;------------------------------------------------------------------------------- ; ; ;Write a TSR to generate the pattern of the frequency tones by reading ;the Real Time Clock (RTC). The duration of the each tone is solely decided by the ;programmer

;-------------------------------------------------------------------------------

;displays a clock with beep after 5 sec

.model tiny

.code

org 100h

main: jmp init0809

int09 dd ?

int08 dd ?

ccl db ?

hr db ?

min db ?

sec db ?

t db ?

temp db 0

disp:

push ax

push bx

push cx

push dx

push ds

push ss

push es

push si

push cs

pop ds

cont08:

mov ah,03h

mov bh,00

int 10h

push cx

push dx

mov ccl,45h

mov ah,02h

int 1ah

mov hr,ch

mov min,cl

mov sec,dh

inc cs:temp

mov al,hr

call printhex

call printcolon

mov al,min

call printhex

call printcolon

mov al,sec

call printhex

pop dx

pop cx

mov ah,02

mov bh,00

int 10h

cmp cs:temp,100

jne exit08

mov cs:temp,00h

mov al, 182 ; Prepare the speaker for the

out 43h, al ; note.

mov ax, 3619 ; Frequency number (in decimal)

; for C.

out 42h, al ; Output low byte.

mov al, ah ; Output high byte.

out 42h, al

in al, 61h ; Turn on note (get value from

; port 61h).

or al, 00000011b ; Set bits 1 and 0.

out 61h, al ; Send new value.

mov bx, 25 ; Pause for duration of note.

pause1:

mov cx, 65535

pause2:

dec cx

jne pause2

dec bx

jne pause1

in al, 61h ; Turn off note (get value from

; port 61h).

and al, 11111100b ; Reset bits 1 and 0.

out 61h, al

exit08:

pop si

pop es

pop ss

pop ds

pop dx

pop cx

pop bx

pop ax

jmp dword ptr cs:int08

init0809:

cli

mov ah,35h

mov al,08

int 21h

mov word ptr cs:int08,bx

mov word ptr cs:int08+2,es

mov ah,25h

mov al,08

mov dx,offset disp

int 21h

mov ah,31h

mov dx,offset init0809

sti

int 21h

printhex proc near

mov t,al

and al,0f0h

mov cl,04

shr al,cl

call printdigit

mov al,t

and al,0fh

call printdigit

ret

printhex endp

printdigit proc near

inc ccl

mov ah,02h

mov bh,00

mov dh,0

mov dl,ccl

int 10h

mov ah,0ah

add al,30h

mov bh,00

mov bl,15

mov cx,0001

int 10h

ret

printdigit endp

printcolon proc near

inc ccl

mov ah,02h

mov bh,00

mov dh,00

mov dl,ccl

int 10h

mov ah,0ah

mov al,':'

mov bh,00

mov bl,15

mov cx,0001

int 10h

ret

printcolon endp

end main

**OUTPUT :**

