**QUESTION 1**

Take a number as input (0-150) from the user. Write an interrupt service routine (ISR) and hook it with interrupt number (input number), in which you will display the number (input number) on screen.

Program should return to main program and continue executing instructions after executing that ISR.

For example:

* User enters the digit 16 in decimal then press enter your program should run the interrupt 0x10.
* User enters the digit 22 in decimal then press enter your program should run the interrupt 0x16.

**Solution:**

;calling the display interrupt by taking the number 16 in decimal from user

;16d==10h

mov cx,3

input:

mov ah,0

int 0x16 ;taking interrupt number from user

cmp al,0x0D

je logic ;

mov dl,al ;

sub dl,0x30 ; converting the input into the decimal value.

mov ax,0 ;

mov al,number ;

mov bl,10 ;

mul bl ;

add al,dl ;

mov number,al ; input number is stored in number variable.

loop input

logic: ; interrupt hooking

; interrupt defined by user is hooked to our service

mov ax,0

mov es,ax

mov ax,4

mov bl,number

mul bl ;bl=bl\*4 for offset

mov bx,ax

lea ax,isr0

mov es:[bx],ax

add bx,2

mov es:[bx],cs

cmp flag ,1 ; controlling the flow of program to execute only once.

je return ; to continue the program after calling the interrupt.

mov flag,1 ;

mov bl,number ; moving the input number in bl.

mov ax,4 ;

mul bl ; multiply the interrupt by 4.

mov bx,ax ;

pushf ; push flags

push cs ; push code segment

mov ax,offset return

push ax ; push IP value of next instruction

jmp far es:[bx] ; calling the interrupt given by the user.

return: ;continue the code after callingthe interrupt.

mov bx,0xffff

.exit

isr0:

mov ax,0xb800

mov es,ax

mov al,number

mov bl,10

mov cx,0

L1: ;getting the digits from the variable

mov ah,0

div bl

mov dl,ah

add dl,0x30

mov dh,0x75

push dx ;push digit in stack to display in correct order (LIFO)

inc cx

cmp al,0

je L2

jmp L1

L2:

pop dx ;pop digits in sequence to display on screen.

mov es:[di],dx

add di,2

loop L2:

iret ; it pops the IP,CS and flags.

number db 0

flag db 0