



# Bài tập môn lập trình hợp ngữ có đáp án

## Bài 1: Viết chương trình hiện ra câu "Hello Assembly"

```
.MODEL small
.STACK 100h
.DATA
Message db 'Hello Assembly!$'
.CODE
ProgramStart:
Mov AX,@DATA
Mov DS,AX
Mov DX,OFFSET Message
Mov AH,9
Int 21h
Mov AH,4Ch
Int 21h
END ProgramStart
```

## Bài 2: Viết chương trình hiện ra hai câu "Chao mung ban den voi Assembly"

"Assembly that de!". Mỗi câu trên một dòng.

```
.MODEL small
.STACK 100h
.DATA
Tbao1 db "Chao mung ban den voi Assembly$"
Tbao2 db 0DH,0AH,"Assembly that de!$"
.CODE
ProgramStart:
Mov AX,@DATA
Mov DS,AX
;xuất thông báo 1
```

```
Lea DX,Tbao1
Mov AH,9
Int 21h
;xuất thông báo 2
Lea DX,Tbao2
Mov AH,9
Int 21h
Mov AH,4Ch
Int 21h
END ProgramStart
```

### **Bài 3: Viết chương trình yêu cầu nhập một ký tự và xuất ra màn hình ký tự vừa nhập**

```
.model small
.stack
.data
TBao1 db "Hay nhap mot ky tu: $"
TBao2 db 0DH,0AH,"Ky tu da nhap: $"
KyTu db ?
.code
ProgramStart:
Mov ax,@data
Mov ds,ax
;nhac nhap
Lea dx, TBao1
Mov ah, 9
int 21h
;nhap 1 ky tu
Mov ah, 1
Int 21h
Mov KyTu, al
```

```

; thong bao ket qua
lea dx, TBao2
mov ah, 9
int 21h
; hien thi ky tu da nhap
Mov ah, 2
Mov dl, KyTu
Int 21h
; ve dos
Mov ah, 4Ch
Int 21h
end

```

**Bài 4: Viết chương trình nhập vào một ký tự. Xuất ra màn hình ký tự đứng trước và ký tự đứng sau.**

```

.model small
.stack
.data
TBao1 db "Hay nhap mot ky tu: $"
TBao2 db 0DH,0AH,"Ky tu da nhap: $"
TBao3 db 0DH,0AH,"Ky tu truoc ky tu da nhap: $"
TBao4 db 0DH,0AH,"Ky tu sau ky tu da nhap: $"
KyTu db ?
.code
ProgramStart:
Mov ax,@data
Mov ds,ax
; nhac nhap
Lea dx, TBao1
Mov ah, 9
int 21h

```

```
; nhap 1 ky tu
Mov ah, 1
Int 21h
Mov KyTu, al
; thong bao ket qua
lea dx, TBao2
mov ah, 9
int 21h
; hien thi ky tu da nhap
Mov ah, 2
Mov dl, KyTu
Int 21h
; hien thi ky tu truoc ky tu nhap
lea dx, TBao3
mov ah, 9
int 21h
mov ah, 2
mov dl, KyTu
dec dl
int 21h
; hien thi ky tu sau ky tu nhap
lea dx, TBao4
mov ah, 9
int 21h
mov ah, 2
mov dl, KyTu
inc dl
int 21h
mov ah, 4Ch
int 21h
```

end

### **Bài 5: Viết chương nhập vào một ký tự. Chuyển ký tự đó sang ký tự hoa**

.MODEL SMALL

.STACK 100h

.DATA

Msg1 DB 'Nhap vao ki tu thuong : \$'

Msg2 DB 0Dh,0Ah,'Chuyen sang ki tu hoa la : '

Char DB ?, '\$'

.CODE

Main PROC

MOV AX,@DATA

MOV DS,AX

; In ra thong bao 1

LEA DX,Msg1

MOV AH,9

INT 21h

; Nhap vao 1 ki tu thuong va doi thanh ki tu hoa

MOV AH,1

INT 21h ; Doc 1 ki tu thuong va luu vao AL

SUB AL,20h ; Doi thanh ki tu hoa

MOV Char,AL

; Hien len chu hoa

LEA DX,Msg2

MOV AH,9

INT 21h

; Ket thuc chuong trinh

MOV AH,4Ch

INT 21h

Main ENDP

END Main

### **Bài 6: Chuyển đổi ký tự hoa thành ký tự thường**

.MODEL SMALL

.STACK 100h

.DATA

Msg1 DB 'Nhap vao ki tu hoa : \$'

Msg2 DB 0Dh,0Ah,'Chuyen sang ki tu thuong la : '

Char DB ?, '\$'

.CODE

Main PROC

MOV AX,@DATA

MOV DS,AX

; In ra thong bao 1

LEA DX,Msg1

MOV AH,9

INT 21h

; Nhap vao 1 ki tu hoa va doi thanh ki tu thuong

MOV AH,1

INT 21h ; Doc 1 ki tu hoa va luu vao AL

ADD AL,20h ; Doi thanh ki tu thuong

MOV Char,AL

; Hien len chu thuong

LEA DX,Msg2

MOV AH,9

INT 21h

; Ket thuc chuong trinh

MOV AH,4Ch

INT 21h

Main ENDP

END Main

nhap chuoi

data segment

tb1 db "Hay nhap vao mot chuoi ky tu: \$"

tb2 db 13, 10,"chuoi vua nhap la: \$"

str db 200,?,200 dup('\$')

ends

stack segment

dw 128 dup(?)

ends

code segment

mov ax,data

mov ds,ax

lea dx,tb1

mov ah,09h

int 21h

mov ah,0Ah

lea dx,str

int 21h

lea DX,tb2

mov ah,09h

int 21h

lea bx,str

mov al,{bx+01h}

mov ah,00h

add bx,ax



```
;mov [bx+2], "$"  
mov ah,09h  
lea dx,str+2  
int 21h  
ends
```

**Bài 7: Viết chương trình nhập vào một chuỗi. In ra màn hình chuỗi thường, chuỗi in. Dừng chương con.**

```
.MODEL small  
.STACK  
.DATA  
tb1 DB 'Nhap vao 1 chuoai: $'  
tb2 DB 10,13,'Doi thanh chu thuong: $'  
tb3 DB 10,13,'Doi thanh chu hoa: $'  
s DB 100,?,101 dup('$')  
.CODE  
BEGIN:  
MOV AX, @DATA  
MOV DS,AX  
;xuat chuoai tb1  
MOV AH,09h  
LEA DX,tb1  
INT 21h  
;nhap chuoai s  
MOV AH,0AH  
LEA DX,s  
INT 21h  
;xuat chuoai tb2  
MOV AH,09h
```

LEA DX,tb2

INT 21h

; Goi chuong trinh con in chuoi thuong

CALL InChuoiThuong

; xuat chuoi tb3

MOV AH,09h

LEA DX,tb3

INT 21h

; Goi chuong trinh con in chuoi thuong

CALL InChuoiHoa

MOV AH,4ch

INT 21h

,\*\*\*\*\*

; Doi thanh chuoi ky tu thuong

InChuoiThuong PROC

LEA SI,s+1

XOR CX,CX

MOV CL,[SI]

INC SI

LapThuong:

MOV AH,02h

MOV DL,[SI]

CMP DL,'A'

JB LT1

CMP DL,'Z'

JA LT1

ADD DL,32

LT1: INC SI

```
INT 21h
LOOP LapThuong
RET
InChuoiThuong ENDP
; Doi thanh chuoi ky tu hoa
InChuoiHoa PROC
LEA SI,s+1
XOR CX,CX
MOV CL,[SI]
INC SI
LapHoa:
MOV AH,02h
MOV DL,[SI]
CMP DL,'a'
JB LH1
CMP DL,'z'
JA LH1
SUB DL,32
LH1: INC SI
INT 21h
LOOP LapHoa
RET
InChuoiHoa ENDP

END BEGIN
```

**Bài 8: Viết chương trình nhập vào một chuỗi. Đếm chiều dài của chuỗi nhập vào.**

```
.MODEL small
.STACK
.DATA
```

```
tb1 DB 'Nhap vao 1 chuoi: $'  
tb2 DB 10,13,'Tong chieu dai cua chuoi: $'  
s DB 100,?,101 dup('$')  
.CODE  
BEGIN:  
MOV AX, @DATA  
MOV DS,AX  
;xuat chuoi tb1  
MOV AH,09h  
LEA DX,tb1  
INT 21h  
;nhap chuoi s  
MOV AH,0AH  
LEA DX,s  
INT 21h  
;xuat chuoi tb2  
MOV AH,09h  
LEA DX,tb2  
INT 21h  
;Tinh chieu dai chuoi  
XOR AX,AX  
MOV AL,s+1 ;Chuyen chieu dai chuoi vao ax  
MOV CX,0 ;Khoi tao bien dem  
MOV BX,10  
LapDem1:  
MOV DX,0  
DIV BX  
PUSH DX  
INC CX  
CMP AX,0
```

```
JNZ LapDem1
;xuât chieu dai chuoï
MOV AH,2
LapDem2:
POP DX
OR DL,'0' ;chuyen chu so -> so
INT 21H
LOOP LapDem2

MOV AH,4ch
INT 21h
END BEGIN
```

# Một số các mẫu bài tập lập trình hợp ngữ

**Bài 1: Lập chương trình nhập vào 2 số kiểu byte, in ra màn hình tích 2 số vừa nhập**

```
.model tiny
.stack 100h
.data
tb1 db 'nhap a=$'
tb2 13,10,'nhap b=$'
tb3 13,10,'tich 2 so la:$'
so1 db 0
so2 db 0
.code
main proc
Mov ax,@Data
Mov ds,ax
Mov ah,9
;in thong bao nhap so thu 1
Lea dx,tb1
int 21h
nhap1:
mov ah,1
int 21h
cmp al,13 ;so sanh al voi 13
```

```
je nhap2 ;neu bang thi nhay den nhap 2
sub al,30h ; chuyen ky tu thanh so
mov dl,al ;cat al vao dl
mov al,so1 ; dua so vua nhap ve kieu byte
mov bl,10 ;gan bl =10
mul bl ;nhan al voi 10
add al,dl ;lay ket qua vua nhan cong voi so vua nhap
mov s01,al ;cat ket qua sau khi doi vao bien so1
jmp nhap1 ;nhay den nhan nhap 1
nhap2:
lea dx,tb2; in thong bao nhap so thu 2
mov ah,9
int 21h
nhap: mov ah,1
int 21h
cmp al,13 ; so sanh so vua nhap voi enter
je tinhlich ;neu bang thi tinh lich
sub al,30h ; chuyen xau vua nhap thanh so
mov dl,al ;cat so vua nhap vao dl
mov al,so2 ;dua so vua nhap ve kieu byte
mov bl,10 ;gan bl=10
mul bl ; lay so ban dau nhan voi 10
add al,dl ;lay ket qua vua nhan cong voi so vua nhap
mov s02,al ;cat ket qua sau khi doi vao bien so2
```

```
jmp nhap
tinhtich:
mov al,so1 ;dua so vua nhap ra thanh ghi al
mul so2 ;nhan voi so 2
mov bx,ax ;lay ket qua vua tinh chuyen vao thanh ghi bx
jmp tinhtich
;in tich
mov ah,9 ;hien thong bao in tich
lea dx,tb3
int 21h
mov ax,bx ;chuyen ket qua ra thanh ghi ax
mov bx,10 ;gan bx=10
xor cx,cx ;khoi tao bien dem
chia: xor dx,dx ;xoa bit cao
div bx ;lay ket qua chia cho 10 du dat dx,thuong dat ax
push dx ;day du trong dx vao ngan xep
inc cx ;tang bien dem
cmp ax,0 ;so sanh thuong voi 0
ja chia ;neu lon hon thi chia
mov ah,2 ;lay chuc nang in ky tu ra man hinh
lay: pop dx ;lay du trong stasck khoi dx
add dl,30h ;chuyen so vua nhap sang dang ky tu
int 21h ;thuc hien in ky tu nam trong dl ra man hinh
loop lay
```



```
ra: mov ah,4ch
```

```
int 21h
```

```
Main endp
```

```
End main
```

## **Bài 2 : Lập chương trình nhập vào 2 số kiểu word,in ra màn hình tổng 2 số vừa nhập**

```
.model tiny
```

```
.stack 100h
```

```
.data
```

```
tb1 db 'nhap so thu 1:$'
```

```
tb2 db 13,10 ,'nhap so thu 2:$'
```

```
tb3 db 13,10,'tong 2 so$'
```

```
so1 dw 0
```

```
so2 dw 0
```

```
tong dw 0
```

```
.code
```

```
main proc
```

```
mov ax,@data
```

```
mov ds,ax
```

```
;in thong bao nhap so thu nhat
```

```
lea dx,tb1
```

```
mov ah,9
```

```
int 21h
```

```
nhap1:
```

```

mov ah,1
int 21h
cmp al,13 ;so sanh ky tu vua nhap voi 13
je nhap2 ;neu bang nhap so thu 2
sub al,30h ;doi ky tu sang so
mov ah,0 ;xoa bit cao
mov cx,ax ;cat so vua nhap vào cx
mov ax,so1 ;đưa biến số 1 về kiểu byte để chuẩn bị nhân với 10
mov bx,10 ;gan bx =10
mul bx ; nhân ax với 10
add ax,cx ;cộng kết quả vừa nhân với số vừa nhập kết quả cất vào ax
mov so1,ax ; cất kết quả vào biến số1
jmp nhap1
nhap2:
lea dx,tb2 ;hiển thông báo nhập số thu 2
mov ah,9
int 21h
nhap: mov ah,1 ;nhập số thu 2
int 21h
cmp al,13 ;so sánh ký tu vua nhập voi 13
je tinhcong ;nếu bằng thì tính tổng
sub al,30h ;chuyển ký tu sang dạng số
mov ah,0 ;xóa bit cao
mov cx,ax ;cất kết quả vừa nhập vào cx

```

```
mov ax,so2 ;đưa biên số 2 về kiểu byte
mov bx,10 ;gán bx=10
mul bx ;nhân kết quả vừa nhập với 10
add ax,cx ;cộng kết quả vừa nhân với số vừa nhập
mov so2,ax ;cất kết quả vào biên số 2
jmp nhập
tinhtong:
mov dx,tong
mov ax,so1 ;đưa biên số 1 ra thành ghi ax
mov bx,so2 ;đưa biên số 2 ra thành ghi bx
add ax,bx ;cộng ax với bx kết quả cất vào ax
mov tong,ax ;đưa kết quả từ ax vào biên tổng
inso: mov ah,9 ;hiển thông báo in tổng
lea dx,tb3
int 21h
mov ax,tong ;đưa kết quả trong biến tổng ra thành ghi ax
mov dx,0 ;xóa bit cao dx
mov bx,10 ;gán bx=10
mov cx,0 ;khởi tạo biên đếm
chia: div bx ;lấy kết quả chia cho 10
push dx ;đưa dx đây vào ngăn xếp
inc cx ;tăng biên đếm
cmp ax,0 ;so sánh thương với 0
je hienkq ;nếu bằng thì hiển kết quả
```

```
xor dx,dx ;xoa bit cao trong dx
jmp chia
hienkq: pop dx ;lây du trong ngăn xếp ra khoi dx
add dl,30h ;chuyển số thành dạng ký tự
mov ah,2 ;in tổng
int 21h
loop hienkq
ra: mov ah,4ch
int 21h
Main endp
End main
```

**Bài 3 : Cho một mảng M gồm 20 phần tử kiểu Word giá trị tùy ý(không phải nhập giá trị các phần tử). Tính tổng giá trị các phần tử có giá trị chia hết cho 7**

```
.model tiny
.stack 100h
.data

st1 db 13,10,'tong cac phan tu chia het cho 7:$'
st2 db 13,10,'$'
m db 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20
a db 0
.code
main proc
mov ax,@data
```

```

mov ds,ax
;in thông báo nhập số thứ 1
lea dx,st1
mov ah,9
int 21h
mov cx,20 ;gán cx=20
lea si,m ;si trỏ đến ngăn nhớ đầu tiên của mảng M
mov a,0 ; khởi tạo a=0
duyet:
mov al,[si] ; đưa các giá trị trong mảng do si trỏ đến vào al
mov bl,7 ;gán bl=7
mov ah,0 ;xoá bit cao
div bl ;chia al cho 7
cmp ah,0 ;so sánh thương với 0
je tong ;nếu bằng thì tính tổng
jmp tiep
tong:
mov al,[si] ; đưa các giá trị trong mảng do si trỏ đến vào al
mov bl,a ; đ ư a số a vào bl
add al,bl ;c ộng al v ớ bl kết quả cất vào al
mov a,al ;chuyển kết quảe\l vào biến a
tiep:
inc si ;tăng chỉ số mảng
inc di ;tăng di

