1. READ AN ALPHANUMERIC CHARACTER AND DISPLAY ITS EQUIVALENT ASCII CODE AT THE CENTER OF THE SCREEN.

```
.MODEL SMALL
GETCHAR MACRO
        MOV AH,01H
        INT 21H
ENDM

PUTCHAR MACRO CHAR
        MOV AH,02H
        MOV DL,CHAR
        INT 21H
ENDM

PRINTF MACRO MSG
        MOV AH,09H
        LEA DX,MSG
        INT 21H
ENDM

.DATA
        MSG1 DB "ENTER A CHARACTER:",10,13,"$"
        MSG2 DB "ASCII VALUE-->$"                       ; no 10,13 for this
please
        X DB 12
        Y DB 34
.CODE
        MOV AX,@DATA
        MOV DS,AX
        PRINTF MSG1
        GETCHAR
        MOV BH,AL       }       ; taking a copy of al into bh and bl
        MOV BL,AL       }
        AND BL,0FH              ; extracting lower nibble
        CMP BL,0AH
        JL L1
        ADD BL,07H
```

```
L1:  ADD BL,30H          ; converting to ascii value
     AND BH,0F0H         ; extracting upper nibble
     MOV CL,04
     SHR BH,CL
     CMP BH,0AH
     JL L2
     ADD BH,07H

L2:  ADD BH,30H          ; converting to ascii
     PUSH BX             ; save bh, bl values
     CALL CLS            ; calling the clear screen procedure

     MOV DH,X
     MOV DL,Y            ; setting the cursor
     MOV AH,02H
     INT 10H             ; its 10h not 21h (don't confuse)

     PRINTF MSG2
     POP BX              ; get saved value
     PUTCHAR BH          ; print BH & BL
     PUTCHAR BL
     MOV AH,4CH          ; terminate
     INT 21H

     CLS PROC NEAR
     MOV AH,0FH          ; get the current mode
     INT 10H
     MOV AH,00H          ; clear that current mode
     INT 10H
     RET
     CLS ENDP
     END
```

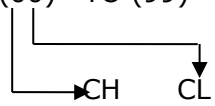
2. READ YOUR NAME FROM THE KEYBOARD AND DISPLAY IT AT A SPECIFIED LOCATION ON SCREEN IN FRONT OF THE MESSAGE 'WHAT IS YOUR NAME?' YOU MUST CLEAR THE ENTIRE SCREEN BEFORE DISPLAY

```
.MODEL SMALL
.DATA
        X DB 18          ; cursor positions
        Y DB 34

        NAME DB 30 DUP("$")     ;  30  bytes  all  filled  with  '$' symbol…temporarily.
        ASK1 DB "TYPE YOUR NAME: $"
        VTUSTRING DB "WHAT IS YOUR NAME: $"
.CODE
        MOV AX,@DATA
        MOV DS,AX
        LEA SI,NAME
        LEA DX, ASK1
        MOV AH,09H       ; display ASK
        INT 21H

TAKE:   MOV AH,01H
        INT 21H
        MOV [SI],AL      ; take a single character, put that in the array.
        INC SI           ; repeat until enter is pressed (13h).
        CMP AL,13
        JNE TAKE

        CALL CLS         ; CLRSCR();

        MOV DH,X
        MOV DL,Y
        MOV AH,02H       ; set the cursor
        INT 10H
        LEA DX, VTUSTRING
        MOV AH,09H       ; display VTUSTRING
        INT 21H
```

```
        LEA DX,NAME
        MOV AH,09H      ; display name
        INT 21H

        MOV AH,4CH      ; terminate
        INT 21H


CLS PROC NEAR
        MOV AH,0FH      ; get the current mode
        INT 10H
        MOV AH,00H      ; clear that current mode
        INT 10H
        RET
CLS ENDP
END
```

