11b. Generate a Half Rectified Sine waveform using the DAC interface. (The output of the DAC is to be displayed on the CRO).

```
.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

exit macro
    mov ah,4ch          ; to terminate
    int 21h
endm
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

.data
    pa equ 1190h    ;One is Enough-setting the port address for port A
    cr equ 1193h
    cw db 82h       ; 82h is the value in control word 10000010, which
                      makes port A as output port

    table db 80H,96H,0ABH,0C0H,0D2H,0E2H,0EEH,0F8H,0FEH,0FFH;+ve 1st half
          db 0FEH,0F8H,0EEH,0E2H,0D2H,0C0H,0ABH,96H,80H ;+ve 2nd half
          db 80H,80H,80H,80H,80H,80H,80H,80H,80H   ;all zeros (T-OFF)
          db 80H,80H,80H,80H,80H,80H,80H,80H,80H   ;all zeros (T-OFF)

    anykeytoexit db 10,13,"PRESS ANY KEY TO EXIT $"

.code

    initds
    init8255
    printf anykeytoexit          ;or you can use 25h

    start:
        mov cx,37       ;count value is taken 37 bcz the table
                                contains 37  values
        lea si,table    ; table address is loaded to si
```

This is the only change in this pgm

Look at the conversion table at the end of this program. Then you will understand these

```
    back:
        mov al,[si]     ;the contents of si is moved to al i.e. single
                                value of table is moved
        outpa           ; moved value is sent to hardware module
                                through port a
        call delay
        inc si          ; si is pointed to the next value of table
        loop back       ; loop repeats until all the contents of table
                                is moved (till cx becomes 0)

        mov ah,1
        int 16h         ; checks if any key is pressed in keyboard. if
        jz start                 you haven't, then go to start
        exit            ; if you press any key, just call exit macro


 delay proc
    mov bx,0fffh        ; note: single loop delay is enough
    inner:
        dec bx
        jnz inner       ; you can't use CX as it is used to hold the count
                                (37) in our above program

        ret
delay endp

end
```