```

```
loop duyet
mov al,a ;chuyển số trở lại thành ghi al
mov bl,10 ;gán bl =10
mov cx,0 ;khởi tạo biến đếm
chia:
mov ah,0 ;xoá bit cao
div bl ;lấy kết quả chia cho 10
mov dl,ah ;chuyển dư vào dl
add dl,30h ;chuyển số sang dạng ký tự
push dx ;dẩy dư vào ngăn xếp
inc cx ;tăng biến đếm
cmp al,0 ;so sánh thương với 0
je inso ;nếu bằng thì in số
jmp chia
inso:
pop dx
mov ah,2
int 21h
loop inso

ra:
mov ah,4ch
int 21h
main endp
```

end main

#### **Bài 4 : Lập chương trình nhập vào 1 số kiểu word in ra màn hình mã nhị phân tương ứng của số đó**

.model tiny

.stack 100h

.data

st1 db 'Nhap so kieu WORD :\$'

st2 db 13,10,'Ma nhi phan tuong ung:\$'

so dw 0

.code

main proc

mov ax,@data

mov ds,ax

lea dx,st1

mov ah,9

int 21h ;In xau st1

nhap:

mov ah,1

int 21h

cmp al,13

je nhiphan

sub al,30h

mov ah,0

mov cx,ax

mov bx,10

mov ax,so

mul bx

add ax,cx

mov so,ax

jmp nhap

nhiphan:

lea dx,st2

mov ah,9

int 21h

mov cx,0

mov bx,2

mov ax,so

chia:

mov dx,0

div bx

add dx,30h

push dx

inc cx

cmp ax,0

je inso

jmp chia

inso:

pop dx



```
mov ah,2
int 21h
loop inso
ra:
mov ah,4ch
int 21h
main endp
end main
```

**Bài 5 : Lập chương trình nhập vào 1 số kiểu word in ra màn hình mã Hexa tương ứng của số đó**

```
.model tiny
.stack 100h
.data
st1 db 'nhap so kieu word:$'
st2 db 13,10,'so do duoi dang hecxa:$'
a dw 0
.code
main proc
;
mov ax,@data
mov ds,ax
;
lea dx,st1
mov ah,9
```

int 21h

nhap:

mov ah,1

int 21h

cmp al,13

je inso

mov ah,0

sub al,30h

mov cx,ax

mov ax,a

mov bx,10

mul bx

add ax,cx

mov a,ax

jmp nhap

inso:

;

lea dx,st2

mov ah,9

int 21h

mov bx,16

mov ax,a

mov cx,0

```
chia:
mov dx,0
div bx
cmp dx,10
jae doi
add dx,30h
jmp cat
doi: add dx,37h
cat:
push dx
inc cx
cmp ax,0
je hien
jmp chia
hien:
pop dx
mov ah,2
int 21h
loop hien
ra:
mov ah,4ch
int 21h
main endp
end main
```

**Bài 6 : Nhập vào 1 mảng 15 ptu kiểu word in ra màn hình mã Hexa tương ứng của số đó.**

```
.model tiny
```

```
.stack 100h
```

```
.data
```

```
tb1 db 'Nhap mang 15 phan tu kieu word: $'
```

```
tb2 db 10,13,'Nhap phan tu : $'
```

```
tb3 db 10,13,'Phan tu co gia tri lon nhat la: $'
```

```
a dw 20 dup(0)
```

```
.code
```

```
main proc
```

```
mov ax,@data
```

```
mov ds,ax
```

```
;-----
```

```
lea dx,tb1
```

```
mov ah,9
```

```
int 21h
```

```
mov cx,15 ;Nhap 15 phan tu kieu word
```

```
xor si,si
```

```
nhapmang:
```

```
mov ah,9
```

```
lea dx,tb2
```

```
int 21h
```

```
push cx
```

nhapso:

mov ah,1

int 21h

cmp al,13

je catkq

sub al,30h

mov cl,al

xor ch,ch

mov bx,10

mov ax,a[si]

mul bx

add ax,cx

mov a[si],ax

jmp nhapso

catkq:

add si,2

pop cx

loop nhapmang

lea dx,tb3

mov ah,9

int 21h

xor si,si

mov ax,a[si]

```
mov cx,15 ; mov cx,14 add si,2
```

```
duyet:
```

```
cmp a[si],ax
```

```
jbe qua
```

```
mov ax,a[si]
```

```
qua:
```

```
add si,2
```

```
loop duyet
```

```
;mov ax,a[0]
```

```
xor cx,cx
```

```
mov bx,10
```

```
chia:
```

```
xor dx,dx
```

```
div bx
```

```
push dx
```

```
inc cx
```

```
cmp ax,0
```

```
jne chia
```

```
mov ah,2
```

```
hienso:
```

```
pop dx
```

```
add dl,30h
```

```
int 21h
```

loop hienso

;-----

mov ah,4ch

int 21h

main endp

end main

.

# Trắc nghiệm Vi Xử lý và Lập trình hợp ngữ

## Bài 1:

1-  Bộ vi xử lý là:

- A- Hệ thống các vi mạch tương tự hoạt động theo chương trình.
- B- Là một vi mạch tương tự hoạt động theo chương trình.
- C- Hệ thống các vi mạch số học hoạt động theo chương trình.
- D- Là một vi mạch số hoạt động theo chương trình.

2-  Một lệnh của bộ vi xử lý là:

- A- Chuỗi các bit 0 và 1 nằm trong vi xử lý.
- B- Chuỗi các bit 0 và 1 nằm trong một ô nhớ.
- C- Chuỗi các bit 0 và 1 cung cấp cho vi xử lý để nó thực hiện được một chức năng số cơ bản.
- D- Chuỗi các bit 0 và 1 bất kỳ.

3-  Chương trình của hệ thống vi xử lý là:

- A- Tập lệnh của nhà sản xuất bộ vi xử lý cung cấp.
- B- Là một tập hợp các lệnh sắp xếp một cách ngẫu nhiên.
- C- Là một tập hợp các lệnh điều khiển vi xử lý hoạt động.
- D- Là tập hợp các lệnh được sắp xếp theo giải thuật của một bài toán hay một công việc nào đó.



4-  Hệ thống vi xử lý bao gồm bao nhiêu khối chức năng chính?

A- 1

B- 2

C- 3

D- 4

5-  Trong hệ thống vi xử lý Bus là:

A- Là nhóm đường tí hiệu có cùng chức năng trong hệ thống vi xử lý.

B- Thiết bị vận chuyển thông tin giữa các khối mạch của hệ thống vi xử lý.

C- Đường truyền thông tin giữa các khối mạch của hệ thống vi xử lý.

D- Là nơi chứa dữ liệu trong hệ thống vi xử lý.

6-  Trong hệ thống vi xử lý Bus điều khiển dùng để:

A- Truyền dữ liệu với vi xử lý

B- Xác định các chế độ hoạt động của hệ thống vi xử lý

C- Lấy địa chỉ vào vi xử lý

D- Lấy lệnh để thực hiện

7-  Bus địa chỉ là:

A- Các ngõ vào của vi xử lý

B- Các ngõ ra của vi xử lý

- C- Các đường hai chiều của bộ vi xử lý
- D- Là các đường trở kháng cao của bộ nhớ.

8-  Bus dữ liệu là:

- A- Các ngõ vào của vi xử lý
- B- Các ngõ ra của vi xử lý
- C- Các đường hai chiều của bộ vi xử lý
- D- Là các đường trở kháng cao của bộ nhớ

9-  Các thiết bị ngoại vi được nối tới bộ vi xử lý:

- A- Thông qua các cổng vào ra
- B- Thông qua bộ nhớ
- C- Thông qua Bus dữ liệu
- D- Thông qua Bus địa chỉ

10-  Bộ nhớ có thể truy cập ngẫu nhiên là:

- A- Bộ nhớ ROM và bộ nhớ RAM
- B- Bộ nhớ RAM
- C- Bộ nhớ ROM
- D- Bộ nhớ băng từ.

## Bài 2:

1-  Khi không được chọn Bus dữ liệu của bộ nhớ ở trạng thái:

A- 0

B- 1

C- High Z (Trở kháng cao)

D- Ngẫu nhiên

2-  Khi các tín hiệu điều khiển đọc và điều khiển ghi bộ nhớ không tích cực thì Bus dữ liệu của bộ nhớ sẽ ở trạng thái:

A- 0

B- 1

C- High Z (Trở kháng cao)

D- Ngẫu nhiên

3-  Bus điều khiển của bộ nhớ là:

A- Ngõ ra của bộ nhớ

B- Ngõ vào của bộ nhớ

C- Là các đường hai chiều của bộ nhớ

D- Có đường là ngõ vào, là ngõ ra, hai chiều

4-  Khi tín hiệu cho phép không tích cực, ngõ ra của cổng đệm 3 trạng thái sẽ:

A- Giống trạng thái ngõ vào trước đó

B- Luôn bằng 0

C- Luôn bằng 1

D- High Z (Trở kháng cao)

5-  Nếu địa chỉ đầu của một vùng nhớ có kích thước là 642KB là 0 thì địa chỉ cuối của vùng nhớ này là: (Mỗi ô nhớ chứa 1 byte)

A- 64200H

B- 00642H

C- A07FFH

D- A700FH

6-  Cần bao nhiêu chip nhớ 8KB để có dung lượng nhớ 1MB:

A- 100

B- 112

C- 128

D- 256

7-  Cần 8 bộ nhớ bao nhiêu KB để có dung lượng nhớ 1 MB:

A- 256

B- 128

C- 64

D- 32

8-  Trong hệ nhị phân một byte bằng:

A- 7 bit

B- 8 bit

C- 9 bit

D- 10 bit

9-  Trong hệ thống nhị phân số âm có:

A- Bit MSB bằng 1

B- Bit LSB bằng 1

C- Bit MSB bằng 0

D- Bit LSB bằng 0

10-  Trong hệ nhị phân số âm sẽ bằng:

A- Đảo bit MSB của số dương tương ứng từ 0 lên 1

B- Là số bù hai của số dương tương ứng

C- Là số bù 1 của số dương tương ứng

D- Là số bù 10 của số dương tương ứng.

### Bài 3:

1-  Trong hệ nhị phân số âm 1 biểu diễn bằng tổ hợp 16 bit sẽ bằng:

A- 1111 1111 1111 1111B

B- 1000 0000 0000 0001B

C- 0000 0000 1000 0001B

D- 0000 0000 1111 1111B

2-  Bao nhiêu bit nhị phân được viết gọn lại thành 1 Hex?

A- 2

B- 3

C- 4

D- 5

3-  Với mã BCD số 80 thập phân có thể biểu diễn bằng ít nhất:

A- 5 bit

B- 6 bit

C- 7 bit

D- 8 bit

4-  Mã ASCII của số 5 là:

A- 05H

B- 15H

C- 25H

D- 35H

5-  Mã ASCII của ESC (Escape) là:

A- 13H

B- 23H

C- 1BH

D- 27H

6-  Thanh ghi DX là một thanh ghi:

A- Đa năng

B- Địa chỉ

C- Dữ liệu

D- Đoạn

7-  Số thanh ghi đoạn của 8086/88 là:

A- 3

B- 4

C- 5

D- 6

8-  Thanh ghi BX có thể sử dụng để:

A- Chứa một dữ liệu

B- Chứa một địa chỉ độ dài

C- Một địa chỉ đoạn

D- Cả A và B đều đúng

9-  Thanh ghi ES sử dụng để truy cập:

A- Vùng nhớ dữ liệu

B- Vùng nhớ mã lệnh

C- Vùng nhớ ngăn xếp

D- Cả A và C cùng đúng

10-  Nhóm các thanh ghi nào sau đây đều có thể sử dụng giữ địa chỉ độ dài khi truy cập bộ nhớ dữ liệu?

A- AX, BX, CX, DX

B- CS, DS, SS, ES

C- BX, BP, DI, SI

D- IP, SP, AH, AL



**Bài 4:**

1-  Thanh ghi nào được mặc định giữ số đếm trong các lệnh lặp:

A- AX

B- BX

C- CX

D- DX

2-  Các thanh ghi nào giữ kết quả trong các lệnh nhân chia 16 bit:

A- AX và BX

B- AX và CX

C- AX và DX

D- AX và DI

3-  Thanh ghi nào giữ địa chỉ đoạn khi CPU 8086/88 truy cập vùng nhớ lệnh?

A- CS

B- DS

C- ES

D- SS

4-  Các thanh ghi nào giữ địa chỉ Offset khi 8086/88 truy cập vùng nhớ gần xếp:

A- BP và SP

B- Chỉ có SP

C- BX và SP

D- Chỉ có BP

5-  CPU 8086/88 có bao nhiêu cờ trạng thái:

A- 6

B- 7

C- 8

D- 9

6-  Cờ nhớ (CF) của CPU 8086/88 được lập lên 1 khi:

A- Kết quả các phép tính bằng 0

B- Kết quả các phép tính tràn khỏi dung lượng chứa của toán hạng đích

C- Kết quả các phép tính khác 0

D- Kết quả phép tính không vượt quá dung lượng chứa của toán hạng đích.

7-  Cờ zero (ZF) của CPU 8086/88 được lập lên 1 khi:

A- Kết quả các phép tính bằng 0

B- Kết quả các phép tính lớn hơn 0

C- Kết quả các phép tính khác 0

D- Kết quả phép tính nhỏ hơn 0

8-  CPU có bao nhiêu chu kì máy truy cập bộ nhớ và vào ra:

A- 1

B- 2

C- 3

D- 4

9-  Bảng vecter ngắt trong hệ thống 8086/88 bắt đầu tại địa chỉ:

A- 00000H

B- FFFFFH

C- 00001H

D- FFFF0H

10-  Bảng vecter ngắt trong hệ thống 8086/88 sử dụng để:

A- Chứa chương trình ngắt

B- Chứa địa chỉ chương trình ngắt

C- Chứa dữ liệu cho chương trình ngắt

D- Chứa dữ liệu cho chương trình ngắt

### Bài 5:

1-  Trước khi thực hiện chương trình ngắt, CPU 8086/88 sẽ thực hiện các công việc:

A- Lưu giá trị trong các thanh ghi đa năng vào đỉnh ngăn xếp

B- Lưu giá trị trong các thanh ghi đoạn vào đỉnh ngăn xếp

C- Lưu giá trị trong các thanh ghi CS, IP và thanh ghi cờ vào đỉnh ngăn xếp

D- Lưu giá trị trong các thanh ghi chỉ số vào đỉnh ngăn xếp

2-  Trong hệ thống 8086/88 sau khi lấy dữ liệu từ ngăn xếp thanh ghi con trỏ ngăn xếp (lệnh POP) sẽ:

A- Tự động tăng

B- Tự động giảm

C- Không thay đổi

D- Xoá về 0

3-  Độ đệm lệnh trong CPU giúp chương trình sẽ thực hiện nhanh hơn do:

A- Quá trình lấy lệnh diễn ra nhanh hơn

B- Quá trình thực hiện lệnh diễn ra nhanh hơn

C- Quá trình lấy lệnh thực hiện đồng thời với quá trình thực hiện lệnh

D- Không mất chu kỳ lấy lệnh từ bộ nhớ

4-  Khi có bộ giải mã lệnh:

A- Mã các lệnh sẽ ngắn hơn

B- Các lệnh sẽ được thực hiện nhanh hơn

C- Mã các lệnh sẽ dài hơn

D- Một số lệnh trong chương trình sẽ không phải thực hiện

5-  Để truy cập bộ nhớ CPU cung cấp địa chỉ gì cho bộ nhớ:

A- Đoạn (Segment)

B- Độ dời (offset)

C- Vật lý

D- Logic

6-  Các thanh ghi đa năng của 8086/88 có:

A- 4 bit

B- 8 bit

C- 16 bit

D- 32 bit

7-  Mã lệnh từ bộ nhớ chương trình bên ngoài, sau khi được CPU đọc vào sẽ được chứa tại bộ phận nào trong CPU:

- A- Thanh ghi PC
- B- Thanh ghi IR
- C- Đơn vị giải mã lệnh và điều khiển
- D- ALU

8-  Nhiệm vụ của CPU là:

- A- Điều hành hoạt động của toàn hệ thống theo ý định của người sử dụng thông qua chương trình điều khiển
- B- Thi hành chương trình theo vòng kín gọi là chu kỳ lệnh
- C- Giao tiếp với các thiết bị xuất nhập
- D- Cả hai câu a và b đều đúng

9-  Mã BCD nén là:

- A- Kết hợp hai số BCD thành 1 byte
- B- Thay 4 bit cao bằng 0H
- C- Số BCD dài 1 byte
- D- Thay 4 bit thấp bằng 0H

10-  Mã bù 2 của 1 số nhị phân được tạo ra bằng cách:

- A- Đảo trạng thái tất cả các bit của số nhị phân
- B- Cộng thêm 1 vào mã bù 1

- 
- C- Cộng thêm 2 vào mã bù 1
- 
- D- Lấy bù 1 trừ đi 1
- 

### Bài 6:

- 1-  Mã bù 1 của 1 số nhị phân được tạo ra bằng cách:
- A- Đảo trạng thái tất cả các bit của số nhị phân
  - 
  - B- Cộng thêm 1 vào mã bù 1
  - 
  - C- Cộng thêm 2 vào mã bù 1
  - 
  - D- Lấy bù 1 trừ đi 1
  -
- 2-  Giao tiếp nối tiếp là phương thức giao tiếp (so sánh với giao tiếp song song):
- A- Truyền từng bit, tốc độ truyền chậm, khoảng cách truyền gần
  - 
  - B- Truyền từng bit, tốc độ truyền chậm, khoảng cách truyền xa
  - 
  - C- Truyền từng bit, tốc độ truyền nhanh, khoảng cách truyền gần
  - 
  - D- Truyền từng bit, tốc độ truyền nhanh, khoảng cách truyền xa
  -
- 3-  Thứ tự các thành phần trong dòng lệnh trong chương trình hợp ngữ từ trái qua sẽ là:

- A- Nhãn, lệnh, các toán hạng, chú thích
- B- Nhãn, các toán hạng, lệnh, chú thích
- C- Nhãn, chú thích, lệnh, các toán hạng
- D- Nhãn, các toán hạng, chú thích, lệnh

4-  Các lệnh nào sau đây là các lệnh chỉ dẫn hợp dịch:

- A- .Model, .Stack, .Data, .Code
- B- equ, segment, public, extrn, include
- C- bd, dd, dw, dt
- D- Tất cả các lệnh trên đều là chỉ thị hợp dịch

5-  Các chuỗi ký tự sau, chuỗi nào có thể làm nhãn đúng trong dòng lệnh hợp ngữ?

