Ch#03

1. Which registers are changed by the CMP instruction?

The compare instruction subtracts the source operand from the destination operand, updating the flags without changing either the source or the destination. So, it only changes the flag register.

2. What are the different types of jumps available? Describe position relative addressing.

The types of jumps are:

- Near Jumps
- Short Jumps
- Far Jumps

Position relative addressing: position relative addressing in contrast to absolute addressing does not tell the exact address rather it is telling how much forward or backward to go from the current position of IP in the current code segment.

3.If AX=8FFF and BX=0FFF and "cmp ax, bx" is executed, which of the following jumps will be taken? Each part is independent of others. Also give the value of Z, S, and C flags.

a. jg greater

b. jl smaller

c. ja above

d. jb below

Instructions	**Jump**	**ZF**	**SF**	**CF**
Jg greater	Not taken	0	1	0
Jl smaller	Taken	0	1	0

Ja above	Taken	0	1	0
Jb below	Not taken	0	1	0

4.Write a program in Assembly Language to find the maximum number and the minimum number from an array of ten numbers.

Solution:

```
 [org 0x0100]
          jmp start              ; unconditionally jump over data

array1:    dw   10, 5, 30, 4, 50, 1, 20, 6, 40, 8
min:        dw   0
max:         dw   0

start:

          mov  bx, 0              ; initialize array index to zero
          mov  ax, 0              ; initialize min to zero
         mov  ax, [array1+bx]    ; minimum number to ax
         mov cx,10

top1:     cmp  ax, [array1+bx]      ; are we find the minimum number
          jle  end1                          ; if less or equal number
          mov ax,[array1+bx]      ;ax contains the minimum number

end1:
          add  bx, 2            ; advance bx to next index
          loop top1
         mov  [min], ax        ; write back minimum in memory
         mov  bx, 0              ; initialize array index to zero
         mov  ax, 0              ; initialize max to zero
         mov  ax, [array1+bx]    ; maximum number to ax
         mov cx,10

top2:     cmp  ax, [array1+bx]    ; are we find the maximum number
          jge  end2                     ; if greater or equal number
           mov ax,[array1+bx]   ;ax contains the maximum number


end2:
          add  bx, 2              ; advance bx to next index
          loop top2

           mov  [max], ax         ; write back maximum number in memory
          mov  ax, 0x4c00         ; terminate program
          int  0x21
```

5. Write a program to search a particular element from an array using binary search. If the element is found set AX to one and otherwise to zero.

```
;Binary Search

[org 0x0100]

        jmp start1

data:  db 1,2,3,4,5,6,7,8,9,10,11
start: db  0
end:   db  10
key:   db  -1

start1:  mov al,[key]

loop1: mov cl,[start]
          cmp cl,[end]
          ja end1                                    ;Checking if(start<=end), if not then jump to
end1

          mov dl,[start]
          add dl,[end]                       ;dl is basically now start + end
          sar dl,1                                   ;here dl is being divided by 2
          mov bl,dl                                  ;bl is mid and is calculated by (start + end)/2


         cmp  al, [data + bx]
         je store                                    ; agar data mil gaya tw program end kar do
         ja step1                                    ; agar data greater hai current element sey
         jb step2                                    ; agar data smaller hai current element sey

step1: add dl,1                              ;mid + 1 kar do
          mov [start],dl                             ;start ko ab mid + 1 kar do taakey hum mid se
aagey jaga par dekhein
          jmp loop1

step2: sub dl,1                              ;mid -1 kar do
          mov[end],dl                                ;end ko ab mid - 1 kar do taakey hum mid se
previous jaga par dekhein
```

```
          jmp loop1


store: mov ax, 1
          mov ax,0x4c00
          int 21h

end1:  mov ax,0
          mov ax,0x4c00
          int 21h
```

6.Write a program to calculate the factorial of a number where factorial is defined as:
factorial(x) = x*(x-1)*(x-2)*...*1 factorial(0) = 1

```
[org 0x0100]
mov bx,0
mov si,[l]
l1:
mov ax,[n+bx]
mov cx,ax
sub cx,1
l2:mul cx
        sub cx,1
        jnz l2
        mov [fact_num+bx],ax
        add bx,2
sub si,1
jne l1

mov ax, 0x4c00
int 0x21
n: dw 3,5,4,8,7
fact_num: dw 0,0,0,0,0
l: dw 5
```