Lecture-02

Flow Control Instructions:

1, Unconditional Jump

JMP instruction: It causes an unconditional transfer of control

Syntax: JMP destination

2. Conditional Jumps

Syntax: **Jxxx destination_label**

The CMP instruction:

The jump condition is often provided by the CMP (compare) instruction. It has the form

Syntax: **CMP destination, source**

```
CMP AX, DX       ; AX-DX
SUB AX, DX       ; AX=AX-DX
```

1. **Read a character and display it 80 times on the next line.**

Solution (instructions):

```
MOV CX, 80
MOV AH, 1
INT 21H     ; By default the input character is stored in AL register
MOV AH, 2
MOV DL, AL
label:
INT 21H
```

```
LOOP label
```

2. **Write a program to display a “?”, read two capital letters, and display them on the next line in alphabetical order.**

Assembly code for 8086:

```
.MODEL SMALL
.STACK 100H
.DATA
ms db ‘Enter two letters $’
ch1 DB ?                ; ‘?’ symbol for variables that are not initialized.
ch2 DB ?
.CODE
MAIN PROC
MOV AX, @data;
MOV DS, AX
;prompt the user
MOV AH, 9
LEA DX, ms
INT 21H
;read two characters
MOV AH, 1
INT 21H
MOV ch1, AL
INT 21H
MOV ch2, AL
; Comparing the values
CMP ch1, AL
```

```
JB ascending              ; JB=Jump if below, condition CF=1
JA descending             ; JA= Jump if above, condition: CF=0 and ZF=0
JMP exit
ascending:
MOV AH, 2
MOV DL, ch1
INT 21H
MOV DL, ch2
INT 21H
JMP exit

descending:
MOV AH, 2
MOV DL, ch2
INT 21H
MOV DL, ch1
INT 21H
JMP exit
Exit:
MOV AH, 4CH
INT 21H
MAIN ENDP
END MAIN
```

Homework

Lab Work-02:

1. Draw the following pattern (N.B. the length of the pyramid can be changed)

```
   *
  ***
 *****
*******
```

2. Using only MOV, ADD, SUB, INC, DEC and NEG translate the following high level language assignment statements into assembly language. A, B and C are word variables.
   - i. A=B-A
   - ii. A= - (A+1)
   - iii. C=A+B
   - iv. B=3*B+7
   - v. A=B-A-1

   Sample input/output:

   Enter the value of A and B= 2 3

   Choose option (problem number): 3

   The result of 3 no. problem is=5
3. Even or odd check
4. Whether a input number is prime or not/ Prime check
5. Reverse an input string.

   Sample input: A report

   Sample output: troper A
6. Write a assembly code to perform the following:

   Put the sum 1+4+7+….+148 in AX