- A- ANH mailto:@49N:
- B- @49N:
- C- 1NH:
- D- N KH:

6-  Các khai báo dữ liệu sau, khai báo nào không bị lỗi:

- A- Xon DB 1,2,3,fh
- B- Yes DB 4,7,h,9
- C- Rcl DB 19,7,6,10,3



D- Anh DB 9,3,8,7,0

7-  Cho biết khi viết chương trình hợp ngữ theo khung chương trình sau, chương trình dịch sẽ dịch sẽ dịch ra file chạy dưới dạng đuôi nào?

.Model Small

.Stack 100h

.Data

.Code

;Các định nghĩa cho biến và hằng để tại phần này

Start:

Mov AX, @Data ; khởi tạo DS

Mov DS, AX ; nếu cần phải viết thêm lệnh

Mov ES, AX

;Các lệnh của chương trình chính.

Mov AH, 4CH

INT 21H ; Trở về DOS

;các chương trình con để tại phần này.

End Start ; kết thúc toàn bộ chương trình

A- .exe

B- .com

C- .bat

D- .com hoặc .exe

8-  Trong mã lệnh phần chỉ thị toán hạng (Operand), sử dụng để mã hóa:

A- Loại lệnh

B- Vị trí sẽ thực hiện lệnh

C- Vị trí chứa dữ liệu sử dụng trong lệnh

D- Độ dài dữ liệu sử dụng trong lệnh

9-  Trong chế độ địa chỉ gián tiếp thanh ghi, dữ liệu là:

A- Một số được mã hóa trong lệnh

B- Giá trị trong một thanh ghi

C- Giá trị trong một ô nhớ có địa chỉ mã hóa trong lệnh

D- Giá trị nằm trong một ô nhớ có địa chỉ giữ trong một thanh ghi

10-  Trong chế độ địa chỉ tương đối chỉ số, dữ liệu sử dụng trong lệnh nằm trong một ô nhớ có địa chỉ bằng:

A- Giá trị chứa trong thanh ghi DI hoặc SI

B- Giá trị chứa trong thanh ghi DI hoặc SI cộng với một số độ dời

C- Giá trị chứa trong thanh ghi BX hoặc BP

D- Giá trị chứa trong BX hoặc BP cộng với giá trị chứa trong DI hoặc SI

### Bài 7:

1-  Sau khi thực hiện lệnh `MOV AL,0` sẽ xác định được:

A-  $ZF = 0$

B-  $CF = 0$

C-  $PF = 0$

D- Không cờ nào bị thay đổi.

2-  Sau khi thực hiện các lệnh:

```
MOV AH,05  
MOV AL,03  
XCHG AH, AL
```

A- AH=AL=03

B- AH=AL=05

C- AH=03; AL=05

D- AH=05; AL=03

3-  Cho biết kết quả thực hiện chương trình sau lệnh XLATB:

```
ORG 100h  
LEA BX, dat  
MOV AL, 2  
XLATB  
RET  
dat DB 11h, 22h, 33h, 44h, 55h
```

A- AL = 1BH

B- AL = 33H

C- AL = 27

D- AL = 0

4-  Giả sử có: AX=1000H; BX=2000H; CX=3000H

Sau khi thực hiện được các lệnh:

```
PUSH AX  
PUSH BX  
PUSH CX  
POP AX  
POP BX  
POP CX
```

Sẽ được:

A- AX=1000H; BX=2000H; CX=3000H

B- AX=2000H; BX=3000H; CX=1000H

C- AX=3000H; BX=1000H; CX=2000H

D- AX=3000H; BX=2000H; CX=1000H

5-  Các lệnh truyền dữ liệu với vào ra(IN và OUT) và các lệnh truyền dữ liệu với bộ nhớ:

A- Có tín hiệu điều khiển khác nhau và số bit địa chỉ cung cấp khác nhau.

B- Có tín hiệu điều khiển giống nhau, số bit địa chỉ cung cấp khác nhau

C- Có tín hiệu điều khiển khác nhau và số bit địa chỉ cung cấp giống nhau

D- Có tín hiệu điều khiển giống nhau, số bit địa chỉ cung cấp giống nhau

6-  Giả sử AX=2; BX=3 sau lệnh ADC AX,BX thì:

A- AX=5; BX=3

B- AX=5; BX=0

C- AX=5 hoặc 6; BX=0

D- AX=5 hoặc 6; BX=3

7-  Giả sử có AL=9; AH=7; BL=5

Sau khi thực hiện các lệnh sau AX sẽ có giá trị bằng:

ADD AL,BL

DAA

AND AL,0FH

ADD AX,3020H

ADD AL,AH

AAA

A- 3801H

B- 3031H

C- 0103H

D- 3001H

8-  CPU của họ 80x86 được chia ra thành mấy phần chính:

A- 2

B- 3

C- 4

D- 5

9-  Đơn vị thi hành (EU) và đơn vị giao tiếp BUS (BIU) trong 8086 thực thi 1 lệnh:

A- Song song với nhau

B- EU thực thi xong, BIU mới thực thi

C- BIU thực thi xong, EU mới thực thi

D- Không liên quan gì đến nhau

10-  Đơn vị nào quản lý việc nhập xuất, địa chỉ, lấy lệnh, ghi đọc bộ nhớ:

A- EU

B- CPU

C- BIU

D- CU

Bài 8:

1-  Cờ nhớ được ký hiệu là:

A- CF

B- PF

C- AF

D- ZF

2-  Cờ chặn lẻ được ký hiệu là:

A- CF

B- PF

C- AF

D- ZF

3-  Cờ nhớ phụ được ký hiệu là:

A- CF

B- PF

C- AF

D- ZF

4-  Cờ không được ký hiệu là:

A- CF

B- PF

C- AF

D- ZF

5-  Cờ dấu được ký hiệu là:

A- SF

B- IF

C- DF

D- OF

6-  Cờ báo có ngất được ký hiệu là:

A- SF

B- IF

C- DF

D- OF

7-  Cờ định hướng xâu ký tự được ký hiệu là:

A- SF

B- IF

C- DF

D- OF

8-  Cờ tràn được ký hiệu là:

A- SF

B- IF

C- DF

D- OF

9-  Cờ nào được sử dụng trong gỡ rối chương trình, chạy từng lệnh một:

A- TF

B- IF

C- DF

D- OF

10-  Hãy chỉ ra chế độ địa chỉ thanh ghi:

A- MOV BX, DX



B- MOV CL,10

C- MOV AL, [0243H]

D- MOV AL, [BX]

### Bài 9:

1-  Hãy chỉ ra chế độ địa chỉ tức thì:

A- MOV BX, DX

B- MOV CL,10

C- MOV AL, [0243H]

D- MOV AL, [0243H] B) MOV AL, [BX]

2-  Hãy chỉ ra chế độ địa chỉ trực tiếp:

A- MOV BX, DX

B- MOV CL, 10

C- MOV AL, [0243H]

D- MOV AL, [BX]

3-  Hãy chỉ ra chế độ địa chỉ gián tiếp thanh ghi:

A- MOV BX, DX

B- MOV CL, 10

C- MOV AL, [0243H]

D- MOV AL, [BX]

4-  Hãy chỉ ra chế độ địa chỉ thanh ghi:

A- MOV AX, CX

B- MOV CH, 23

C- MOV AL, [0143]

D- MOV AL, [DX]

5-  Hãy chỉ ra chế độ địa chỉ tức thì:

A- MOV CX, AX

B- MOV CH, 220

C- MOV AL, [1243H]

D- MOV AL, [DX]

6-  Hãy chỉ ra chế độ địa chỉ trực tiếp:

A- MOV AX, DX

B- MOV CL, 10h

C- MOV AH, [2243H]

D- MOV AH, [AX]

7-  Hãy chỉ ra chế độ địa chỉ gián tiếp thanh ghi:

A- MOV DX, AX

B- MOV CL, 220h

C- MOV AL, [0243H]

D- MOV AH, [BX]

8-  Hãy chỉ ra chế độ địa chỉ thanh ghi:

A- MOV BH, DL

B- MOV CH, 10h+22

C- MOV AL, [0243H-5]

D- MOV AL, [BX+2]

9-  Hãy chỉ ra chế độ địa chỉ tức thì:

A- MOV BX, DX+3

B- MOV AH, 10h

C- MOV AL, [043H]

D- MOV AL, [BX-9]

10-  Hãy chỉ ra chế độ địa chỉ trực tiếp:

- A- MOV BH, BL
- B- MOV CH, 102h
- C- MOV AX, [123H]
- D- MOV DL, [BH]

**Bài 10:**

1-  Hãy chỉ ra chế độ địa chỉ gián tiếp thanh ghi:

- A- MOV BX, DX
- B- MOV CL, 21
- C- MOV AL, [11H]
- D- MOV DX, [BX]

2-  Hãy chỉ ra chế độ địa chỉ thanh ghi:

- A- MOV CX, DX
- B- MOV AX, 10
- C- MOV BX, [22]

D- MOV AH, [BX]

3-  Hãy chỉ ra chế độ địa chỉ tức thì:

A- MOV BH, DX

B- MOV CL, 10H

C- MOV AL, [3H]

D- MOV AL, [BX]

4-  Hãy chỉ ra chế độ địa chỉ trực tiếp:

A- MOV BX, DX

B- MOV CL, 10

C- MOV BX, [3H]

D- MOV AL, [BX+DI+2]

5-  Hãy chỉ ra chế độ địa chỉ gián tiếp thanh ghi:

A- MOV AH, BL

B- MOV BX, 0

C- MOV DH, [0]

D- MOV BX, [BX]

6-  Hãy cho biết đâu là chế độ định địa chỉ tương đối cơ sở cho toán hạng nguồn:

- A- MOV CX, [BX]+10
- B- MOV CX, [SI]+10
- C- MOV BX, [BX]+[SI]+10
- D- MOVSB

7-  Hãy cho biết đâu là chế độ định địa chỉ tương đối chỉ số cho toán hạng nguồn:

- A- MOV CX, [BX]+10
- B- MOV CX, [SI]+10
- C- MOV BX, [BX]+[SI]+10
- D- MOVSB

8-  Hãy cho biết đâu là chế độ định địa chỉ tương đối chỉ số cơ sở cho toán hạng nguồn:

- A- MOV CX, [BX]+10
- B- MOV CX, [SI]+10
- C- MOV BX, [BX]+[SI]+10
- D- MOVSB

9-  Hãy cho biết đâu là chế độ định địa chỉ chuỗi:

- A- MOV CX, [BX]+10
- B- MOV CX, [SI]+10

C- MOV BX, [BX]+ [SI]+10

D- MOVSB

10-  Hãy cho biết đâu là chế độ định địa chỉ tương đối cơ sở cho toán hạng nguồn:

A- MOV CX, [BX+10]

B- MOV CX, [SI+10]

C- MOV BX, [DX]+ [SI]+10

D- MOVSB

### Bài 11:

1-  Hãy cho biết đâu là chế độ định địa chỉ tương đối chỉ số cho toán hạng nguồn:

A- MOV CX, 10+[BX]

B- MOV CX, [SI+10]

C- MOV BX, [BX]+[SI]+10

D- MOVSB

2-  Hãy cho biết đâu là chế độ định địa chỉ tương đối chỉ số cơ sở cho toán hạng nguồn:

A- MOV CX, [BX+10]

- B- MOV CX, [SI+10]
- C- MOV BX, [BX+SI]+10
- D- MOVSB

3-  Hãy cho biết đâu là chế độ định địa công:

- A- MOV CX, [BX]+10
- B- MOV CX, [SI]+10
- C- MOV BX, [BX]+[SI]+10
- D- IN AL, 40h

4-  Hãy cho biết đâu là chế độ định địa chỉ tương đối cơ sở cho toán hạng nguồn:

- A- MOV CX, 10+[BX]
- B- MOV CX, 10+[SI]
- C- MOV BX, 10+[DX]+ [SI]
- D- OUT DX, AX

5-  Hãy cho biết đâu là chế độ định địa chỉ tương đối chỉ số cho toán hạng nguồn:

- A- MOV CX,10+[BX]
- B- MOV CX, [DI]+5
- C- MOV BX, [BX]+ [DI]+10



D- OUT DX, BX

6-  Hãy cho biết đâu là chế độ định địa chỉ tương đối chỉ số cơ sở cho toán hạng nguồn:

A- MOV CX, [BX+10]

B- MOV CX, [SI+10]

C- MOV BX, 10+[BX+SI+10]

D- IN DX, AX

7-  Hãy cho biết đâu là chế độ định địa công:

A- MOV CX, [BX]+10

B- MOV CX, [SI]+10

C- MOV BX, [BX]+ [SI]+10

D- N AH, 300h

8-  Hãy cho biết đâu là chế độ định địa chỉ trực tiếp cho toán hạng đích:

A- MOV DS, AX

B- MOV [4320], CX

C- MOV [DI], AX

D- MOV CX, [BX+10]

9-  Hãy cho biết đâu là chế độ định địa chỉ gián tiếp thanh ghi cho toán hạng

đích:

A- MOV DS, AX

B- MOV [4320], CX

C- MOV [DI], AX

D- MOV CX, [BX+10]

10-  Sau khi thực hiện đoạn chương trình:

Mov al, 5

Mov bl, 10

Add bl, al

Sub bl, 1

Giá trị trong thanh ghi BL bằng bao nhiêu:

A- 5h

B- 0Eh

C- 26h

D- 0

## Bài 12:

1-  Sau khi thực hiện đoạn chương trình:

Mov al, 5

Mov bl, 10

Add bl, al

Sub bl, 1

Giá trị trong thanh ghi BL bằng bao nhiêu:

A- 5h

B- 0Eh

C- 26h

D- 0

2-  Sau khi thực hiện đoạn chương trình:

Org 100h

Mov AX,0

Mov BX,0

Mov al, 5

Mov bl, 10h

Add bl, al

xor bl, 55h

Thanh ghi BX có giá trị bằng bao nhiêu:

A- 5h

B- 10h

C- 30h

D- 40h

3-  Sau khi thực hiện đoạn chương trình

Org 100h

Mov AX, 0

Mov BX, 0

Mov al, 5

Mov bl, 10h

Add al, bl

Xor bl, 55h

Thanh ghi BX có giá trị bằng bao nhiêu:

A- 15h

B- 25h

C- 35h

D- 45h

4-  Sau khi thực hiện đoạn chương trình:

```
org 100h  
mov AX, 0  
mov BX, 0  
mov al, 5  
mov bl, 10h  
add al, bl  
xor bl, 55h
```

Thanh ghi AX có giá trị bằng bao nhiêu:

A- 15h

B- 25h

C- 35h

D- 45h

5-  Đoạn chương trình nào sau đây là in một ký tự ra màn hình:

A- ...

```
.data  
S DB "Hello !$\n"  
.code  
Mov AX, @data  
Mov DS, AX  
Mov DX, OFFSET s  
Mov AH, 9  
Int 21h
```

B- ...

```
Mov ah, 1  
Int 21h
```

C- ...

```
Mov ax,
```

Int 21h

...

