Flow control Instructions

1. **IF –THEN**
2. **IF-THEN-ELSE**
3. **SWITCH-CASE**

LOOP

1. **FOR ; WHILE ; DO-WHILE**

Conditional Jumps instructions:

Signed Jumps

Symbol	Description
JG/JNLE	Jump if greater than / jump if not less than or equal to
JGE/JNL	Jump if greater than or equal to / jump if not less than
JL / JNGE	Jump if less than / jump if not greater than or equal to
JLE / JNG	Jump if less than or equal to / jump if not greater than

Unsigned Jumps

Symbol	Description
JA/JNBE	Jump if above / jump if not below or equal to
JAE/JNB	Jump if above or equal to / jump if not below
JB / JNAE	Jump if below / jump if not above or equal to
JBE / JNA	Jump if below or equal to / jump if not above

Single flag Jumps

Symbol	Description
JE/JZ	Jump if equal / jump if equal to zero
JNE/JNZ	Jump if not equal / jump if not zero
JC	Jump if carry
JNC	Jump if no carry
JO	Jump if overflow
JNO	Jump if no overflow
JS	Jump if sign negative
JNS	Jump if non negative sign
JP/JPE	Jump if parity even
JNP/JPO	Jump if parity odd

CMP instructions

CMP destination, source

For example

```
CMP AX, BX
 JG  BELLOW ; AX > BX
```

Where , AX= 7FFFh, BX= 0001h

For example

```
CMP AX, BX
  JA   BELLOW
```

Where , AX= 7FFFh, BX= 8000h

JMP instructions

JMP destination

For example

```
CMP AX, BX
JA   BELLOW
JMP  TOP
```

IF –THEN

Example :

```
IF AL < 0
THEN
    COPY  BL to AL
END_IF
```

In Assembly Code

```
; if AL < 0 can be coded as
      CMP  AL , 0
      JNL  END_IF
;THEN
     MOV AL , BL
END_IF:
```

IF-THEN-ELSE

IF AL <= BL
THEN
display the character in AL
ELSE
Display the character in BL
END_IF

In Assembly code it can be coded as

```
; print function
            MOV AH,2
;IF AL <= BL
      CMP AL, BL
       JNBE    ELSE_
; THEN
       MOV  DL , AL
      JMP    DISPLAY
ELSE_:
      MOV  DL, BL

DISPLAY:
      INT  21H
```

SWITCH-CASE

Example:

CASE BL

< 0 : display a character in BH

= 0 : display a character in CL

> 0 : display a character in CH

In Assembly it can be coded as

```
; CASE BL
        CMP  BL , 0
        JL    TOP
        JE    MIDDLE
        JG   LAST
TOP:
        MOV   AH, 2
        MOV   DL, BH
        INT   21H
      JMP   EXIT
MIDDLE:
        MOV  AH,  2
        MOV   DL,  CL
        INT  21H
      JMP   EXIT
```

LAST:

```
MOV  AH,  2
MOV   Dl,  CH
INT  21H
```

EXIT:

FOR LOOP

Example: Write a count-controlled Loop to display a row of 80 characters.

```
FOR  80 times  DO
display   '*'
END_FOR
```

In Assembly it can be coded as

```
        MOV   CL, 80 ; CL = 80 (Loop counter)
        MOV   AH, 2  ;  prepare to display
        MOV   DL, '*'  ; DL= '*'
TOP:
        INT  21H   ; CALL  function to display
        LOOP   TOP ; CL = CL- 1(LOOP is Keyword)
```

WHILE LOOP

Example: Write a count-controlled Loop to display a row of 80 characters.

```
WHILE condition   DO
display   '*'  80 times
END_WHILE
```

In Assembly it can be coded as

```
                MOV   CL, 80 ; CL = 80 (Loop counter)
                MOV   AH, 2  ;  prepare to display
                MOV   DL, '*'  ; DL= '*'
                MOV   BL, 0   ; BL = 0
 TOP:
                CMP  CL , BL  ;  if CL == 0
                JE     END_WHILE
                INT  21H   ; CALL  function to display
                DEC   CL ;  CL= CL -1

                JMP  TOP

END_WHILE:
```

DO WHILE LOOP

Example: Write a count-controlled Loop to display a row of 80 characters.

```
DO
display  '*'  80 times
WHILE   condition
```

In Assembly it can be coded as

```
                MOV   CL, 80 ; CL = 80 (Loop counter)
                MOV   AH, 2  ;  prepare to display
                MOV   DL, '*'  ; DL= '*'
               MOV   BL ,  0  ; BL= 0
 TOP:

                INT  21H   ; CALL  function to display


CMP  CL , BL ;  if CL == 0
JE     END_WHILE

DEC   CL ;  CL= CL -1

 JMP  TOP

END_WHILE:
```

1. **Write an Assembly program to find largest number from three input numbers using IF-ELSE.**
2. **Write an Assembly program to find smallest number from three input numbers using IF-ELSE.**
3. **Write an Assembly program that add two number if BL>CL and subtract two number if BL< CL using IF-ELSE.**
4. **Write an Assembly program that add two number if BL>CL and subtract two number if BL< CL using SWITCH-CASE.**
5. **Write an Assembly program that print all Uppercase letter using FOR , WHILE and DO WHILE Loop.**
6. **Write an Assembly program that print all Lowercse letter using FOR , WHILE and DO WHILE Loop.**
7. **Write an Assembly program that print all Hexadecimal digit using FOR , WHILE and DO WHILE Loop.**
8. **Write an Assembly program that print all ASCII characters using FOR , WHILE and DO WHILE Loop.**
9. **Write an assembly program that print Fibonacci series up to N where N<=255 using FOR , WHILE and DO WHILE Loop.**
10. **Write an assembly program that print prime number from 0 to 255 using FOR,WHILE and DO WHILE Loop.**
11. **Write an assembly program that input a number from 0 to 255 and print “prime ” or “not prime” using FOR,WHILE and DO WHILE Loop.**