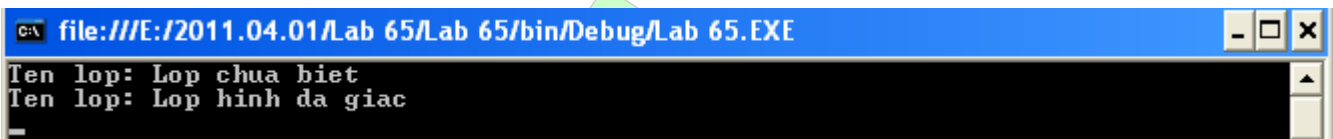
```

```

public HinhHoc(string TenLop)
{
    this.TenLop = TenLop;
}
//Cac phuong thuc thong thuong
public void NhapThongTin()
{
}

public void XuatThongTin()
{
    Console.WriteLine("Ten lop: {0}", TenLop);
}
}
class Program
{
    static void Main(string[] args)
    {
        HinhHoc hh1 = new HinhHoc();
        hh1.XuatThongTin();
        HinhHoc hh2 = new HinhHoc("Lop hinh da giac");
        hh2.XuatThongTin();
        Console.ReadLine();
    }
}
}

```



```

file:///E:/2011.04.01/Lab 65/Lab 65/bin/Debug/Lab 65.EXE
Ten lop: Lop chua biet
Ten lop: Lop hinh da giac

```

Lab 65

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Lab_66
{
    class HocSinh
    {
        public int MaSo;
        public string Ten;
        public float DiemToan,
            DiemLy,
            DiemHoa,
            DiemTB;
        //Ham Khoi tao
        //Khong tham so
        public HocSinh()
        {
            MaSo = 0;
            Ten = "Chua biet";
        }
    }
}

```

```

        DiemToan = DiemLy = DiemHoa = DiemTB = 0;
    }

    //Co tham so
    public HocSinh(int aMaSo, string Ten, float DiemToan, float
DiemLy, float DiemHoa)
    {
        MaSo = aMaSo;
        this.Ten = Ten;
        this.DiemToan = DiemToan;
        this.DiemLy = DiemLy;
        this.DiemHoa = DiemHoa;
        DiemTB = (DiemToan + DiemLy + DiemHoa) / 3;
    }

    //Cac phuong thuc khac
    public void InThongTin()
    {
        Console.WriteLine("Ma So: {0}", MaSo);
        Console.WriteLine("Ten: {0}", Ten);
        Console.WriteLine("Diem Toan: {0}", DiemToan);
        Console.WriteLine("Diem Ly: {0}", DiemLy);
        Console.WriteLine("Diem Hoa: {0}", DiemHoa);
        Console.WriteLine("Diem TB: {0}", DiemTB);
    }
}

class Program
{
    static void Main(string[] args)
    {
        HocSinh hs1 = new HocSinh();
        hs1.InThongTin();
        HocSinh hs2 = new HocSinh(1, "Nguyen Van A", 8.5f, 9,
7.5f);
        hs2.InThongTin();
        Console.ReadLine();
    }
}

```

```

C:\> file:///E:/2011.04.01/Lab 66/Lab 66/bin/Debug/Lab 66.EXE
Ma So: 0
Ten: Chua biet
Diem Toan: 0
Diem Ly: 0
Diem Hoa: 0
Diem TB: 0
Ma So: 1
Ten: Nguyen Van A
Diem Toan: 8.5
Diem Ly: 9
Diem Hoa: 7.5
Diem TB: 8.333333

```

3. Tính thừa kế của object

```
using System;
using Lab67;

namespace Lab67
{
    class HìnhHoc
    {
        //Cac bien thanh vien
        public string TenLop;

        //Cac phuong thuc khoi tao
        //Khong tham so
        public HìnhHoc()
        {
            TenLop = "Lop chua biet";
        }

        //Co tham so
        public HìnhHoc(string TenLop)
        {
            this.TenLop = TenLop;
        }
        //Cac phuong thuc thong thuong
        public void NhapThongTin()
        {
        }

        public void XuatThongTin()
        {
            Console.WriteLine("Ten lop: {0}", TenLop);
        }
    }

    class HìnhVuong : HìnhHoc
    {
        float Canh,
            DienTich,
            ChuVi;

        //Ham khoi tao
        //Khong tham so
        public HìnhVuong()
        {
            TenLop = "Hình Vuong";
            Canh = DienTich = ChuVi = 0;
        }
        public HìnhVuong(float Canh)
        {
            TenLop = "Hình Vuong";
            this.Canh = Canh;
            DienTich = Canh * Canh;
        }
    }
}
```

```

        ChuVi = 4 * Canh;
    }

    public void InThongTin()
    {
        XuatThongTin();
        Console.WriteLine("Canh: {0}", Canh);
        Console.WriteLine("Dien Tich: {0}", DienTich);
        Console.WriteLine("Chu Vi: {0}", ChuVi);
    }
}
class Program
{
    static void Main(string[] args)
    {
        HinhVuong hv1 = new HinhVuong();
        hv1.InThongTin();
        HinhVuong hv2 = new HinhVuong(10);
        hv2.InThongTin();
        Console.ReadLine();
    }
}
}