D- ...

- MOV AH, 2  
MOV DL, 'KY TU'  
INT 21h  
...

6-  Đoạn chương trình nào sau đây là in một sô ký tự ra màn hình:

A- ...

- .data  
S DB "Hello !\$"   
.code  
Mov AX, @data  
Mov DS, AX  
Mov DX, OFFSET s  
Mov AH, 9  
Int 21h

...

B- ...

- MOV ah, 1  
Int 21h

...

C- ...

- MOV ax, 4c00h  
Int 21h

...

D- ...

- MOV AL, 65  
MOV AH, 0Eh  
INT 10h

...

7-  Đoạn chương trình nào sau đây chờ bấm một phím bất kỳ từ bàn phím:

A- ...

- .data  
S DB "Hello !\$"   
.code  
Mov AX, @data  
Mov DS, AX  
Mov DX, OFFSET s

Mov AH, 9  
Int 21h

...

B-

...

MOV ah, 1  
Int 21h

...

C-

...

MOV ax, 4c00h  
Int 21h

...

D-

...

MOV AL, 65  
MOV AH, 0Eh  
INT 10h

...

8-  Đoạn chương trình nào sau đây là trả lại quyền điều khiển cho hệ điều hành (thoát về DOS):

A- ...

.data  
S DB "Hello !\$"  
.code  
Mov AX, @data  
Mov DS, AX  
Mov DX, OFFSET s  
Mov AH, 9  
Mov 21h

...

B-

...

MOV ah, 1  
Int 21h

...

C-

...

MOV ax, 4c00h

Int 21h

...

D-

...

MOV AL, 65  
MOV AH, 0Eh

INT 10h

...

9- ...

```
MOV AL, 65  
MOV AH, 0Eh  
INT 10h
```

...

A- In 5 kí tự '3' ra màn hình

B- Đọc 5 ký tự từ bàn phím vào ô nhớ số 3

C- Đọc đĩa cứng tại sector thứ 3

D- Di chuyển 3 byte từ địa chỉ 5 sang địa chỉ 0eh

10-  Đoạn chương trình sau có mục đích gì:

```
w4k:  
MOV ah, 1  
Int 16h  
Jz w4k  
MOV ah, 0  
Int 16h
```

A- Đợi một phím được bấm, đọc mã phím vào AL

B- In số 1 ra màn hình

C- Tạo vòng lặp 16 lần

D- Thoát về DOS

# Tuyển tập các bài tập Assembly chọn lọc dùng cho hệ kỹ sư Tin

## Dur kiến các bài kiểm tra thực hành lấy điểm giữa kỳ (Từ câu 1 đến câu 12)

1. Tính tổng  $a+b \leq 9$
2. Tính tổng  $a+b \geq 10$
3. Hiệu  $a-b \geq 0$
4. Hiệu  $a-b < 0$
5. In theo thứ tự mã ASCII của 2 ký tự
6. Xác định ký tự chữ số
7. Xác định ký tự chữ hoa
8. Xác định ký tự chữ thường
9. Nhập A->F in ra thập phân
10. Nhập a->f in ra thập phân
11. Kiểm tra số chẵn lẻ
12. In một số ngược
13. Nhập một ký tự. Hãy in mã ASCII của nó ở dạng nhị phân. Ví dụ: Ký tự là: A, Mã ASCII dạng nhị phân là: 0100 0001.
14. Nhập một ký tự. Hãy in mã ASCII của nó ở dạng Hexa. Ví dụ: Ký tự là: A, Mã ASCII dạng Hexa: 41h
15. Cho AX chứa 1 số nguyên tùy ý có kiểu 2B không dấu ở dạng thập phân(SV tự cho). Hãy in số đó theo chiều ngược lại. Ví dụ: AX = 12345. In ngược lại là: 54321
16. Cho AX chứa 1 số nguyên 2B có dấu. Hãy in AX ra ở dạng thập phân. Ví dụ: AX =1234, In ra AX =1234. AX =-1234. In ra: AX =-1234.
17. Cho BX chứa 1 số Hexa 4 chữ số (VD: BX = A3B4h). Hãy in BX ra dạng Hexa.
18. Cho BX chứa 1 số Hexa 4 chữ số (VD: BX = A3B4h). Hãy in BX ra dạng nhị phân, mỗi nhóm 4 bit cách nhau 1 dấu trống.  
Ví dụ: Dạng nhị phân là: 1010 0011 1011 0100
19. Nhập vào 1 số nhị phân tối đa 16 bit. Hãy in số nhị phân đó ở dòng sau với các bit có nghĩa. Ví dụ: Nhập: 10110101, In ra: 1011 0101. Nhập 00110110, In ra: 110110.
20. Hãy tính  $N!$  với  $1 \leq n \leq 8$ . Ví dụ:  $N = 6, N! = 720, N = 8, N! = 40320$ .
21. Tính tổng  $S = 1+2+\dots+N$  với  $1 \leq N \leq 360$ .
22. Nhập 2 số  $1 \leq A, B \leq 32767$ , nguyên. Tìm UCLN của chúng. (Thư viện)
23. Nhập  $1 \leq N \leq 32767$ , nguyên. N là nguyên tố hay hợp số? (Thư viện)



24. Nhập  $1 \leq N \leq 32767$ , nguyên. N có phải là số chính phương không? (Thư viện)
25. Nhập  $1 \leq N \leq 32767$ . N có phải là số hoàn thiện không? (Thư viện)
26. Tìm số Fibonacci thứ N, biết  $F_1 = 1, F_2 = 1. F_n = F_{n-2} + F_{n-1}$  (Thư viện)
27. Cho 2 số  $1 \leq A, B \leq 360$ . Hãy tính tích của  $A*B$  bằng thuật toán Ấn độ. (Thư viện)
28. Nhập một mảng 1 chiều các Word. Hãy in mảng ra trên 1 dòng. Tính tổng mảng và in ra tổng. (Thư viện)
29. Nhập một mảng 1 chiều các Word. Hãy tìm MIN và MAX của mảng (Thư viện)
30. Nhập 1 xâu ký tự, chuyển xâu đó sang xâu khác và in ra xâu thứ hai (Thư viện)
31. Nhập 1 xâu chứa họ và tên 1 người, tách tên vào 1 xâu khác và in ra tên(Thư viện)
32. Nhập 1 xâu ký tự. Xâu đó có đối xứng không? (Thư viện)
33. Nhập 1 xâu ký tự tùy ý. Xâu đó có bao nhiêu từ (Thư viện)

**Bàn phím và màn hình**

34. Xoá màn hình, đặt con trỏ vào giữa màn hình, ấn Insert để thay đổi kích thước con trỏ, ấn ESC để thoát.
35. Sử dụng các phím mũi tên để di chuyển con trỏ. ấn ESC thì thoát.
36. Nhập vào một dòng văn bản, ấn Home: Về đầu dòng, ấn END: Về cuối dòng. ấn ESC thì thoát.
37. Sử dụng chức năng cuộn cửa sổ để tạo 8 dải màu nền phủ đầy màn hình, mỗi dải được tô bằng 1 màu cơ bản.
38. Nhập một ký tự từ bàn phím, hãy in đầy ký tự đó ra màn hình với màu sắc tự chọn bằng phương pháp truy nhập bộ nhớ màn hình ở chế độ văn bản.
39. Hãy vẽ một đường thẳng đứng độ dài 50 Pixel chạy qua chạy lại ngang màn hình, thoát khỏi khi ấn một phím bất kỳ thuộc nhóm phím trạng thái.
40. Hãy vẽ bảng màu gồm 256 hình chữ nhật, mỗi hình chữ nhật được tô bởi 1 màu cơ bản từ màu 0 đến màu 255 trong chế độ 256 màu chuẩn.

**Bụi Gi¶i**

<p><b>Câu 1: Tính tổng 2 số:</b></p> <pre>Code Segment Assume cs: Code Org 100h Start: jmp over         tb1 db 'Nhap a = \$'         tb2 db 10, 13, 'Nhap b = \$'         tb3 db 10, 13, 'Tong 2 so da nhap la \$' over:</pre>	<p><b>Câu 2: Tính hiệu hai số:</b></p> <pre>Code Segment Assume cs: Code Org 100h Start: jmp over         tb1 db 'Nhap a = \$'         tb2 db 10, 13, 'Nhap b = \$'         tb3 db 10, 13, 'Hieu 2 so da nhap la \$' over:</pre>	<p><b>Câu 3: Nhập kt cho ra số Hexa tương ứng:</b></p> <pre>Code Segment Assume cs: Code Org 100h Start: jmp over         tb1 db 10, 13, 'Nhap ki tu kt = \$'         tb2 db 10, 13, 'So thap phan tuong ung la 1\$'</pre>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<pre> Mov ah,9 lea dx,tb1 int 21h  mov ah,1 int 21h mov bl,al  mov ah,9 lea dx,tb2 int 21h  mov ah,1 int 21h add bl,al  mov ah,9 lea dx,tb3 int 21h  sub bl,30h cmp bl,39h jbe thoat sub bl,10 mov ah,2 mov dl,'1' int 21h thoat: mov dl,bl mov ah,2 int 21h  int 20h </pre>	<pre> Mov ah,9 lea dx,tb1 int 21h  mov ah,1 int 21h mov bl,al  mov ah,9 lea dx,tb2 int 21h  mov ah,1 int 21h mov cl,al  mov ah,9 lea dx,tb3 int 21h  cmp bl,cl jae thoat xchg bl,cl  mov ah,2 mov dl,'-' int 21h thoat: sub bl,cl add bl,30h mov ah,2 mov dl,bl int 21h </pre>	<pre> tb3 db 10, 13, 'Hay nhap lai voi ( A &lt;= kt &lt;= F) or (a &lt;= kt &lt;= f)\$ '  over: nhap: Mov ah,9 lea dx,tb1 int 21h  mov ah,1 int 21h mov bl,al cmp bl,'A' jae ss1 jmp loi  ss1: cmp bl,'F' jbe thoat1 cmp bl,'a' jae ss2 jmp loi  ss2: cmp bl,'f' jbe thoat2  loi: Mov ah,9 lea dx,tb3 int 21h jmp nhap  thoat1: sub bl,11h jmp thoat3  thoat2: </pre>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<pre> Code Ends End Start  <b>Câu 4: Kiểm tra tính chẵn lẻ của chữ số</b>  Code Segment Assume cs: Code Org 100h Start: jmp over     tb1 db 10, 13, 'Nhap ki tu so kt = \$'     tb2 db 10, 13, 'Do la so chan \$'     tb3 db 10, 13, 'Do la so le\$'     tb4 db 10, 13, 'Hay nhap lai voi ( 0 &lt;= kt &lt;= 9) \$'     over: nhap: Mov ah,9     lea dx,tb1     int 21h      mov ah,1     int 21h     mov bl,al      cmp bl,30h     jae sosanh     jmp loi  sosanh:     cmp bl,39h     jbe inra </pre>	<pre> int 20h Code Ends End Start  <b>Câu 5: In theo thứ tự bảng mã ASCII:</b>  Code Segment Assume cs: Code Org 100h Start: jmp over     tb1 db 'Nhap ki tu thu 1 : \$'     tb2 db 10, 13, 'Nhap ki tu thu 2 : \$'     tb3 db 10, 13, 'Thu tu bang ma la : \$'     over: Mov ah,9     lea dx,tb1     int 21h      mov ah,1     int 21h     mov bl,al      mov ah,9     lea dx,tb2     int 21h      mov ah,1     int 21h     mov cl,al </pre>	<pre> sub bl,31h thoat3:     Mov ah,9     lea dx,tb2     int 21h     mov ah,2     mov dl,bl     int 21h      int 20h Code Ends End Start  <b>Câu 6: In thương - dư của BL:</b>  Code Segment Assume cs: Code Org 100h Start: jmp over     tb1 db 'Nhap mot so bat ky a = \$'     tb2 db 10, 13, 'Thuong cua BL : \$'     tb3 db 10, 13, 'Du cua BL : \$'     over: Mov bl,19     Mov ah,9     lea dx,tb1     int 21h      mov cl,0     mov ah,1 </pre>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<pre> loi:  mov ah,9       lea dx,tb4       int 21h       jmp nhap  inra: Test bl,1       jne sole       mov ah,9       lea dx,tb2       jmp thoat  sole: mov ah,9       lea dx,tb3  thoat: int 21h       int 20h       Code Ends       End Start </pre>	<pre> mov ah,9 lea dx,tb3 int 21h  cmp bl,cl ja  nhay  lap:  mov ah,2       mov dl,bl       int 21h       mov dl,cl       int 21h       jmp thoat  nhay: xchg bl,cl       jmp lap  thoat: int 20h       Code Ends       End Start </pre>	<pre> int 21h sub al,30h  lap:  sub bl,al       cmp bl,al       inc cl       ja  lap  add bl,30h add cl,30h mov ah,9 lea dx,tb2 int 21h mov ah,2 mov dl,cl int 21h  mov ah,9 lea dx,tb3 int 21h mov ah,2 mov dl,bl int 21h  int 20h Code Ends End Start </pre>
<p><b>Câu 7: In bảng mã ASCII mở rộng:</b></p> <pre> Code Segment Assume cs: Code Org 100h Start: jmp over       tb1 db 'Ma ASCII mo rong tu 128 -&gt; 255 \$'       tb2 db ' \$'       tb3 db 10, 13, '\$' over: </pre>	<p><b>Câu 8: Tính TB cộng 2 số ab:</b></p> <pre> Code Segment Assume cs: Code Org 100h Start: jmp over       x1 db 'Nhap a = \$'       x2 db 10, 13, 'Nhap b = \$'       x3 db 10, 13, 'TBC cua a va b la: \$'       x4 db '.5\$'       x5 db '.0\$' over: </pre>	<p><b>Câu 9: In dãy kí tự</b></p> <pre> Code Segment Assume cs: Code Org 100h Start: jmp over </pre>