3. PROGRAM TO SIMULATE A DECIMAL UP-COUNTER TO DISPLAY 00-99

```
.MODEL SMALL
.CODE
        CALL CLS            ;  clear screen first
        MOV CH,30H          ; starting from      (00)   TO (99)
        MOV CL,30H
NEXT:   MOV AH,2
        MOV DH,12           ; set the cursor
        MOV DL,39
        INT 10H

        MOV DL,CH           ; display the first digit
        MOV AH,2
        INT 21H

        MOV DL,CL           ; display the second digit
        MOV AH,2
        INT 21H

        CALL DELAY          ; call delay (make user to see the stuff)

        INC CL              ; (2nd digit)++
        CMP CL,39H          ; while(2nd digit<=9)
        JBE NEXT                                    -> 39H

        MOV CL,30H          ; once 2nd digit crosses 39H, make it 0 by manually
filling 30H

        INC CH              ; (1st digit)++
        CMP CH,39H          ; while(1st digit<=9)
        JBE NEXT                                    -> 39H

        MOV AH,4CH
        INT 21H             ;exit
```



```
DELAY PROC NEAR
        MOV SI,05FFFH
OLOOP:MOV DI,0FFFFH     ; do a waste job waste number of times!!!!
ILOOP:DEC DI            ;something like this
        JNZ ILOOP
        DEC SI
        JNZ OLOOP
        RET
DLY ENDP

CLS PROC NEAR
        MOV AH,0FH
        INT 10H
        MOV AH,00H
        INT 10H
        RET
CLS ENDP
END
```

```
for (si=bignumber; si>=0; si--)
{
        for(di=bignumber; di>=0;
di--)
        {
                Do nothing;
        }
}
```

basically, keep decrementing a huge number till zero huge number of times.

By the time, microprocessor does this huge decrements, you can actually see your front-end output.

4. PROGRAM TO CREATE A FILE (INPUT FILE) AND DELETE AN EXISTING FILE.

```
.MODEL SMALL
PRINTF MACRO MSG
        LEA DX,MSG
        MOV AH,09H
        INT 21H
ENDM

.DATA
        CREATE DB 10,13," ENTER FILE NAME TO CREATE $"
        DELETE DB 10,13," ENTER FILE NAME TO DELETE $"
        CHOICE DB 10,13," 1: CREATE 2: DELETE --- ENTER YOUR CHOICE$"
        ERRORMSG  DB 10,13," ERROR $"
        FILECREATED DB 10,13,"FILE CREATED SUCCESSFULLY—SEE THE DIRECTORY TO
SEE IT$"
        FILEDELETED DB 10,13,"FILE DELETED SUCCESSFULLY—SEE THE DIRECTORY TO
SEE IT$"

     FILENAME DB 80 DUP(0); array to store your filename - of 80 bytes (all filled
with 0)
                                                    Guys, this should be zero
.CODE

        MOV AX,@DATA
        MOV DS,AX
        PRINTF CHOICE
        MOV AH,1          ; getchar()…the pressed key's ASCII value will go to 'AL'
        INT 21H
        CMP AL,'1'        ; if you hit 1, then go to CREATEFILE
        JZ CREATEFILE                          or CMP AL, 31H
        CMP AL,'2'        ; if you hit 1, then go to DELETEFILE
        JZ DELETEFILE                          or CMP AL, 32H

ERROR:  PRINTF ERRORMSG
        MOV AH,4CH        ; error in any case, display ERROR and then exit.
        INT 21H
CREATEFILE:    PRINTF CREATE
```

```
        CALL READ
        MOV CX,0                    ; a normal file (without any
extension)
        MOV AH,3CH
        LEA DX,FILENAME     file creation code
        INT 21H
        JC ERROR
       PRINTF FILECREATED
        MOV AH,4CH
        INT 21H

DELETEFILE:   PRINTF DELETE
        CALL READ
        LEA DX,FILENAME
        MOV AH,41H          file deletion code
        INT 21H
        JC ERROR
        PRINTF FILEDELETED
        MOV AH,4CH          ; exit
        INT 21H
```

See if something goes wrong while creating or deleting a file, your microprocessor automatically enables or highlights carry flag(CF). Means CF becomes 1.

So if CF=1, then go to ERROR else everything is fine.

```
READ PROC

        LEA SI,FILENAME     ; read character by character, compare that with
enter key's
TAKE: MOV AH,01H
        INT 21H             ASCII value. If that's matching, return. Else, move
that to
        CMP AL,0DH           an array called FILENAME.
        JE DONE
        MOV [SI],AL
        INC SI
        JMP TAKE
DONE:   RET
READ ENDP
END
```