You are required to implement a music equalizer displayed on the text screen. There is a buffer located at physical address [C800:0001], of 10 bytes length. This buffer holds amplitudes of 10 frequencies where amplitudes ranges between 0 and 255. This buffer is continuously modified by another program listed below. The equalizer simply displays the values on the bottom of screen as 10 bars where the height of each represent the amplitude value as shown in figure 1. The maximum amplitude is shown as a bar of 10 characters height. To make it really fun, hook this to the timer interrupt with chaining so that it keeps working even if you are doing something else on the screen.  Note: The ASCII Code for a solid block character is 0xDB. You have to write the COMPLETE CODE.

Note: Run the code given below before you run your own code because it is a TSR and updates the buffer.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| c:\> |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

  Figure 1 – Illustration of the graphical equalizer displayed on screen bottom centre.

Other program:

org 100h

jmp start

seed: dw 17

prime1: dw 1033

prime2: dw 1613

buffer: db 0,0,0,0,0,0,0,0,0,0,0

fillbuffer:

push es

push ax

push bx

push dx

push si

push cx

mov ax, 0xc800

mov es,ax

mov si,0

mov cx,10

;Fill buffer

again:

mov ax,[cs:seed]

mov bx,[cs:prime1]

mul bx

add ax,[cs:prime2]

mov dx,1

mov bx,255

div bx

mov [cs:seed],ax ; New seed

mov [es:si],al

inc si

loop again

mov al, 0x20

out 0x20, al ; end of interrupt

pop cx

pop si

pop dx

pop bx

pop ax

pop es

iret ; return from interrupt

start:

xor ax, ax

mov es, ax ; point es to IVT base

cli ; disable interrupts

mov word [es:8\*4], fillbuffer; store offset at n\*4

mov [es:8\*4+2], cs ; store segment at n\*4+2

sti ; enable interrupts

mov dx, start ; end of resident portion

add dx, 15 ; round up to next para

mov cl, 4

shr dx, cl ; number of paras

mov ax, 0x3100 ; terminate and stay resident

int 0x21