<pre> Mov ah,9 lea dx,tb1 int 21h  mov ah,9 lea dx,tb3 int 21h  mov cx,127 mov bl,128 mov bh,0  lap: mov ah,2 mov dl,bl int 21h  mov ah,9 lea dx,tb2 int 21h  inc bl inc bh cmp bh,9 jbe nhay  mov ah,9 lea dx,tb3 int 21h  mov bh,0 nhay: loop lap </pre>	<pre> Mov ah,9 lea dx,x1 int 21h  mov ah,1 int 21h mov bl,al  mov ah,9 lea dx,x2 int 21h  mov ah,1 int 21h add bl,al sub bl,60h  mov cl,bl shr cl,1 add cl,30h  mov ah,9 lea dx,x3 int 21h  mov ah,2 mov dl,cl int 21h  test bl,1 jne sole mov ah,9 lea dx,x5 </pre>	<pre> x1 db 'Nhap 1 day ky tu \$' x2 db 10, 13, '\$' over:  Mov ah,9 lea dx,x1 int 21h  Mov ah,9 lea dx,x2 int 21h  lai: mov ah,8 int 21h mov bl,al cmp bl,27 je thoat  cmp bl,' ' je inra cmp bl,13 je xuongdong  cmp bl,'A' jb lai cmp bl,'Z' ja chuthuong jmp inra  chuthuong: cmp bl,'a' jb lai cmp bl,'z' ja lai </pre>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<pre> int 20h Code Ends End Start  <b>Câu 10: Đổi mã ký tự từ mã ASCII ra nhị phân:</b> Code Segment Assume cs: Code Org 100h Start: jmp over     x1 db 'Nhap 1 ky tu \$'     x2 db 10, 13, 'Ma ASCII o dang nhi phan la \$'     n db ? over:  Mov ah,9 lea dx,x1 int 21h  mov ah,1 int 21h mov bl,al  mov ah,9 lea dx,x2 int 21h  mov cx,8 mov n,0 lap: shl bl,1 inc n </pre>	<pre> int 21h jmp thoat sole: mov ah,9 lea dx,x4 int 21h thoat: int 20h Code Ends End Start  <b>Câu 11: Nhập ký tự in ra mã ASCII dạng Hecxa:</b> Code Segment Assume cs: Code Org 100h Start: jmp over     x1 db 'Nhap 1 ky tu \$'     x2 db 10, 13, 'Ma ASCII o dang Hecxa la \$' over: Mov ah,9 lea dx,x1 int 21h  mov ah,1 int 21h mov bl,al  mov ah,9 lea dx,x2 int 21h  mov ah,9 lea dx,x2 int 21h </pre>	<pre> inra: mov ah,2 mov dl,bl int 21h jmp lai xuongdong: mov ah,9 lea dx,x2 int 21h jmp lai thoat: int 20h Code Ends End Start  <b>Câu 12: In ngược số 12345</b> Code Segment Assume cs: Code Org 100h Start: mov ax,12345 mov bx,10 mov cx,0 lap: xor dx,dx div bx push ax mov ah,2 or dl,30h int 21h pop ax cmp ax,0 ja lap </pre>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<pre> jc inso1 mov ah,2 mov dl,'0' int 21h jmp nay  inso1: mov ah,2 mov dl,'1' int 21h  nay: cmp n,4 jb tiep mov ah,2 mov dl,' ' int 21h mov n,0  tiep: loop lap  int 20h Code Ends End Start <b>Câu 13: In số nguyên</b> Code Segment Assume cs: Code Org 100h Start: mov ax,-12345 ;hay ax=12345 or ax,ax jns duong push ax mov ah,2 </pre>	<pre> cmp bl,'9' jbe inra  mov ah,2 mov dl,'4' int 21h  cmp bl,'a' jae tiep sub bl,10h jmp inra  tiep: sub bl,30h inra: mov ah,2 mov dl,bl int 21h mov ah,2 mov dl,'h' int 21h  int 20h Code Ends End Start  <b>Câu 14: In số Hecxa</b> Code Segment Assume cs: Code Org 100h Start: mov bx,0a3b4h mov cx,4 lap: xor dx,dx quaytiep: </pre>	<pre> int 20h Code Ends End Start  <b>Câu 15: In dạng nhị phân</b> Code Segment Assume cs: Code Org 100h Start: jmp over n db ? over: mov bx,0a3b4h mov cx,16 mov ah,2 mov n,0 lap: shl bx,1 inc n jc inso1 mov dl,'0' jmp inra inso1: mov dl,'1' inra: int 21h cmp n,4 jb tiep mov ah,2 mov dl,' ' int 21h mov n,0 tiep: loop lap  int 20h Code Ends </pre>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<pre> mov dl,'-' int 21h pop ax neg ax duong: mov bx,10 xor cx,cx chiatiep: xor dx,dx div bx push dx inc cx cmp ax,0 ja chiatiep mov ah,2 inra: pop dx or dl,30h int 21h loop inra pop dx cx bx ax  int 20h Code Ends End Start  <b>Câu 16: Tính n!</b> Giai_thua Macro n Local nhan Mov bx,n and bx,000fh mov ax,1 nhan: mul bx dec bx </pre>	<pre> shl bx,1 rcl dl,1 inc dh cmp dh,4 jb quaytiiep  cmp dl,9 ja chucai or dl,30h jmp inra chucai: add dl,37h inra: mov ah,2 int 21h loop lap int 20h Code Ends End Start  <b>Câu 17: Tính tổ hợp chập k của n:</b> Giai_thua Macro n Local nhan Mov bx,n and bx,000fh ; Doi thanh so mov ax,1 nhan: mul bx dec bx cmp bx,1 ja nhan EndM Code segment </pre>	<pre> End Start  <b>Câu 18: Tìm số lớn trong 2 Word:</b> Max Macro w1 ,w2 Local qua Mov ax,w1 cmp ax,w2 jae qua mov ax,w2 qua: endM Code Segment Assume cs: Code Org 100h Start: jmp over x1 db 10, 13, 'So lon la: \$' a dw 2345 b dw 4567 over: Mov ah,9 lea dx,x1 int 21h Max a,b call in_so  int 20h Include proc.asm Code Ends End Start  <b>Câu 21: Tìm ước chung lớn nhất:</b> </pre>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------



<pre> cmp bx,1 ja nhan EndM Code segment Assume Cs:Code Org 100h Start: Jmp over  x1 db 'Nhap n = \$' x2 db 10, 13, 'N! = \$' over: Mov ah,9 lea dx,x1 int 21h  mov ah,1 int 21h  Giai_thua ax push ax  mov ah,9 lea dx,x2 int 21h  pop ax mov bx,10 mov cx,0 chia: xor dx,dx div bx push dx inc cx cmp ax,0 ja chia </pre>	<pre> Assume Cs:Code Org 100h Start: Jmp over  x1 db 'Nhap n = \$' x2 db 10, 13, 'Nhap k = \$' x3 db 10, 13, 'To hop chap k cua n la : \$' over: Mov ah,9 lea dx,x1 int 21h  mov ah,1 int 21h mov cx,ax  Giai_thua ax push ax  mov ah,9 lea dx,x2 int 21h  mov ah,1 int 21h sub cx,ax  Giai_thua ax mov bx,ax  pop ax div bx </pre>	<pre> Include Mylib.mac Code Segment Assume CS : Code Org 100h Start :  @write 'Nhap a = ' Call nhap_so mov bx,ax @xuongdong  @Write 'Nhap b = ' Call nhap_so lap: cmp ax,bx jae tinh xchg ax,bx tinh: sub ax,bx cmp ax,0 je thoat jmp lap thoat: @xuongdong @write 'UCLN cua 2 so la : ' Mov ax,bx Call in_so  int 20h Include Proc.asm Code Ends End Start </pre>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Câu 24: Kiểm tra số chính**

<pre> mov ah,2 inra: pop dx       or dl,30h       int 21h       loop inra        int 20h       Code Ends       End Start  <b>Câu 19: Tính tổng n:</b> Code Segment Assume Cs : Code org 100h Start : jmp over       x1 db 10, 13, 'Tong S = \$' over:       Mov ah,9       lea dx,x1       int 21h        Mov bx,360       mov ax,0  cong:  add ax,bx       dec bx       cmp bx,0       ja cong        ;in so ra        mov bx,10       xor cx,cx </pre>	<pre> push ax        Giai_thua cx ; tinh (n-k)!       mov bx,ax ;kq (n-k)!       pop ax ;kq n! chia k!       div bx ;thuc hien n!  chia k! chia (n-k)!  push ax mov ah,9 lea dx,x3 int 21h  ;In so pop ax mov bx,10 mov cx,0 chia: xor dx,dx       div bx       push dx       inc cx       cmp ax,0       ja chia       mov ah,2 inra: pop dx       or dl,30h       int 21h       loop inra        int 20h       Code Ends       End Start  <b>Câu 20: Kiểm tra số nguyên tố:</b> </pre>	<pre> <b>phuong</b> include mylib.mac Code Segment Assume cs: Code Org 100h Start:       @write 'Nhap so n = '       Call nhap_so       @xuongdong        mov cx,ax       mov bx,1 lap:  mov ax,bx       mul bx       inc bx       cmp ax,cx       jb lap       je cp       @write 'Ko phai so chinh phuong'       jmp thoat cp:  @write 'So da nhap la so chinh phuong' thoat:       int 20h       include proc.asm       Code Ends       End Start  <b>Câu 27: Tính tổng các phần tử lẻ:</b> Include Mylib.mac Code Segment </pre>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<pre> chia:  xor dx,dx         div bx         push dx         inc cx         cmp ax,0         ja chia          mov ah,2 inra:  pop dx         or dx,30h         int 21h         loop inra          int 20h         Code Ends         End Start </pre>	<pre> Include Mylib.mac Code Segment Assume Cs : Code Org 100h Start : jmp over         n dw ? over:         @Write 'Nhap x = '         Call nhap_so          cmp ax,2         jbe ngt          mov n,ax         shr ax,1         mov cx,ax         mov bx,2 </pre>	<pre> Assume Cs : Code Org 100h Start : jmp over         a dw 20 dup(?)         n dw ? over:         @write 'nhap mang n = '         Call nhap_so         mov n,ax         mov cx,ax         xor bx,bx </pre>
<p><b>Câu 22: Kiểm tra số hoàn thiện</b></p> <pre> Include Mylib.mac Code Segment Assume cs : code Org 100h Start : jmp over         n dw ? over:         @Write 'Nhap n = '         Call Nhap_so          mov n,ax         mov bx,2         xor cx,cx lap: </pre>	<pre> chia:         xor dx,dx         mov ax,n         div bx         inc bx         cmp dx,0 ; hay or dx,dx         je hopso         loop chia  ngt:         @xuongdong         @write 'Do la so nguyen to'         jmp thoat  hopso:         @xuongdong         @write 'Do la hop so ' </pre>	<pre> nhap:         @xuongdong         @write 'Nhap 1 phan tu: '         Call nhap_so         mov a[bx],ax         add bx,2         loop nhap          mov cx,n         xor bx,bx         xor ax,ax  lap:         xor dx,dx         mov dx,a[bx]         Test dx,1         je tiep         add ax,a[bx]  tiep:         add bx,2         loop lap </pre>

<pre> xor dx,dx div bx cmp dx,0 jne tiep add cx,ax tiep: inc bx mov ax,n cmp bx,ax jbe lap  cmp cx,n jne khong @xuongdong @write 'So da cho la hoan thien ' jmp thoat khong: @xuongdong @write 'So da cho ko hoan thien' thoat: int 20h Include Proc.asm Code Ends End Start  <b>Câu 25: Tính tích 2 số</b> include mylib.mac code segment assume cs:code org 100h start: </pre>	<pre> thoat: int 20h Include Proc.asm Code ends End Start  <b>Câu 23: Tính số Fibonacci thứ n</b> Include mylib.mac Code Segment Assume cs: Code Org 100h Start: jmp over n dw ? over: @write 'Nhap so Fibonacci thu n = ' Call nhap_so @xuongdong  cmp ax,2 jbe thoat  mov n,ax mov ax,1 mov bx,1 mov cx,2 ;tinh tu n&gt;2 tinh: add bx,ax sub ax,bx neg ax inc cx cmp n,cx je thoat1 </pre>	<pre> @xuongdong @write 'tong la : ' Call in_so  int 20h Include Proc.asm Code Ends End Start  <b>Câu 30: Tìm Min-Mã của mảng:</b> Include Mylib.mac Max Macro w1,w2 local thoat mov ax,w1 cmp ax,w2 ja thoat mov ax,w2  thoat: EndM Min Macro w1,w2 local thoat mov ax,w1 cmp ax,w2 jbe thoat mov ax,w2  thoat: EndM Code Segment Assume Cs : Code Org 100h Start : jmp over </pre>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<pre> @write 'nhap a:' call nhap_so mov bx,ax  @xuongdong @write 'nhap b:' call nhap_so @xuongdong  xor cx,cx lap: test bx,1 je sochan add cx,ax sochan: shl ax,1 shr bx,1 cmp bx,0 ja lap  @write 'tich cua a*b la:'  mov ax,cx call in_so int 20h include proc.asm code ends end start  <b>Câu 28:Sắp xếp các p tử tăng dần:</b> include mylib.mac code segment assume cs: code </pre>	<pre> jmp tinh thoat: mov bx,1 thoat1: @write 'So Fibonacci thu n la ' Mov ax,bx Call in_so  int 20h include proc.asm Code Ends End Start  <b>Câu 26: Tính tổng các phần tử chẵn</b> Include Mylib.mac Code Segment Assume Cs : Code Org 100h Start : jmp over a dw 20 dup(?) n dw ? over: @write 'nhap mang n = ' Call nhap_so mov n,ax mov cx,ax </pre>	<pre> a dw 20 dup(?) n dw ? over: @write 'Nhap mang n = ' Call Nhap_so Mov n,ax Mov cx,ax xor bx,bx  nhap: @xuongdong @write 'Nhap 1 phan tu : ' Call nhap_so mov a[bx],ax add bx,2 loop nhap  mov cx,n xor bx,bx  lap1: Max a[0],a[bx] mov a[0],ax add bx,2 loop lap1  @xuongdong @write 'So lon nhat la : ' Call in_so  mov cx,n xor bx,bx  lap2: Min a[0],a[bx] mov a[0],ax </pre>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<pre> org 100h start: jmp over       ;--Khai bao       n dw ?       a dw 50 dup(?) over:  ;-----       @write 'Nhfp n: '       call nhap_so       mov n,ax       mov cx,n       xor bx,bx       xor dx,dx  fornh_:       @xuongdong       @write 'Nhap phan tu thu '       inc dx       mov ax,dx       call in_so       @write ': '       call nhap_so       mov a[bx],ax       add bx,2       loop fornh_       @xuongdong       mov cx,n       mov dx,n       shl dx,1       sub dx,2 for_:       xor bx,bx lap_:       mov ax,a[bx] </pre>	<pre> xor bx,bx nhap:       @xuongdong       @write 'Nhap 1 phan tu: '       Call nhap_so       mov a[bx],ax       add bx,2       loop nhap        mov cx,n       xor bx,bx       xor ax,ax  lap:       xor dx,dx       mov dx,a[bx]       Test dx,1       jne tiep       add ax,a[bx] tiep:       add bx,2       loop lap        @xuongdong       @write 'tong la : '       Call in_so        int 20h       Include Proc.asm       Code Ends       End Start </pre>	<pre> add bx,2 loop lap2        @xuongdong       @write 'So be nhat la : '       Call in_so        Int 20h       Include Proc.asm       Code Ends       End Start  <b>Câu 33: Nhập chuỗi ký tự chuyển chữ hoa sang chữ khác và in</b>       Include Mylib.mac       Code Segment       Assume CS : Code       Org 100h       Start : Jmp over       x1 db 80 dup(?)       x2 db 80 dup(?) over:       @write 'Nhap xau : '       lea di,x1       xor bx,bx       cld nhap:       mov ah,1       int 21h       cmp al,0dh       je chuyen       stosb       inc bx </pre>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<pre> cmp ax,a[bx+2] jle qua_ xchg ax,a[bx+2] mov a[bx],ax qua_: add bx,2 cmp bx,dx jb lap_ loop for_ @xuongdong @write 'Day sau khi sap: ' mov cx,n xor bx,bx forin_: mov ax,a[bx] call In_so add bx,2 @write ' ' loop forin_ int 20h include proc.asm code ends end start </pre>	<p><b>Câu 29: Nhập xâu kt chuyển chữ thường sang xâu khác và in</b></p> <pre> Include Mylib.mac Code Segment Assume CS : Code Org 100h Start : Jmp over x1 db 80 dup(?) x2 db 80 dup(?) over: @write 'Nhap xau : ' lea di,x1 xor bx,bx cld nhap: mov ah,1 int 21h cmp al,0dh je chuyen stosb inc bx jmp nhap chuyen: mov byte ptr[di],'\$' inc bx lea si,x1 @xuongdong lea di,x2 </pre>	<pre> jmp nhap chuyen: mov byte ptr[di],'\$' inc bx lea si,x1 @xuongdong lea di,x2 mov cx,bx lap: mov al,[si] cmp al,'A' jb nhay cmp al,'Z' ja nhay stosb nhay: inc si loop lap mov byte ptr[di],'\$' lea si,x2 @xuongdong mov ah,9 lea dx,x1 int 21h </pre>
<p><b>Câu 31: Nhập họ tách tên</b></p> <pre> Include Mylib.mac Code Segment Assume CS : Code Org 100h Start : Jmp over hten db 30 dup(?) ten db 10 dup(?) over: </pre>	<pre> @xuongdong lea di,x2 </pre>	<pre> @xuongdong mov ah,9 lea dx,x2 int 21h int 20h Code Ends </pre>

<pre> @write 'Nhap ho &amp; ten: ' lea di,hten cld nhap: mov ah,1 int 21h cmp al,0dh je chuyen stosb jmp nhap chuyen: mov byte ptr[di],'\$  @xuongdong @write 'Ho ten la : ' mov ah,9 lea dx,hten int 21h  dec di std mov al,' '  repe scasb inc cx inc di mov bx,cx repne scasb  inc cx add di,2 sub bx,cx </pre>	<pre> mov cx,bx lap: mov al,[si] cmp al,'a' jb nay cmp al,'z' ja nay stosb nhay: inc si loop lap mov byte ptr[di],'\$' lea si,x2  @xuongdong mov ah,9 lea dx,x1 int 21h  @xuongdong mov ah,9 lea dx,x2 int 21h int 20h Code Ends End Start <b>Câu 32: Kiểm tra tính đối xứng của chuỗi</b> include mylib.mac code segment assume cs:code org 100h start: jmp over xau db 80 dup ('\$') </pre>	<pre> End Start <b>Câu 35: Đếm từ trong chuỗi</b> include mylib.mac code segment assume cs:code org 100h start: jmp over xau db 80 dup ('\$') over: @write 'nhap xau: ' xor cx,cx xor bx,bx lea di,xau cld mov ah,1 nhap: int 21h cmp al,13 je tiep inc cx stosb jmp nhap tiep: jcxz inra dec di std mov al,' ' lap: repe scasb or cx,cx je inra inc cx </pre>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------



