***LAB EXAM 01***

.model small

.stack 100h ;256

.code

main proc

mov ah, 1

int 21h

mov bl,al

mov ah,2

mov dl,10

int 21h

mov dl,13

int 21h

mov ah, 2

mov dl,bl

int 21h

exit:

mov ah,4ch

int 21h

main endp

end main

**LAB EXAM 02**

.model small

.stack 100h

.data

a db 5

b db ?

.code

main proc

mov ax,@data

mov ds,ax

mov ah,1

int 21h

mov b,al

mov ah,2

mov dl,b

int 21h

mov ah,2

mov dl,a

add dl,48

int 21h

exit:

mov ah,4ch

main endp

end main

**LAB EXAM 03**

.model small

.stack 100h

.data

a db "Bangladesh is a smaall country $"

.code

main proc

mov ax,@data

mov ds,ax

mov ah,9

lea dx,a

int 21h

exit:

mov ah,4ch

int 21h

main endp

end main

**Lab Exam 04**

.model small

.stack 100h

.data

a db "Enter a number : $ "

b db "The result is : $ "

.code

main proc

mov ax,@data

mov ds,ax

mov ah,9

lea dx,a

int 21h

mov ah,1

int 21h

mov bl,al

mov ah,2

int 21h

mov dl,10

int 21h

mov dl,13

int 21h

mov ah,9

lea dx,b

int 21h

mov ah,2

mov dl,bl

int 21h

exit:

mov ah,4ch

int 21h

main endp

end main

**ADD NUMBER**

.model small

.stack 100h

.code

main proc

mov ah,1

int 21h

mov bl,al

mov ah,1

int 21h

mov bh,al

add bl,bh

mov ah,2

mov dl,bl

; sub dl,48

int 21h

exit:

mov ah,4ch

int 21h

main endp

end main

**UPPERCASE TO LOWERCSE CONVERTION**

.model small

.stack 100h

.code

main proc

mov ah,1

int 21h

mov bl,al

mov ah,2

mov dl,bl

add dl,32

int 21h

exit:

mov ah,4ch

int 21h

main endp

end main

**CONDITIONAL CONCEPT**

.model small

.stack 100h

.data

a db "The number is gater than b $ "

b db "The number is les than a $ "

main proc

mov ax,@data

mov ds,ax

mov ah,1

int 21h

mov bl,al

mov ah,1

int 21h

mov bh,al

cmp bl,bh

jge L1

jmp L2

mov ah,9

lea dx,b

int 21h

L1:

mov ah,2

mov dl,bl

int 21h

jmp exit:

mov ah,9

lea dx,a

int 21h

L2:

mov ah,2

mov dl,bh

int 21h

jmp exit:

exit:

mov ah,4ch

int 21h

main endp

end main

**Negative and positive number**

.model small

.stack 100h

.data

a db "Negative $ "

b db "Positive $ "

main proc

mov ax,@data

mov ds,ax

mov bl,-10

cmp bl,0

jge L1

jmp L2

L1:

mov ah,9

lea dx,b

int 21h

jmp exit:

L2:

mov ah,9

lea dx,a

int 21h

jmp exit:

exit:

mov ah,4ch

int 21h

main endp

end main

***even odd code***

.model small

.stack 100h

.data

a db "The number is even $ "

b db "The number is odd $ "

c db ?

.code

main proc

mov ax,@data

mov ds,ax

mov ah,1

int 21h

int 21h

cwd

mov bx,2

div bx

cmp dx,0h

je then

jne else

then:

mov ah,9

lea dx,a

int 21h

jmp exit:

else:

mov ah,9

lea dx,b

int 21h

jmp exit:

exit:

mov ah,4ch

int 21h

main endp

end main

**LOOPING CONCEPT**

.model small

.stack 100h

.data

a db ?

main proc

mov ax,@data

mov ds,ax

mov cl,'1'

mov bh,'3'

any:

cmp cl,bh

jg exit:

mov ah,1

int 21h

mov a,al

inc cl

jmp any:

mov ah,2

mov dl,a

int 21h

exit:

mov ah,4ch

int 21h

main endp

end main