8086 Lab Programs

AAM

```
mov ah,00h
mov al,09h
mov bl,09h
mul bl
aam
```

AAM

```
mov cx,3639h
and cx,0f0fh
mov al,ch
mul cl
aam
add ax,3030h
```

Binary to hexa

```
mov ah,00110010b
mov al,ah
shr al,4
mov bl,al
mov al,ah
shl al,4
shr al,4
mov cl,al
```

Binary to hexa new

```
jmp here
msg1 db 'enter binary number:$'
msg2 db 0dh,0ah,'the number of 1 bits is :$'
res db '1,2,3,4,5,6,7,8,9A,B,C,D,E,F'
hexa db 1dup(0)
lea dx,msg1
mov ah,9
int 21h
xor bx,bx
mov cx,16
mov ah,1
mov al,ah
shr al,4
mov al,ah
shl al,4
shr al,4
```

Create a file using interrupt

```
.model small
.stack 64h
.data
fp db 'c:\masm\sample.txt','$',0
.code
start:mov ax,@data
mov ds,ax
mov ah,3ch
```

mov cx,00h

mov dx,offset fp

int 21h

mov ah,4ch

int 21h

end start

.end

Display a string

mov dl,'a'

mov ah,2h

int 21h

mov ax,4ch

int 21h

Display a string

jmp here

message db 'hello world',13,10,'$'

here:mov dx,offset message

mov ah,9h

int 21h

mov ax,4ch

int 21h

hlt

Signed division

mov al,-10

mov bl,2

```
cbw
idiv bl
```

Echo back character

```
.model small
stack 64h
.data
message db 'hello world',13,10,'$'
.code
mov ax,@data
mov ds,ax
mov dx,offset message
mov ah,9h
int 21h
mov ax,4ch
int 21h
end start
```

Factorial

```
mov ax,01h
mov bx,04h
l1:mul bx
dec bx
jnz l1
```

Fibonacci

```
DATA SEGMENT
```

```
f1 db 00h
f2 db 01h
f3 db ?
msg1 db "The Fibonacci series is", 10, 13, "$"
n db 12
DATA ENDS
code segment
assume cs:code, ds:data
start:  mov ax, @data
        mov ds, ax
        lea dx, msg1
        mov ah, 09h
        int 21h
        mov bl, f1
        CALL DISPNUM
        mov dl, ' '
        mov ah, 02h
        int 21h
        mov bl, f2
        CALL DISPNUM
        mov dl, ' '
        mov ah, 02h
        int 21h
        mov ch, 00h
up1:    cmp ch, n
        jae exit
```

```
        mov al, f1
        add al, f2
        mov f3, al
        mov bl, f3
        CALL DISPNUM
        mov dl, ' '
        mov ah, 02h
        int 21h
        mov al, f2
        mov f1, al
        mov al, f3
        mov f2, al
        inc ch
        jmp up1
exit:   mov ah, 4ch
        int 21h
DISPNUM PROC NEAR
        MOV DL, BL
        AND DL, 0F0H        ; display 1st digit
        MOV CL, 04H
        SHR DL, CL
        CMP DL, 09H
        JBE L2
        ADD DL, 07H
 L2:    ADD DL, 30H
        MOV AH, 02H
        INT 21H
```

```
        MOV DL, BL
        AND DL, 0FH
        CMP DL, 09H      ;display 2nd digit
        JBE L3
        ADD DL, 07H
 L3: ADD DL, 30H
        MOV AH, 02H
        INT 21H
        RET
DISPNUM ENDP
            code ends
end start
```

Hexa to binary

```
mov si,3000h
mov bl,08h
mov ah,0eH
next:shl ah,01h
jc loop1
mov [si],00h
jmp again
loop1:mov [si],01h
again:inc si
dec bl
jnz next
```

Hexa to binary

```
jmp here
x db 0a9h
here: mov cl,x
mov bl,02h
mov si,2001h
again: mov al,cl
div bl
mov cl,al
mov [si],ah
mov ah,00h
inc si
cmp al,00h
jnz again
hlt
```

Largest

```
list db 52h,23h,56h,45h
count equ 04h
largest db 01h dup(?)
lea si,list
mov bl,count
mov al,[si]
xyz:cmp al,[si+1]
jnl abc
mov al,[si+1]
abc:inc si
dec bl
```

```
jnz xyz
lea si,largest
mov [si],al
```

Largest

```
jmp here
x equ 04h
y db 01h,02h,04h,0ah
here: lea si,y
      mov al,[si]
      mov cl,x-1
      inc si
xyz1: cmp al,[si]
      jnc xyz
      mov bl,[si]
      xchg al,bl
xyz: inc si
     dec cl
     jnz xyz1
```

Masm

```
mount c c:\
c:
cd masm
edit filename.asm

.model small
```

```
.stack 64
.data
.code
start:mov ax, @data            mount c c:\
      mov ds,ax                C:
      mov ax,05h               CD MASM
      mov bx,02h               masm filename.asm
      add ax,bx                link filename.obj
      mov si,3000h             debug filename.exe
      mov [si],ax
      mov ah,4ch
      int 21h
      end start
```

Positive and negative

```
jmp start
list dw 2579h,0a500h,0c009h,0159h,0b900h
count equ 05h
start:mov cl,count
mov si,offset list
again:mov ax,[si]
shl ax,01
jc neg
inc bx
jmp next
neg:inc dx
next:add si,02
```

```
dec cl
jnz again
ret
```

Rename a file

```
.model small
.stack 64h
.data
fp db 'c:\masm\sample.txt','$',0
pp db 'c:\masm\vit.txt','$',0
.code
start:mov ax,@data
mov ds,ax
mov es,ax
mov ah,56h
mov cx,00h
mov dx,offset fp
mov di,offset pp
int 21h
mov ah,4ch
int 21h
end start
.end
```

Search an element

```
jmp here
arr db 02h,04h,07h,09h,01h
```

```
here:mov cx,05h
mov di,00h
mov si, offset arr
again:mov al,[si]
cmp al,03h
jne next
inc dl
next:inc si
loop again
hlt
```

Length of the string

```
jmp here
str1 db 'microprocessor$'
length db ?
here:mov al,'m'
mov cx,00h
mov si,offset str1
back:cmp al,[si]
je go
inc cl
inc si
jmp back
go:mov length,cl
hlt
```

Reverse a string

```
jmp here
str1 db 'microprocessor$'
count equ 14h
str2 db dup(0)
here: mov cl,count
mov si,offset str1
mov di,offset str1
add di,13
back: mov al,[si]
xchg [di],al
mov [si],al
inc si
dec di
dec cl
jnz back
hlt
```

Sum of n numbers

```
    mov bh,01h
    mov ah,00h

    mov dh,0ah
    xyz:add ah,bh
      cmp ah,dh
      jnz xyz

      hlt
```

Display a time

```
.model small
.code
mov ah,2ch
int 21h
mov al,ch
call disp
mov dl, ':'
mov ah,2
int 21h
mov al,cl
call disp
mov dl, ':'
mov ah,2
int 21h
mov al,dh
call disp
mov ah,4ch
int 21h
disp proc near
aam
add ax,3030h
mov bx,ax
mov dl,ah
mov ah,02h
int 21h
mov dl,bl
```

```
int 21h
ret
disp endp
end
```

String

```
.model small
.stack 64
.code
main proc
   mov ah,1
   int 21h
   mov bl,al
   sub bl,20h
   mov ah,2
   mov dl,bl
   int 21h
   exit:
   mov ah,4ch
   int 21h
   main endp
end main
```