<pre> mov cx,bx mov si,di lea di,ten cld rep movsb  mov byte ptr[di],'\$' @xuongdong @write 'Ten la:' lea dx,ten mov ah,9 int 21h  int 20h Code Ends End Start  <b>Câu 34: Kiểm tra chữ hoa:</b> code segment assume cs:code org 100h start: jmp over tb1 db 'nhap mot ki tu:\$' tb2 db 10,13,'la chu hoa\$' tb3 db 10,13,'khong\$' over: mov ah,9 lea dx,tb1 int 21h mov ah,1 int 21h cmp al,'A' jb ko cmp al,'Z' ja ko mov ah,9 lea dx,tb2 int 21h jmp het </pre>	<pre> over: @write 'nhap xau:' @xuongdong lea di,xau xor cx,cx cld mov ah,1 nhap: int 21h cmp al,13 je tiep inc cx stosb jmp nhap tiep: dec di shr cx,1 lea si,xau lap: cmpsb jne kdxung sub di,2 loop lap @xuongdong @write 'Xau doi xung' jmp thoat kdxung: @xuongdong @write 'Xau khong doi xung' thoat: int 20h include proc.asm </pre>	<pre> inc di inc bx repne scasb inc cx inc di jmp lap inra: @xuong_dong @write 'So tu trong xau la:' mov ax,bx call in_so int 20h include proc.asm code ends end start </pre>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<pre>ko: mov ah,9 lea dx,tb3 int 21h het: int 20h code ends end start</pre>	<pre>code ends end start</pre>	
-----------------------------------------------------------------------------	--------------------------------	--

--	--	--

# Bài tập môn Lập trình hợp ngữ - Số 1

## Bài tập:

- ; Nhập mảng 1 chiều có n phần tử (mỗi phần tử có giá trị từ 1 đến 9)
- ; + tìm max, min
- ; + nhập x = kiểm tra x có trong mảng hay không?
- ; + sắp tăng dần

=====

.model small

-----MACRO-----

NhapKyTu Macro dl

push ax

mov ah, 8h

int 21h

mov dl, al

pop ax

EndM

XuongHang Macro

push ax

mov ah, 2h

mov dl, 10

```
int 21h
```

```
pop ax
```

```
EndM
```

```
XuatKyTu Macro d1
```

```
push ax
```

```
mov ah, 2h
```

```
mov dl,d1
```

```
int 21h
```

```
pop ax
```

```
EndM
```

```
XuatChuoii Macro
```

```
mov ah, 9h
```

```
int 21h
```

```
EndM
```

```
Max2so Macro d1,d2
```

```
local KetThuc
```

```
mov al,d1
```

```
cmp al,d2
```

```
ja KetThuc
```

```
mov al,d2
```

```
KetThuc:
```

```
EndM
```

```
Min2so Macro d1,d2
```

```
local KetThuc
```

```
mov  al,d1
cmp  al,d2
jb   KetThuc
mov  al,d2
```

```
KetThuc:
```

```
EndM
```

```
HoanVi      Macro d1,d2
```

```
push  ax
mov   al, d1
mov   ah, d2
mov   d1,ah
mov   d2,al
pop   ax
```

```
EndM
```

```
.stack      100h
```

```
.data
```

```
n      db      ?
a      db      9 dup(0)
max    db      ?
min    db      ?
x      db      ?
tb1    db      'Nhap n=$'
tb2    db      'Nhap phan tu mang:$'
```

```
tb3 db 'Max : $'  
tb4 db 'Min : $'  
tb6 db 'Nhap x=$'  
tb7 db 'x co trong mang$'  
tb8 db 'x khong co trong mang$'  
tb9 db 'Mang tang dan:$'
```

```
.code
```

```
; lay dia chi cua vung du lieu
```

```
mov ax, @data
```

```
mov ds, ax
```

```
lea dx, tb1
```

```
XuatChuoii
```

```
NhapLai:
```

```
    NhapKyTu  n
```

```
    mov  al, n
```

```
    cmp  al, '1'
```

```
    jb   NhapLai
```

```
    cmp  al, '9'
```

```
    ja   NhapLai
```

```
XuatKyTu  n
```

```
XuongHang
```

```
lea  dx, tb2
```

XuatChuoi

call NhapMang

xuonghang

lea dx,tb3

XuatChuoi

call TimMax

XuatKyTu max

xuonghang

lea dx,tb4

Xuatchuoi

XuatKyTu min

xuonghang

lea dx,tb6

Xuatchuoi

NhapKyTu x

XuatKyTu x

call KiemTra

call SapTang



xuonghang

lea dx,tb9

XuatChuo

call xuatmang

; tro ve dos

mov ah, 4ch

int 21h

;-----Thu tuc-----

NhapMang Proc

mov cl, n

mov ch, 0

sub cl,'0'

mov si,0

Lap:

NhapKyTu a[si]

XuatKyTu a[si]

inc si

Loop Lap

ret

NhapMang EndP

;-----max-----

TimMax Proc

mov cl, n

mov ch, 0

```
sub    cl,'0'
dec    cl
mov    si,0
mov    ch,a[si]
mov    max,ch
mov    ch,a[si]
mov    min,ch
mov    ch,0
cmp    cl,0    ; neu nhap vao chi co 1 ky tu
je     thoi
```

Lap1:

```
    inc    si
    Max2so    max,a[si]
    mov    max, al
    Min2so    min, a[si]
    mov    min,al
```

Loop Lap1

thoi:

ret

TimMax endp

;-----

SapTang Proc

```
    xor    cx,cx ; gan cx=0
```

```
    mov    cl, n
```

```
sub cl,'0'
```

```
Ngoai:
```

```
push cx
```

```
xor cx,cx ; gan cx=0
```

```
mov cl, n
```

```
sub cl,'0'
```

```
mov si,0
```

```
dec cx
```

```
Trong:
```

```
mov al,a[si]
```

```
cmp al,a[si+1]
```

```
jb tiep
```

```
HoanVi a[si],a[si+1]
```

```
tiep:
```

```
inc si
```

```
Loop Trong
```

```
pop cx
```

```
loop Ngoai
```

```
ret
```

```
SapTang Endp
```

```
KiemTra Proc
```

```
xor cx,cx
```

```
mov cl,n
```

```
sub cl,'0'
```

```

mov  si,0
mov  al,x
Lap2:
      cmp  al,a[si]
      je   ngung
      inc  si

Loop Lap2
xuonghang
lea  dx,tb8
Xuatchuoi
ret

ngung:
xuonghang
lea  dx,tb7
Xuatchuoi
ret

KiemTra      endp

xuatmang  proc
      mov  cl, n
      mov  ch, 0
      sub  cl,'0'
      mov  si,0
      lap4:
            XuatKyTu  a[si]

```

```
inc si
```

```
Loop Lap4
```

```
ret
```

```
xuatmang endp
```

```
end
```

# Bài tập Lập trình hợp ngữ - Số 2

## Bài tập 1:

; bai tap :

; 2) Nhap 1 gia tri N. Dem so bit 0 va bit 1 trong N. Ket qua

;

;

.model small

NhapKyTu macro d1

mov ah, 8

int 21h

mov d1, al

EndM

XuatKyTu macro d1

mov ah, 2

mov dl,d1

int 21h

EndM

XuatThongBao macro

mov ah, 9

int 21h

endM

XuongHang macro

mov ah, 2

```

        mov     dl, 10
        int     21h
        endM
.stack   100h
.data
        tong   db   ?
        n      db   ?
        tb2    db   'Nhap ky tu N: $'
        tb3    db   'Tong So bit 0:$'
        tb4    db   'Tong so bit 1:$'

.code
; lay dia chi vung du lieu
mov     ax, @data
mov     ds, ax

lea     dx, tb2
XuatThongBao
NhapKyTu  n
XuongHang

; ket qua luu vao ah
mov     al,n
call   TongBitO
add     ah,'0'

```

mov tong,ah

lea dx,tb3

XuatThongBao

XuatKyTu tong

xuonghang

; ket qua luu vao ah

mov al,n

call TongBit1

add ah,'0'

mov tong,ah

lea dx,tb4

XuatThongBao

XuatKyTu tong

XuongHang

;tro ve dos

mov ah,4ch

int 21h

=====PROC=====

TongBitO PRoc

xor ah,ah

xor cx,cx



```
    mov  cx,8
Lap1:
    rol  al,1
    jc   Tiep1
    inc  ah
Tiep1:
    Loop lap1
    ret
TongBitO  EndP
```

```
TongBit1  PRoc
    xor  ah,ah
    xor  cx,cx
    mov  cx,8
Lap2:
    rol  al,1
    jnc  Tiep2
    inc  ah
Tiep2:
    Loop lap2
    ret
TongBit1  EndP
```

End

## Bài tập 2:

; bài tập :

; 3) Nhập N.

; Dem so bit 1 o tri chan ,

; Dem so bit 0 o vi tri le,

; xuất kết quả

```
.model small
```

```
NhapKyTu macro dl
```

```
mov ah, 1
```

```
int 21h
```

```
mov dl, al
```

```
EndM
```

```
XuatKyTu macro dl
```

```
mov ah, 2
```

```
mov dl, dl
```

```
int 21h
```

```
EndM
```

```
XuatThongBao macro
```

```
mov ah, 9
```

```
int 21h
```

```
endM
```

```
XuongHang macro
```

```
mov ah, 2
```

```

        mov     dl, 10
        int     21h
        endM
.stack   100h
.data
        tong   db   ?
        n      db   ?
        tb2    db   'Nhap ky tu N: $'
        tb3    db   'Tong So bit 0 o Vi Tri LE :$'
        tb4    db   'Tong so bit 1 o Vi Tri CHAN :$'
.code
; lay dia chi vung du lieu
mov     ax, @data
mov     ds, ax

lea     dx, tb2
XuatThongBao
NhapKyTu  n
XuongHang

; ket qua luu vao ah
mov     al,n
call   TongBit_O_Le
add     ah,'0'

```

mov tong,ah

lea dx,tb3

XuatThongBao

XuatKyTu tong

xuonghang

; ket qua luu vao ah

mov al,n

call TongBit\_1\_Chan

add ah,'0'

mov tong,ah

lea dx,tb4

XuatThongBao

XuatKyTu tong

XuongHang

;tro ve dos

mov ah,4ch

int 21h

=====PROC=====

TongBit\_O\_Le PRoc

    Jmp thuchien

vt db ?

tam db ?

thuchien:

xor ah,ah

xor cx,cx

mov cx,4

mov vt,2 ; gan vt=0000 0010 ( vi tri dau tien)

mov tam,2 ; gan temp=0000 0010

Lap1:

test al,vt

jnz Tiep1 ; neu la bit 1 thi khong tang ah

inc ah ; neu la bit 0 thi ah++

Tiep1:

push ax

push cx

mov cl,tam

mov vt,cl ;

inc vt

inc vt ; tang vt len 2 lan

mov cl,vt

mov tam,cl ; luu lai vt vao temp de dam bao luon

; tang vt dung 2 lan

mov cl,vt

```
mov  al,1
shl  al,cl  ; dich den vi tri vt
mov  vt,al  ; luu lai vt
```

```
pop  cx
pop  ax
```

```
Loop lap1
```

```
ret
```

```
TongBit_O_Le          EndP
```

```
TongBit_1_Chan       PRoc
```

```
jmp  tieptuc
i    db  ?
temp db  ?
```

```
tieptuc:
```

```
xor  ah,ah
xor  cx,cx
mov  cx,4
mov  i,1  ; gan i=0000 0001 ( vi tri dau tien)
mov  temp,0; gan temp=0000 0000
```

```
Lap2:
```

```
test  al,i
jz    Tiep2 ; neu la bit 0 thi khong tang ah
inc  ah    ; neu la bit 1 thi ah++
```

Tiep2:

push ax

push cx

mov cl,temp

mov i,cl ;

inc i

inc i ; tang i len 2 lan

mov cl,i

mov temp,cl ; luu lai i vao temp de dam bao luon

; tang i dung 2 lan

mov cl,i

mov al,1

shl al,cl ; dich den vi tri i

mov i,al ; luu lai i

pop cx

pop ax

Loop lap2

ret

TongBit\_1\_Chan

EndP

end

# Bài tập Lập trình hợp ngữ - Số 3

## Bài tập:

; nhập  $0 \leq x_1, x_2, x_3 \leq 9$

; tính tổng, hiệu, tích, thương

; tìm max, min;

.model small

.stack 100h

.data

tb1 db 10,'Nhập x1: \$'

tb2 db 10,'Nhập x2: \$'

tb3 db 10,'Nhập x3: \$'

x1 db ?

x2 db ?

x3 db ?

tb4 db 10,'Tổng:  $x_1+x_2+x_3=$  \$'

tb5 db 10,'Hiệu:  $x_1-x_2=$  \$'

tb6 db 10,'Tích:  $x_1*x_2=$  \$'

tb7 db 10,'Thương:  $x_1/x_2=$  \$'

tong db 0

hieu db 0

.code

; lấy địa chỉ đoạn dữ liệu



```
mov ax, @data
```

```
mov ds, ax
```

```
; xuất thông báo 1
```

```
mov dx, offset tb1
```

```
call SubXuat
```

```
;Nhập x1
```

```
call SubNhap
```

```
mov x1,al
```

```
; xuất thông báo 2
```

```
mov dx, offset tb2
```

```
call SubXuat
```

```
;Nhập x2
```

```
call SubNhap
```

```
mov x2,al
```

```
; xuất thông báo 3
```

```
mov dx, offset tb3
```

```
call SubXuat
```

```
;Nhập x3
```

```
call SubNhap
```

```
mov x3,al
```

```

; xuất thông báo4
mov  dx, offset tb4
call  SubXuat

; tính tổng
call  SubTong

mov  dx, offset tb5
call  SubXuat

; tính hiệu
call  SubHieu

;call  SubTich

;call  SubThuong

;call  subMax

;call  SubMin

; tro ve dos
mov  ah, 4ch
int  21h

;-----Xuất thông báo-----
SubXuat  PROC
        pop  bx    ; lưu địa chỉ để quay về
        mov  ah, 9h
        int  21h
        push bx

```

ret

SubXuat ENDP

;-----Nhap Ky tu-----

SubNhap PROC

pop bx ; luu dia chi de quay ve

mov ah, 1h

int 21h

push bx

ret

SubNhap ENDP

;-----Tong x1+x2+x3-----

SubTong PROC

pop bx ; luu dia chi de quay ve

mov al,x1

sub al,'0'

mov tong, al

mov al,x2

sub al,'0'

add al, tong

mov tong, al ; tong = x1+ x2

mov al,x3

sub al,'0'

add al, tong ; tong = x1+x2+x2

cmp al, 14h

jae HienHon20

cmp al, 0ah

jae HienHon10

jmp HienTong

HienHon20:

; luu lai al

mov tong, al

; xuat ky tu 2

mov ah, 2h

mov dl, '2'

int 21h

mov al, tong ; gan tong cho al

sub al, 14h

add al, 30h

mov dl, al

int 21h

jmp ketthuc

HienHon10:

; luu lai al

mov tong, al

; xuat ky tu 2

mov ah, 2h

mov dl, '1'

```
int 21h
mov al,tong ; gan tong cho al
sub al, 0ah
add al, 30h
mov dl, al
int 21h
jmp ketthuc
```

HienTong:

```
add al, 30h
mov ah, 2h
mov dl, al
int 21h
```

ketthuc:

```
push bx
ret
```

SubTong ENDP

;-----Hieu x1-x2-----

SubHieu Proc

```
mov al, x2
sub al, '0'
mov hieu, al
```

```
mov al, x1
sub al, '0'
```

```
sub    al, hieu    ; tinh x1-x2
cmp    al, 0h      ; so sanh voi 0
jl     XuatAm
add    al, 30h
mov    dl, al
mov    ah, 2h
int    21h
jmp    KThuc
```

XuatAm:

; xuat dau -

```
mov    hieu, al    ; luu al vao hieu
```

```
mov    dl, '-'
```

```
mov    ah, 2h
```

```
int    21h
```

; xuat al

```
mov    al, 0ffh
```

```
sub    al, hieu
```

```
add    al, 31h
```

```
mov    dl, al
```

```
mov    ah, 2h
```

```
int    21h
```

KThuc:

```
ret
```

```
SubHieu      Endp
;-----Tich x1*x2-----
;SubTich      Proc
;ret
;SubTich      endp
;-----Thuong x1/x2-----
;SubThuong Proc
;ret
;SubThuong endp
;-----Max -----
;SubMax      PROC
;ret
;SubMax      ENDP
;-----
;SubMin      PROC
;ret
;SubMin      endP
;=====
end
```

