**Experiment-3: Implementation of External Device in emu8086.**

**Task-1: Comparing number in emulator screen.**

org 100h

.DATA

msg1 db 0Dh, 0Ah, "enter a number or any other character to exit: $"

equal\_5 db " is five! (equal) $" ,0Dh,0Ah

below\_5 db " is below five! $" , 0Dh,0Ah

above\_5 db " is above five!$" , 0Dh,0Ah

.CODE

MOV AX, @DATA

MOV DS,AX

game: lea dx, msg1

mov ah, 9

int 21h

; read character in al:

mov ah, 1

int 21h

cmp al, '0'

jb stop

cmp al, '9'

ja stop

cmp al, '5'

jb below

ja above

mov dx, offset equal\_5

jmp print

below: mov dx, offset below\_5

jmp print

above: mov dx, offset above\_5

print: mov ah, 9

int 21h

jmp game ; loop.

stop: ret ; stop

**Task-2: Monitoring numbers in LED display.**

org 100h

#start=led\_display.exe#

#make\_bin#

MOV AX,0H

COUNT:

OUT 199,AX

INC AX

CMP AX,64H

JLE COUNT

HLT

Ret

**Task-3: Checking a palindrome number.**

; this sample checks if string is a palindrome or not.

; palindrome is a text that can be read backwards

; and give the same meaning as if it was read forward.

name "pali"

org 100h

jmp start

m1:

s db '123321'

s\_size = $ - m1

db 0Dh,0Ah,'$'

start:

; first let's print it:

mov ah, 9

mov dx, offset s

int 21h

lea di, s

mov si, di

add si, s\_size

dec si ; point to last char!

mov cx, s\_size

cmp cx, 1

je is\_palindrome ; single char is always palindrome!

shr cx, 1 ; divide by 2!

next\_char:

mov al, [di]

mov bl, [si]

cmp al, bl

jne not\_palindrome

inc di

dec si

loop next\_char

is\_palindrome:

; the string is "palindrome!"

mov ah, 9

mov dx, offset msg1

int 21h

jmp stop

not\_palindrome:

; the string is "not palindrome!"

mov ah, 9

mov dx, offset msg2

int 21h

stop:

; wait for any key press:

mov ah, 0

int 16h

ret

msg1 db " this is palindrome!$"

msg2 db " this is not a palindrome!$"