```

```

file:///E:/2011.04.01/Lab 67 Tinh thua ke object/Lab 67 Tinh thua ke object/bin/Debug/La...
Ten lop: Hinh Vuong
Canh: 0
Dien Tich: 0
Chu Vi: 0
Ten lop: Hinh Vuong
Canh: 10
Dien Tich: 100
Chu Vi: 40

```

Lab 68

Từ bài Lab 65. Hãy bổ sung thêm 2 lớp con từ lớp Hình Học là hình chữ nhật và hình tròn. Trong chương trình chính hãy khởi tạo 2 biến thuộc lớp trên có tham số và không tham số, sau đó gọi method (phương thức) In thông tin() để kiểm tra

Lab 69

Trong 1 lớp cha có tên là học sinh, trong đó có tên và Điểm TB

- Tạo lớp con SV ĐH có 6 cột điểm Toán, Tin, Văn, Lịch Sử, Quốc Phòng, Pháp Luật. Điểm Toán và Điểm Văn là hệ số nhân 2

Lab 70

Trong 1 lớp cha có tên là học sinh, trong đó có tên và Điểm TB

```

namespace Lab_70
{
    class HocSinh
    {
        public string Ten;
        public float DiemTB;

        //Khởi tạo không tham số
        public HocSinh()
        {
            Ten = "Chưa có Tên";
            DiemTB = 0;
        }
    }
}

```



```
//Khởi tạo có tham số
public HocSinh(string Ten, float DiemTB)
{
    this.Ten = Ten;
    this.DiemTB = DiemTB;
}

// Các phương thức
public string XepLoai()
{
    if (DiemTB >= 9)
        return "Gioi";
    else if (DiemTB >= 7)
        return "Kha";
    else if (DiemTB >= 5)
        return "TB";
    else if (DiemTB > 3)
        return "Yeu";
    else return "Kem";
}
}
class HocSinhCap1 : HocSinh
{
    float DiemToan, DiemVan;
    //Hàm khởi tạo không tham số
    public HocSinhCap1()
    {
        DiemToan = 0;
        DiemVan = 0;
    }

    public HocSinhCap1(string Ten, float DiemToan, float DiemVan)
    {
        this.Ten = Ten;
        this.DiemToan = DiemToan;
        this.DiemVan = DiemVan;
        DiemTB = (DiemToan + DiemVan) / 2;
    }
    public void InThongTin()
    {
        System.Console.WriteLine("Ten Hoc Sinh: {0}", Ten);
        System.Console.WriteLine("Diem TB: {0}", DiemTB);
        System.Console.WriteLine("Xep Loai: {0}", XepLoai());
    }
}
class HocSinhCap2 : HocSinh
{
    float DiemToan, DiemVan, DiemLy, DiemHoa;
    //Hàm khởi tạo không tham số
    public HocSinhCap2()
    {
        DiemToan = 0;
        DiemVan = 0;
        DiemLy = 0;
    }
}
```

```

        DiemHoa = 0;
    }

    public HocSinhCap2(string Ten, float DiemToan, float DiemVan,
float DiemLy, float DiemHoa)
    {

        this.Ten = Ten;
        this.DiemToan = DiemToan;
        this.DiemVan = DiemVan;
        this.DiemLy = DiemLy;
        this.DiemHoa = DiemHoa;
        DiemTB = (DiemToan + DiemVan + DiemLy + DiemHoa) / 4;
    }
    public void InThongTin()
    {

        System.Console.WriteLine("Ten Hoc Sinh: {0}", Ten);
        System.Console.WriteLine("Diem TB: {0}", DiemTB);
        System.Console.WriteLine("Xep Loai: {0}", XepLoai());
    }
}

class Program
{
    static void Main(string[] args)
    {
        HocSinhCap1 Teo = new HocSinhCap1("Nguyen Van Teo", 10,
7);

        Teo.InThongTin();
        HocSinhCap2 aTeo = new HocSinhCap2("Nguyen Van Teo", 8, 6,
8, 7);

        aTeo.InThongTin();
        System.Console.ReadLine();
    }
}
}

```

```

C:\> file:///E:/2011.04.01/Lab 70/Lab 70/bin/Debug/Lab 70.EXE
Ten Hoc Sinh: Nguyen Van Teo
Diem TB: 8.5
Xep Loai: Kha
Ten Hoc Sinh: Nguyen Van Teo
Diem TB: 7.25
Xep Loai: Kha

```