# Bài tập lập trình hợp ngữ - Số 4

## Bài tập 1:

- ; Nhập số nhị phân và đưa vào BX.
- ; Chương trình có kiểm tra ký tự nhập phải hợp lệ.
- ; việc nhập kết thúc khi nhấn Enter hay đã đủ 16 Bit.
- ; xuất số nhị phân .

```
.model small
```

```
NhapKyTu macro d1
```

```
    mov     ah, 8
```

```
    int     21h
```

```
    mov     d1, al
```

```
    EndM
```

```
XuatKyTu macro d1
```

```
    mov     ah, 2
```

```
    mov     dl,d1
```

```
    int     21h
```

```
    EndM
```

```
XuatThongBao macro
```

```
    mov     ah, 9
```

```
    int     21h
```

```
    endM
```

```
XuongHang macro
```



```

        mov     ah, 2
        mov     dl, 10
        int     21h

        endM

.stack   100h

.data

        i      db    ?
        tb1    db    'Nhap so nhi phan: $'

.code

; lay dia chi vung du lieu
mov     ax, @data
mov     ds, ax

lea     dx, tb1

XuatThongBao
call Nhapsonhiphan
XuongHang

call XuatSoNhiPhan
xuonghang

;tro ve dos
mov     ah, 4ch

int 21h

;=====

```

NhapSoNhiPhan proc

```
; GIAI THUAT
; b1: xoa bx
; b2: Nhap 1 ky tu
; b3: neu ky tu la enter thi ket thuc
; b4:kiem tra ky tu vua nhap co phai la '0' hay '1' khong
;     neu ko thi nhay toi b2
;     nguoc lai toi b5
; b5: xuat ky tu vua nhap
; b6: doi ky tu thanh 0 hay 1
; b7: dich trai bx
; b8: dua tri da doi ( o buoc 6 ) vao bit LSB cua bx
; b9: toi b2 ( cho toi khi du 16)
; b10: ket thuc
```

```
xor  cx,cx ; gan cx =0
```

```
xor  bx,bx ; gan bx=0
```

```
mov  cx,16
```

NhapLai:

```
    NhapKyTu  i
```

```
    cmp  al,13 ; neu la enter thi ket thuc
```

```
    je   KetThuc
```

```
    cmp  al,'0'
```

```
    jb  NhapLai
```

```
    cmp    al,'1'  
    ja     NhapLai  
    XuatKyTu    i  
    and    al,0fh ; doi '1'/'0' thanh 1/0  
    shl    bx,1   ; dich sang trai bx di mot bit  
    or     bl,al  ; dua bit vua nhap vao al
```

Loop NhapLai

ketThuc:

ret

NhapSoNhiphan endp

XuatSoNhiPhan proc

; GIAI THUAT

; quay trai bx de lay bit cuc trai

; neu cf=1 thi xuat '1'

; nguoc lai xuat '0'

jmp tieptuc

tam db ?

tieptuc:

xor cx,cx ; gan cx =0

mov cx,16

NhapLai1:

rol bx,1 ; quay tri bx de lay bit cuc trai

jc xuat1 ; neu la CF = 1

Xuat0:

mov tam,'0'

jmp xuat

Xuat1:

mov tam,'1'

xuat:

xuatkytu tam ; xuat '0'

Loop NhapLai 1

ketThuc1:

ret

XuatSoNhiphan endp

end

## Bài tập 2:

; bài tập :

; 1). Nhập vào 1 số  $i$  [0..7], và 1 giá trị  $N$  bất kỳ (số, ký tự).

; kiểm tra bit thứ  $i$  trong  $N$  có là bit 1 hay không?

; (trong quá trình nhập :

; Nếu :  $i < 0$  không xuất  $i$  ra màn hình

; Và :  $i > 7$  không xuất  $i$  ra màn hình

; chỉ xuất  $i$  khi thỏa điều kiện)

.model small

NhapKyTu macro dl

mov ah, 8

```

        int          21h
        mov          dl, al
                EndM
XuatKyTu macro dl
        mov          ah, 2
        mov          dl,dl
        int          21h
                EndM
XuatThongBao macro
        mov          ah, 9
        int          21h
                endM
XuongHang macro
        mov          ah, 2
        mov          dl, 10
        int          21h
                endM
.stack      100h
.data
        i          db      ?
        n          db      ?
        tb1       db      'Nhap vi tri bit i: $'
        tb2       db      'Nhap ky tu N: $'
        tb3       db      'Co xuat hien$'

```

```
        tb4    db    'Khong xuat hien $'
```

```
.code
```

```
; lay dia chi vung du lieu
```

```
mov     ax, @data
```

```
mov     ds, ax
```

```
lea     dx, tb1
```

```
XuatThongBao
```

```
; kiem tra i thuoc 0->7 ( bit)
```

```
NhapLai:
```

```
        NhapKyTu    i
```

```
        cmp    al,'0'
```

```
        jb     NhapLai
```

```
        cmp    al,'7'
```

```
        ja     NhapLai
```

```
XuatKyTu    i
```

```
mov     al, i
```

```
sub     al,'0'
```

```
mov     i,al
```

```
; gan al bang=1 tuc la 0000 0001
```

```
; hay co the xem nhu la mask
```

```
mov     al,1
```

```
mov     cl,i
```

```
; dich chuyen den vi tri bit thu i
```

shl al,cl

; luu vao vi tri bit thu i vao i

mov i,al

XuongHang

lea dx, tb2

XuatThongBao

NhapKyTu n

XuatKyTu n

XuongHang

mov ah,0

mov al,n

test al,i

jnz BitLa1

lea dx, tb4

XuatThongBao

jmp kethuc

BitLa1:

lea dx, tb3

XuatThongBao

;tro ve dos

kethuc:

```
mov      ah, 4ch
```

```
int 21h
```

```
end
```

### **Bài tập 3:**

```
;-----baoca82-----
```

```
;viet chuong trinh tinh dien tich va chu vi hinh chu nhat
```

```
; ket qua co dang:
```

```
; Nhap chieu dai (0...255): 5
```

```
; nhap chieu rong (0...255):3
```

```
;
```

```
; dien tich: 15
```

```
; chu vi: 16
```

```
.model    small
```

```
NhapKyTu  macro d1
```

```
    mov     ah, 8
```

```
    int     21h
```

```
    mov     d1, al
```

```
        EndM
```

```
XuatKyTu  macro d1
```

```
    mov     ah, 2
```

```
    mov     dl,d1
```

```
    int     21h
```

```
        EndM
```

```
XuatThongBao  macro
```



```

        mov     ah, 9
        int     21h
        endM
XuongHang macro
        mov     ah, 2
        mov     dl, 10
        int     21h
        endM
.stack     100h
.data
        songuyen    dw    ?      ; gia tri tam de nhap
        tb1         db    'Nhap so Thap Phan (-32768....32767)(Enter ket thuc) :$'
        tb2         db    'So thap phan vua nhap: $'
.code
mov     ax, @data
mov     ds,ax
; xuat thong bao 1
lea     dx, tb1
XuatThongBao
call    NhapSoThapPhan
xuonghang

; xuat thong bao2
lea     dx, tb2

```

```
; xuất số thập phân chưa trong ax
```

```
mov ax, songuyen
```

```
call xuatothapphan
```

```
; trở về dos
```

```
mov ah,4ch
```

```
int 21h
```

```
=====
```

```
NhapSoThapPhan Proc
```

```
; các thanh ghi
```

```
jmp Khaibao
```

```
flag db ?
```

```
tam db ?
```

```
khaibao:
```

```
push bx
```

```
push cx
```

```
push dx
```

```
push si
```

```
mov flag,0 ; flag=0 : ký tự đầu tiên là '+' || '-'
```

```
; flag=1; ký tự đầu tiên là '0'...'9'
```

```
@Begin:
```

```
; bx chứa giá trị nhập
```

```
xor bx,bx
```

```
; cx=1: giá trị nhập <0; cx=0 : giá trị Nhập >0
```

```
xor cx,cx
```

@KyTuDauTien : ; ky tu dau tien phai la dau '-' || '+' || '0'...'9'

batdau:

```
NhapKytu    tam
    cmp      al,13
    je       @Exit
    cmp      al,'+'
    je       LasoDuong
    cmp      al,'-'
    je       LaSoAm
    cmp      al,'0'
    jb       BatDau
    cmp      al,'9'
    ja       BatDau
    mov      flag,1
```

LaSoDuong:

```
    mov     cx,0
    jmp     LuuKyTuVuaNhap
```

LaSoAm:

```
    mov     cx,1
    jmp     LuuKyTuVuaNhap
```

@Exit:

```
    jmp     Exit
```

LuuKytuVuaNhap: ; chi luu khi flag=1

```
    xuatkyTu    tam ; xuat ky tu vua nhap
```

```

mov  al,flag
cmp  al,1
jne  NhapKyTuTiepTheo
@Save:
xor  ax,ax
mov  al,tam
and  al,0fh      ;
mov  si,ax
mov  ax,10
mul  bx      ;dx:ax= bx*10
mov  bx,si
add  bx,ax  ;bx=bx*10 + si

```

NhapKyTuTiepTheo:

```

NhapKyTu  tam
cmp      al,13
je      Exit
cmp      al,'0'
jb      NhapKyTuTiepTheo
cmp      al,'9'
ja      NhapKyTuTiepTheo

```

```

xor  ax,ax  ; gan ax=0
mov  al,tam
and  al,0fh      ;

```

```

cbw

mov  si,ax

mov  ax,10

mul  bx    ;dx:ax= bx*10
mov  bx,si
add  bx,ax ;bx=bx*10 + si

cmp  dx,0  ; khi phep nhan lon hon 16 bit
      ; ( tuc > MAX unsiged int)

jne  Exit

cmp  cx,1  ; la so am
je   @Soam

cmp  bx,32767  ; MAX nhat cua SODUONG
ja   LonHon_Bx
jmp  @xuatkytu

@SoAm:

cmp  bx,32768  ; MIN nhat cua SOAM
ja   LonHon_Bx
jmp  @xuatKyTu

LonHon_Bx:

mov  bx,songuyen

```

```
        jmp  NhapKyTuTiepTheo
```

```
@xuatKyTu:
```

```
        mov  songuyen, bx
```

```
        XuatKytu  tam
```

```
        jmp  NhapKyTuTiepTheo
```

```
Exit:
```

```
cmp  cx,1  ; so am
```

```
je   doidau
```

```
jmp  khongdoidau
```

```
doidau:
```

```
neg  songuyen
```

```
khongdoidau:
```

```
        ; ket qua chua vao songuyen
```

```
pop  si
```

```
pop  dx
```

```
pop  cx
```

```
pop  bx
```

```
ret
```

```
NhapSoThapPhan      Endp
```

```
XuatSoThapPhan      Proc
```

```
        push  bx
```

```
        push  cx
```

```
        push  dx
```

```
        push  si
```

```

; xuat dau '-' neu ax <0
or    ax,ax
jge   @Positive
push  ax
xuatKyTu  '-'
pop   ax
neg   ax
@positive:
    xor    cx,cx ; xoa bien dem
    mov   bx,10 ; so chia
@repeat1:
    xor    dx,dx
    div   bx    ; dx:ax chia cho 10
    push  dx
    inc   cx
    or    ax,ax ; ax=0?
    jnz   @repeat1
; xuat cx ky so da cat
@For:
    pop   dx
    or    dl,'0'
    xuatkytu  dl
loop @for
pop   si

```

```

    pop    dx
    pop    cx
    pop    bx
    ret
xuathapphan          endp
end

```

#### **Bài tập 4:**

; viết chương trình giả lập một máy tính tay thực hiện 4 phép tính

; cộng, trừ, nhân, chia, số nguyên không dấu

; kết quả có dạng:

;     Chương trình giả lập máy tính tay:

; Nhập phép tính +,-,\*,/ ( hoặc Enter): +

; toán hạng 1: 125

; toán hạng 2: 37

; kết quả: 162

; Nhập phép tính +,-,\*,/ ( hoặc Enter): -

; toán hạng 1: 12

; toán hạng 2: 34

; kết quả:    -22 ( trong giao trình là: 22 )

```
.model     small
```

```
NhapKyTu   macro d1
```

```
    mov     ah, 8
```

```
    int     21h
```



```

        mov     d1, al
        EndM
XuatKyTu macro d1
        mov     ah, 2
        mov     dl,d1
        int     21h
        EndM
XuatThongBao macro
        mov     ah, 9
        int     21h
        endM
XuongHang macro
        mov     ah, 2
        mov     dl, 10
        int     21h
        endM
TinhDienTich Macro d1,d2
        mov  ax,d1
        mul  d2
        ENdM
TinhChuVi Macro d1,d2
        mov  ax,d1
        add  ax,d2
        add  ax,ax

```

ENdM

.stack 100h

.data

songuyen dw ?

chieudai dw ?

chieurong dw ?

dientich dw ?

chuvi dw ?

tb1 db 'Chuong trinh gia lap may tinh tay:\$'

tb2 db 'Nhap phep tinh +,-,\*,/ ( hoac Enter): \$'

tb3 db 'toan hang 1: \$'

tb4 db 'toan hang 2: \$'

tb5 db 'ket qua: \$'

.code

mov ax, @data

mov ds,ax

; Nhap chieu dai

lea dx, tb1

XuatThongBao

call NhapSoThapPhan

mov ax, songuyen

mov chieudai, ax

xuonghang

;Nhap chieu rong

lea dx, tb2

XuatThongBao

call NhapSoThapPhan

mov bx, songuyen

mov chieurong, bx

xuonghang

; tinh dien tich = chieudai \* chieurong

; ket qua luu vao ax

TinhDienTich chieudai,chieurong

mov dientich,ax

; tinh chu vi = (chieudai + chieurong) /2

; ket qua luu vao ax

Tinhchuvi chieudai,chieurong

mov chuvi,ax

; xuat dientich

lea dx,tb3

xuatthongbao

mov ax, dientich

call xuatsothapphan

xuonghang

```

; xuat chuvi
lea    dx,tb4
xuatthongbao
mov    ax,chuvi
call   xuatsothapphan
xuonghang
; tro ve dos
mov    ah,4ch
int    21h

;=====

```

```

NhapSoThapPhan      Proc

```

```

    ; cat cac thanh ghi

```

```

    jmp    Khaibao

```

```

        tam    db    ?

```

```

        flag   db    ?

```

```

khaibao:

```

```

    push  bx

```

```

    push  cx

```

```

    push  dx

```

```

    push  si

```

```

    mov   flag,0 ; flag=0 : ky tu dau tien la '+' || '-'

```

```

                ; flag=1; ky tu dau tien la '0'...'9'

```

```

    ; bx chua gia tri nhap

```

```
xor    bx,bx
```

```
@Begin:
```

```
    NhapKyTu    tam
```

```
    cmp         al,13
```

```
    je         Exit
```

```
    cmp         al,'0'
```

```
    jb         @BeGin
```

```
    cmp         al,'9'
```

```
    ja         @begin
```

```
    mov    al,tam
```

```
    and    al,0fh    ;
```

```
    cbw
```

```
    mov    si,ax
```

```
    mov    ax,10
```

```
    mul    bx    ;dx:ax= bx*10
```

```
    mov    bx,si
```

```
    add    bx,ax    ;bx=bx*10 + si
```

```
    cmp    bx,255; MAX nhat cua de bai
```

```
    ja    LonHon_Bx
```

```
    jmp    @xuatKyTu
```

LonHon\_Bx:

mov bx,songuyen

jmp @Begin

@xuatKyTu:

mov songuyen, bx

XuatKytu tam

jmp @Begin

Exit:

pop si

pop dx

pop cx

pop bx

ret

NhapSoThapPhan Endp

XuatSoThapPhan Proc

push bx

push cx

push dx

push si

; xuat dau '-' neu ax <0

or ax,ax

jge @Positive

push ax

xuatKyTu '-'

pop ax

neg ax

@positive:

xor cx,cx ; xoa bien dem

mov bx,10 ; so chia

@repeat1:

xor dx,dx

div bx ; dx:ax chia cho 10

push dx

inc cx

or ax,ax ; ax=0?

jnz @repeat1

; xuat cx ky so da cat

@For:

pop dx

or dl,'0'

xuatkytu dl

loop @for

pop si

pop dx

pop cx

pop bx

ret

xuatsothapphan

endp

end



# Bài tập lập trình hợp ngữ - Số 5

## Bài 1:

- ; Nhập số thập lục( tối đa 4 ký tự) BX.
- ; Chương trình có kiểm tra ký tự nhập phải hợp lệ.
- ; việc nhập kết thúc khi nhấn Enter hay đã đủ 4 ký tự.
- ; xuất số thập lục

```
.model small
```

```
NhapKyTu macro dl
```

```
mov ah, 8
```

```
int 21h
```

```
mov dl, al
```

```
EndM
```

```
XuatKyTu macro dl
```

```
mov ah, 2
```

```
mov dl, dl
```

```
int 21h
```

```
EndM
```

```
XuatThongBao macro
```

```
mov ah, 9
```

```
int 21h
```

```
endM
```

```
XuongHang macro
```

```
mov ah, 2
```

```

        mov     dl, 10
        int     21h

        endM

.stack   100h

.data

        i      db    ?

        tb1    db    'Nhap so thap luc phan: $'

        tb2    db    'So vua nhap:$'

.code

; lay dia chi vung du lieu

mov     ax, @data

mov     ds, ax

lea     dx, tb1

XuatThongBao

call NhapsoThapLuc

XuongHang

call XuatSoThapLuc

xuonghang

;tro ve dos

mov     ah, 4ch

int 21h

;=====

