**Program :**

Data segment

    msg db 0dh,0ah,"-- -- -- -- OF DF IF TF SF ZF -- AF -- PF -- CF$"

    newl db 0dh,0ah,"$"

Data ends

Code segment

    assume CS:Code,DS:Data

start:

    mov ax, Data

    mov DS, ax

    mov dx,offset msg  ; Display contents of variable msg

    mov ah,09h

    int 21h

    mov dx,offset newl  ; go to new line

    mov ah,09h

    int 21h

    cld ; clear dirn flag

    stc ; set carry flag

    sti ; set interrupt enable flag

    pushf

    pop bx

    mov cx,10h ;set counter

    loop1 :

        mov ax, 8000h

        and ax,bx

        jz zero

        mov dl,31h ;print 1

        mov ah,02h

        int 21h

        jmp space

    zero :

        mov dl,30h ;print 0

        mov ah,02h

        int 21h

    space :

        mov dl,' ' ;print two spaces

        mov ah,02h

        int 21h

        int 21h

    sal bx,01h

    loop loop1

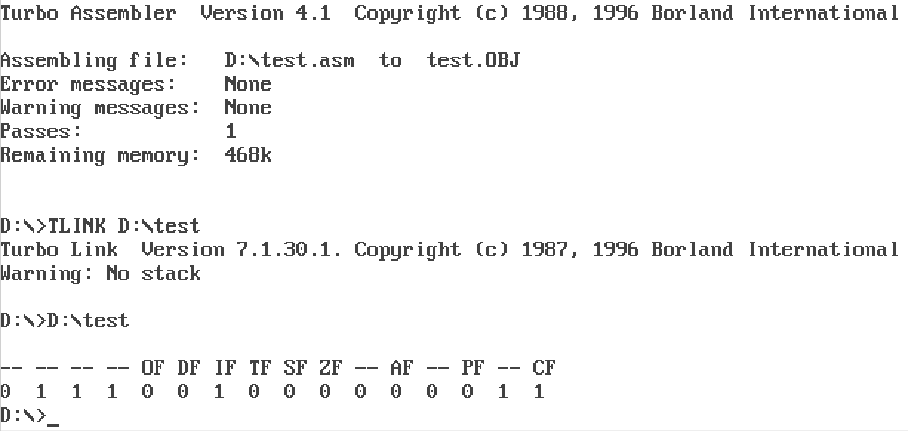
    mov ah,4ch

    int 21h

Code ends

end start

**Output :**

****

**Program :**

Data segment

    msg db 0dh,0ah,"Enter the number: $"

    result db 0dh,0ah,"The factorial of number is: $"

Data ends

Code segment

    assume CS:Code,DS:Data

start:

    mov ax, Data

    mov DS, ax

    mov dx,offset msg ; Display contents of variable msg

    mov ah,09h

    int 21h

    mov ah,01h

    int 21h

    sub al,30h

    mov bl,al

    rol bl,4

    mov ah,01h

    int 21h

    sub al,30h

    add bl,al

    cmp bl,02h

    jc one

    mov ax,01h

    loop1:

    mul bl

    dec bl

    jnz loop1

    mov bl,al

    mov cl,al

    jp skip

    one:

        mov bl,01h

        mov cl,01h

    skip:

    mov dx,offset result

    mov ah,09h

    int 21h

    mov dl, al

    mov ah,02h

    int 21h

    mov al,cl

    and al,0fh

    call HexToAscii

    mov dl, al

    mov ah,02h

    int 21h

    mov ah,4ch ; Terminate the program

    int 21h

        HexToAscii proc

        cmp al,0ah ; If it is greater than or equal to 0a then we also need to add 07 along with 30

        jc skip1

        add al,07h

        skip1: add al,30h

        ret

        endp

Code ends

end start

**Output :**

