12.

a) Write a program to create a file (input file) and to delete an existing file.

```
.model small

.data

.code

msg1 db 10,13, "Press 1:create file 2:Delete$"

msg2 db 10,13, "Enter your choice$"

m4 db 10,13,"enter the filename$"

msg3 db 10,13, "Error$"

f1 db 20 dup(00)

f2 db 20 dup(00)

mov ax,@data

mov ds,ax

 lea dx,msg1

 call display

 mov ah,01h

 int 21h

 cmp al,'1'

 je CREATE

 cmp al,'2'

 je DELETE

 error:

 call display

 exit:

 int 21h

 CREATE:

 mov ah,09h
 int 21h
 lea si,f1
enter: mov ah,01h
 int 21h
 cmp al,0dh
 je c1
 mov [si],al
 inc si
 jmp enter
c1: mov cx,0
 lea dx,F1
 mov ah,3ch
 int 21h
 jc error
 jmp exit
DELETE:
 mov ah,09h
 int 21h
 lea si,f2
enter1: mov ah,01h
 int 21h
 cmp al,0dh
 je c2
 mov [si],al
 inc si
 jmp enter1
 c2: lea dx,F2
 mov ah,41h

 int 21h

 jc error

 jmp exit

 display PROC

 mov ah,09h

 int 21h

 RET

lea dx,msg3

mov ah,4ch

lea dx,m4

lea dx,m4

 display ENDP

 end
```

b) Drive an elevator interface in the following way:

i. Initially the elevator should be in the ground floor, with all requests in OFF state.

ii. When a request is made from a floor, the elevator should move to that floor, wait there

for a couple of seconds (approximately), and then come down to ground floor and stop. If some

requests occur during going up or coming down they should be ignored.

```
.model small

.stack

.data

.code

up1: in al,dx

GND: mov cx,01H
```

```
PA equ 9800H

PB equ 9801H

PC equ 9802H

CR equ 9803H

mov ax,@data

mov ds,ax

mov al,82H

mov dx,CR

out dx,al

mov dx,0F0H

mov dx,PA

out dx,al

mov dx,PB

and al,0FH

cmp al,0FH

je up1

shr al,01H

jnc GND

shr al,01H

jnc First

shr al,01H

jnc Second

shr al,01H

jnc Third

call SERVICE

jmp exit

First: mov cx,04H
```

```
call SERVICE

jmp exit

Second: mov cx,07H

call SERVICE

jmp exit

Third : mov cx,0aH

call SERVICE

jmp exit

exit : mov ah,4cH

int 21H

SERVICE PROC

mov bl,cl

mov al,0F0H

mov dx,PA

up2: out dx,al

call delay

inc al

loop up2

call delay

call delay

dec al

mov cl,bl

up3: out dx,al

call delay

dec al

loop up3

RET
```

```
SERVICE ENDP
delay PROC
push bx
push cx
up5: mov bx,0FFFFH
up4: dec bx
jnz up4
loop up5
pop cx
pop bx
RET
delay ENDP
end
```