```

NhapSoThapLuc proc

; GIAI THUAT

; b1: xoa bx

; b2: Nhap 1 ky tu

; b3: neu ky tu la enter thi ket thuc

; b4:kiem tra ky tu vua nhap co phai tu '0' hay 'F' khong

; neu ko thi nhay toi b2

; nguoc lai toi b5

; b5: xuat ky tu vua nhap

; b6: doi ky tu thanh HEX tuong ung

; b7: dich trai bx 4 bit

; b8: dua tri da doi ( o buoc 6 ) vao 4 bit thap cua bx

; b9: toi b2 ( cho toi khi du 4)

; b10: ket thuc

xor cx,cx ; gan cx =0

xor bx,bx ; gan bx=0

mov cl,4

mov si, 4

NhapLai:

NhapKyTu i

cmp al,13 ; neu la enter thi ket thuc

je KetThuc

```
cmp al,'0'  
jb NhapLai  
cmp al,'9'  
ja Tiep  
jmp LaSo
```

Tiep:

```
cmp al,'A' ;kiem tra la chu HOA  
jB NhapLai  
cmp al,'F' ;kiem tra la chu HOA  
jA TiepNua  
jmp LaChuHoa
```

TiepNua:

```
cmp al,'a' ;kiem tra la chu THUONG  
jB NhapLai  
cmp al,'f' ;kiem tra la chu THUONG  
jA NhapLai  
jmp LaChuThuong
```

LaSo:

```
SUB al,'0' ;doi sang so tuong ung  
jmp dichBX
```

LaChuHoa:

```
sub al,37h  
jmp dichBX
```

LaChuThuong:

```
sub    al,57h
```

```
jmp    dichBX
```

```
dichBX:
```

```
shl    bx,cl    ; dich sang trai bx di CL bit
```

```
or     bl,al    ; dua bit vua nhap vao al
```

```
dec    si
```

```
XuatKyTu    i
```

```
jnz    NhapLai
```

```
ketThuc:
```

```
ret
```

```
NhapSoThapLuc    endp
```

```
XuatSoThapLuc    proc
```

```
;    GIAI THUAT
```

```
; chep bh vao dl
```

```
; dich phai dl 4 bit
```

```
; neu dl < 10 doi '0'..'9' tuong ung
```

```
; neu khong doi thanh 'A'..'F' tuong ung
```

```
; xuất ký tự trong dl
```

```
; quay trai bx 4 bit
```

```
jmp    tieptuc
```

```
    tam    db    ?
```

tieptuc:

```
xor  cx,cx ; gan cx =0
```

```
mov  cx,4
```

```
mov  si, 4
```

NhapLai 1:

```
mov  dl, bh ; chep bh vao dl
```

```
shr  dl,cl  ; dich phai dl
```

```
cmp  dl, 10 ; dl< 10, doi thanh so
```

```
jb   LaSo2
```

```
add  dl, 37h ; la chu
```

```
jmp  xuất
```

LaSo2:

```
ADD  dl,30h
```

xuat:

```
xuatkytu  dl  ; xuất '0'
```

```
rol  bx, cl
```

```
dec  si
```

```
jnz  NhapLai 1
```

```
ret
```

XuatSoThapLuc endp

End

## **Bài 2:**

; Viet Chuong trinh nhap 1 ky tu va hien thi Ma ASCII theo

; dang nhi phan va thap luc phan.

```
; vi du:
; nhap ky tu : A
; Ma ASCII: 41h
; dang nhi phan: 01000001b
.model      small
```

```
NhapKyTu macro d1
    mov     ah, 8
    int     21h
    mov     d1, al
    EndM
```

```
XuatKyTu macro d1
    mov     ah, 2
    mov     dl,d1
    int     21h
    EndM
```

```
XuatThongBao macro
    mov     ah, 9
    int     21h
    endM
```

```
XuongHang macro
    mov     ah, 2
    mov     dl, 10
    int     21h
```

```

                endM
.stack          100h
.data
    i           db    ?
    tb1         db    'Nhap ky tu  : $'
    tb2         db    'Ma ASCII   : $'
    tb3         db    'Dang Nhi Phan: $'

.code
; lay dia chi vung du lieu
mov            ax, @data
mov            ds, ax

lea           dx, tb1
XuatThongBao
NhapKyTu     i
XuongHang

lea           dx, tb2
XuatThongBao
call XuatSoThapLuc
XuatKyTu     'h'
xuonghang

lea           dx, tb3

```



XuatThongBao

call XuatSoNhiPhan

XuatKyTu 'b'

xuonghang

;tro ve dos

mov ah, 4ch

int 21h

=====

XuatSoNhiPhan proc

; GIAI THUAT

; quay trai bx de lay bit cuc trai

; neu cf=1 thi xuat '1'

; nguoc lai xuat '0'

jmp tieptuc

tam db ?

tieptuc:

xor cx,cx ; gan cx =0

mov cx,8

NhapLai1:

rol bx,1 ; quay tri bx de lay bit cuc trai

jc xuat1 ; neu la CF = 1

XuatO:

mov tam,'0'

```

                jmp  xuất
Xuat1:
                mov  tam,'1'
xuất:
                xuấtkytu   tam   ; xuất '0'
Loop  NhapLai1
ketThuc1:
                ret
XuatSoNhiphan   endp

XuatSoThapLuc   proc
                xor   cx,cx ; gan cx =0
                mov  cx,4
                mov  si, 2
                mov  bh,i
NhapLai2:
                mov  dl, bh ; chep ky tu vua nhap vao dl
                shr  dl,cl  ; dich phai dl
                cmp  dl, 10 ; dl< 10, doi thanh so
                jb   LaSo2
                add  dl, 37h ; la chu
                jmp  xuất2
LaSo2:
                ADD  dl,30h

```

```

        xuất2:
        xuấtkytu    dl    ; xuất '0'
        rol    bh, cl
        dec    si
        jnz    NhapLai2
        ret
XuatSoThapLuc    endp

```

End

### **Bài 3:**

```

;    Viet Chuong trinh nhap 2 so nhi phan 8 bit
;    tinh tong va xuất ra ket qua.
;    vi du:
;    nhap so nhi phan 1(8 bit): 0110
;    nhap so nhi phan 2(8 bit): 111011
;    tong: 1000001
.model    small

```

```

NhapKyTu    macro dl
        mov    ah, 8
        int    21h
        mov    dl, al
        EndM

```

```

XuatKyTu    macro dl
        mov    ah, 2

```

```

        mov     dl,d1
        int     21h

        EndM

XuatThongBao macro
        mov     ah, 9
        int     21h

        endM

XuongHang macro
        mov     ah, 2
        mov     dl, 10
        int     21h

        endM

.stack   100h

.data

b1      db     ?
b2      db     ?

tb1     db     'Nhap So Nhi phan 1  : $'
tb2     db     'Nhap So Nhi Phan 2  : $'
tb3     db     'Tong : $'

.code

; lay dia chi vung du lieu

mov     ax, @data

mov     ds, ax

```

```
lea          dx, tb1
```

```
XuatThongBao
```

```
call NhapSoNhiPhan
```

```
mov  b1,b1      ; luu lai ky tu vua nhap
```

```
XuongHang
```

```
lea          dx, tb2
```

```
XuatThongBao
```

```
call NhapSoNhiPhan
```

```
mov  b2,b1      ; luu lai ky tu vua nhap
```

```
xuonghang
```

```
lea          dx, tb3
```

```
XuatThongBao
```

```
call TongNhiPhan
```

```
xuonghang
```

```
call xuatSonhiPhan
```

```
;tro ve dos
```

```
mov          ah, 4ch
```

```
int 21h
```

```
=====
```

```
NhapSoNhiPhan  proc
```

```
    jmp  tiep
```

```
    tam  db  ?
```

tiếp:

```
xor  bx, bx ; gan bx=0
```

```
xor  cx,cx ; gan cx=0
```

```
mov  cx,8 ; nhap cho toi 8 lan
```

Nhap:

```
    NhapKyTu  tam
```

```
    cmp      al,13
```

```
    je      Ngung
```

```
    cmp      al,'0'
```

```
    jb      Nhap
```

```
    cmp      al,'1'
```

```
    ja      Nhap
```

```
    XuatKyTu  tam
```

```
    mov      ah,tam
```

```
    sub      ah,'0'
```

```
    shl      bl,1 ; dich bl sang trai 1 bit
```

```
    or       bl,ah ; lay bit 0 hay 1
```

Loop Nhap

Ngung:

ret

NhapSoNhiPhan endp

XuatSoNhiPhan Proc

```
jmp khaibao
```

```
    tam2 db    ?
```

```
khaibao:
```

```
xor  cx,cx ; gan cx =0
```

```
mov  cx,8
```

```
Lap2:
```

```
    rol  bl,1
```

```
    jc   Xuat1 ; neu CF=1
```

```
Xuat0:
```

```
    mov  tam2,30h
```

```
    Jmp  xuat
```

```
Xuat1:
```

```
    mov  tam2,31h
```

```
    jmp  xuat
```

```
xuat:
```

```
    XuatKyTu  tam2
```

```
loop lap2
```

```
mov  b2,b1
```

```
ret
```

```
XuatSoNhiPhan Endp
```

```
TongNhiPhan   proc
```

```
    ; giai thuat:
```

```
    ; dich phai 1 bit doi voi b1 va b2
```

```
    ; sau do cong lai
```

```
; + neu:
;
jmp K_bao
    tong db ?
    bitSo1 db ?
    bitSo2 db ?
    du db ? ; bang 1 neu du nguoc lai bang 0
```

K\_bao:

```
xor cx, cx
mov cx, 8
xor bx, bx
mov bl, b1 ; bl la so thu nhat
mov bh, b2 ; bh la so thu hai
xor dl, dl
mov tong, 0
mov du, 0 ; khong co du
```

@Lap:

SoDauTien:

```
ror bl, 1 ; xoay phai 1 ky tu
```

```
jc LaBit_1
```

Labit\_0:

```
mov bitSo1, 0
```

```
jmp SoTiepTheo
```

Labit\_1:



```
mov bitSo1,1
```

SoTiepTheo:

```
ror bh,1 ; xoay phai 1 ky tu
```

```
jc Bit1
```

bit0:

```
mov bitSo2,0
```

```
jmp TinhTong
```

bit1:

```
mov bitSo2,1
```

; tong 2 bit

TinhTong:

```
mov dl, bitSo1
```

```
add dl, bitSo2 ; dl= bitSo1 + BitSo2
```

```
add dl, du ; dl= bitSo1 + BitSo2 + du
```

```
cmp dl, 2
```

```
je TongDu1
```

```
or tong,dl
```

```
shl tong,1
```

```
mov du,0 ; khong con du
```

```
jmp LamTiep
```

TongDu1:

```
mov du,1
```

```
shl tong,1
```

```
LamTiep:
```

```
loop @lap
```

```
mov bl,tong
```

```
ret
```

```
TongNhiPhan endp
```

```
end
```

# Bài tập lập trình hợp ngữ - Nhập mảng và chuỗi các ký tự

## Bài tập nhập mảng:

;Nhập vào n (n từ 0 đến 9)

; Nhập vào 1 mảng n số từ 0 đến 9

; in ra số lớn nhất

;tính tổng

; sắp xếp mảng tăng dần

.model small

max2so macro d1,d2

local thoi

mov al, d1

cmp al, d2

ja thoi

mov al,d2

thoi:

endm

min2so macro d1,d2

local thoi

mov al, d1

```
    cmp    al, d2
    jb     thoi
    mov    al,d2
thoi:
```

endm

hoanvimacro d1,d2

```
    push  ax    ; luu lai ax
    mov   al, d1
    mov   ah, d2
    mov   d1, ah
    mov   d2,al
    pop   ax
```

endm

XuongHang macro

```
    mov   ah,2h
    mov   dl,10
    int   21h
```

endm

NhapKyTu macro

```
    mov   ah, 1h
    int   21h
```

endm

XuatKyTu macro d1

```
        mov  ah, 2h
        mov  dl, d1
        int  21h
endm
.stack  100h
.data
n       db   ?
max     db   ?
min     db   ?
tb      db   'Nhap vao so n: $'
tb1     db   'Nhap a[] :$'
tb2     db   'Max:$'
tb3     db   'Min:$'
arr     db   9 dup('0')
```

```
.code
```

```
;lay dia chi vung du lieu
```

```
mov     ax, @data
```

```
mov     ds, ax
```

```
; xuat thong bao 1
```

```
lea     dx, tb
```

```
call    XuatMess
```

; Nhap vao n

NhapKyTu

mov n, al

; xuat thong bao 2

XuongHang

lea dx, tb1

call XuatMess

;nhap mang

call NhapMang

; xuat thong bao 3

XuongHang

lea dx, tb2

call XuatMess

call TimMax

XuatKyTu max

XuongHang

lea dx, tb3

call XuatMess

XuatKyTu min

XuongHang

;=====

```

;tro ve dos
mov      ah, 4ch
int      21h

XuatMess      proc
    mov      ah, 9
    int      21h
    ret
XuatMess      endp

NhapMang      proc

    mov      cl, n
    mov      ch, 0
    sub      cl,'0'
    mov      si,0

    Lap:
        NhapKyTu
        mov  arr[si],al
        inc  si

    loop Lap
    ret
NhapMang      endp

TimMax      proc

```

```

mov     cl,n
mov     ch,0
sub     cl,'0'
mov     si,0
mov     ch, arr[si]
mov     max,ch
mov     min,ch
mov     ch,0
inc     si
dec     cx
lap1:
        max2so     max,arr[si]
        mov         max, al
        min2so     min,arr[si]
        mov         min,al
        inc     si
        loop lap1
        ret
TimMax     endp

```

```

;=====

```

End

**Bài tập nhập chuỗi:**

```

;Nhap 1 chuoi

```

```

; xuat 1 chuoi

```



```

.model      small
.stack      100h

.data
    tb1     db    'Nhap 1 chuo(dung Nhan): $'
    tb2     db    'Nhap 1 chuo(dung Loop): $'
    tb3     db    'xuat chuo nguoc: $'
    tb4     db    10,'Nhap Mang $'
    tb5     db    10,'Phan tu thu :$'

.code

; ket thuc

; Lay dia chi vung du lieu
mov  ax, @data
mov  ds, ax

; xuat thong bao 1
lea  dx, tb1
call XuatTB
call NhapChuo1

; xuat thong bao 2
lea  dx, tb2
call XuatTB
call NhapChuo2

; xuat tb3
lea  dx, tb3
call xuattb

```

```
call  XuatChuoiNguoc
```

```
; tro ve dos
```

```
mov  ah, 4ch
```

```
int  21h
```

```
=====
```

```
;-----nhap chuoi binh thuong-----
```

```
NhapChuoi1 PRoc
```

```
    ; luu dia chi quay ve
```

```
Nhap1:
```

```
mov  ah, 1h
```

```
int  21h
```

```
cmp  al, 13
```

```
je   thoat1
```

```
jmp  Nhap1
```

```
thoat1:
```

```
call  xuonghang
```

```
ret
```

```
NhapChuoi1 endp
```

```
;-----nhap chuoi luu vao ngan xep-----
```

```
NhapChuoi2 PRoc
```

```
pop  bx    ; luu dia chi quay ve
```

```
xor  cx,cx ; xoa bien dem
```

```
mov ah, 1h
```

```
Nhap2:
```

```
int 21h
```

```
cmp al, 13
```

```
je thoat2
```

```
inc cx
```

```
push ax
```

```
jmp Nhap2
```

```
thoat2:
```

```
push bx
```

```
call xuonghang
```

```
ret
```

```
NhapChuoi2 endp
```

```
;-----xuat chuoi nguoc-----
```

```
XuatChuoiNguoc PRoc
```

```
; luu dia chi quay ve
```

```
pop bx
```

```
cmp cx, 1
```

```
jl thoat
```

```
xuat:
```

```
pop ax
```

```
mov dl, al
```

```
mov ah, 2h
```

```
int 21h
```

```
loop  xuat
thoat:
mov   ah, 2h
mov   dl,10
int   21h
push  bx
ret
```

XuatChuoiNguoc endp

;-----

```
XuatTB    PROC
        mov   ah, 9h
        int   21h
        ret
```

XuatTB Endp

;-----xuat ky tu cuong hang

```
xuonghang  proc
        mov   ah, 2h
        mov   dl,10
        int   21h
        ret
```

xuonghang endp

end