8. a. Design and develop an assembly program to demonstrate BCD Up-Down Counter (00-99) on the Logic Controller Interface.

```
.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)
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
    pa equ 1190h    ; setting the port address for port A
    cr equ 1193h    ; setting the port address for control reg
    cw db 82h       ; control word is 82 (PORT A is O/P)

    select db 10,13,"select 1: up counter 2: down counter $"

    exitmsg db 10,13,"press any key to exit $"

.code
    initds          ; initialize data segment
    init8255        ; initialize 8255

    printf select   ; print the choice
    getchar         ; input the choice to AL          ; or cmp al,31h

    cmp al,'1'      ; if your input is 1, go to upcounter
    je upcounter
                                                      ; or cmp al,32h
    cmp al,'2'      ; if your input is 2, go to downcounter
```

```
    je downcounter

    exit                    ; well, upon any other input, just exit.

    upcounter:
        mov al,0            ; initial value of up counter is 0
        up:
            outpa          ; display the contents of al on the interface
            call delay     ; have some delay (let the user see the o/p)
            call keyboardhit    ; if you press any key, then exit.
            add al,1       ; increment the count
            daa            ; daa-decimal adjust after addition
            cmp al,99h     ;compares with 99 in order to count till 99
            jne up         ;upon adding 1, if not equal to 99, go to up
            exit           ; if it crosses 99, exit.

    downcounter:
        mov al,99h         ; initial value of down counter is 99
        down:
            outpa          ; down counter starts
            call delay     ; have some delay (let the user see the o/p)
            call keyboardhit    ; if you press any key, then exit.
            sub al,1       ; decrement the count
            das            ; daa-decimal adjust after subtraction
            cmp al,0       ;compares with 0 in order to count till 0
            jne down       ;upon subtracting 1,if not equal to 0,go to down
            exit           ; if it crosses 0, exit.

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

    innerfor:
        loop innerfor
      dec bx
      jnz outerfor
      ret

delay endp

keyboardhit proc
    push ax         ;save your precious ax value
    mov ah,1        ;checks if any key is pressed in between the count
    int 16h         ;if you press any key, it becomes non-zero. so go
    jnz done            to done and exit.

    pop ax          ;if you don't press any key, it becomes zero. so
                       take out your precious value and return.
    ret
```

```
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
```

```
done:
     exit           ;so you have pressed a key, go to exit.

keyboardhit endp
 end
```