*8b. Design and develop an assembly program to read the status of two 8-bit inputs (X & Y) from the Logic Controller Interface and display X*Y.*

```
.model small

initds macro
    mov ax,@data       ; initializing the data segment
    mov ds,ax          ; it is ds, not dx
endm

init8255 macro
    mov al,cw       ; initialization of 8255 using control word
    mov dx,cr         by passing 82h to control reg.
    out dx,al         (to make port A as output & port B as input)
endm

inpb macro
    mov dx,pb         ; initialization of port B as input
    in al,dx
endm

outpa macro
    mov dx,pa       ; initialization of port A as output
    out dx,al
endm

printf macro msg
    lea dx,msg         ; load the effective address to dx
    mov ah,9           ; function number is 9
    int 21h            ; using dos interrupt 21h
endm

getchar macro
    mov ah,1           ; this macro takes 1 key input,
    int 21h            ; its ascii value in hex stores in al
endm

exit macro
    mov ah,4ch       ; to terminate
    int 21h
endm
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
.data
    askx db 10,13,"set value for x,then press any key $"
    asky db 10,13,"set value for y,then press any key $"

    pa equ 1190h     ; setting the port address for port a
    pb equ 1191h     ; setting the port address for port b
    cr equ 1193h     ; setting the port address for control reg
    cw db 82h     ; control word is 82 (PORT A is O/P, PORT B is I/P)

.code
    initds
    init8255

    printf askx             ; ask x
    getchar                 ; press any key
```

```
    inpb                    ; reads 1st value i.e. x, which is set
                              through logic controller module, value
                                will be automatically stored in al
    mov bl,al               ; contents of al is copied to bl

    printf asky             ; ask y
    getchar                 ; press any key
    inpb                    ; reads 1st value i.e. x, which is set
                              through logic controller module, value
                                will be automatically stored in al

    mul bl   ; the contents of al is multiplied with contents of bl
                    Result is stored in AX
    outpa
    call delay

    mov al,ah

    outpa
    call delay

    exit

 delay proc
    mov bx,0ffffh       ; do a waste job waste number of times!!!!
    outerfor:
        mov cx,0ffffh

    innerfor:
        loop innerfor
      dec bx
      jnz outerfor
      ret

delay endp


end
```

the result of multiplication is stored in AX reg i.e. AL will be having first 8 bits result, AH – next 8 bits. AL is displayed first on the output module, after some delay, rest 8 bits which are in AH is copied to AL and then displayed on the module.

```
for (bx = bignumber; bx >= 0; bx --)
{
     for(cx = bignumber; cx >= 0; cx --
)
     {
          Do nothing;
     }
}
```

basically, keep decrementing a huge number till zero huge number of times.

By the time, microprocessor does this huge decrements, you can actually see your front end output