Assembly Language Lab Four

JA label

Short Jump if first operand is Above second operand (as set by CMP instruction). Unsigned.

Algorithm:

if (CF = 0) and (ZF = 0) then jump

JAE label

Short Jump if first operand is Above or Equal to second operand (as set by CMP instruction). Unsigned.

Algorithm:

if CF = 0 then jump

JB label

Short Jump if first operand is Below second operand (as set by CMP instruction). Unsigned.

Algorithm:

if CF = 1 then jump

JBE label

Short Jump if first operand is Below or Equal to second operand (as set by CMP instruction). Unsigned.

Algorithm:

if CF = 1 or ZF = 1 then jump

JMP label

Unconditional Jump. Transfers control to another part of the program. *4-byte address* may be entered in this form: 1234h:5678h, first value is a segment second value is an offset.

Algorithm:

always jump

Q) Write an assembly code to find the average value of five numbers store in memory start at location 200h, and store the result in AL.

```
        MOV CL,5
        MOV BX,200H
        XOR AX, AX
NEXT2:  ADD AL, [BX]
        INC BX
        DEC CL
        JNZ NEXT2
        MOV DH,5
        DIV DH
```

Q) Write an assembly code to find the large number between five numbers store in memory start at location 200h, and store the result in AL.

```
        org 100h
        MOV CL,5
        MOV BX,200H
NEXT3: MOV AL, [BX]
NEXT2: DEC CL
        JZ NEXT1
        INC BX
        CMP AL, [BX]
        JAE NEXT2
        JMP NEXT3
NEXT1: ret
```

Q) Write an assembly code to find the location of character “A” in memory start from 200h to 209h, and store the location in BX.

```
        org 100h
        MOV CL,10
        MOV SI,200H
        MOV AH,41H
        XOR BX, BX
Ali:    CMP AH, [SI]
        JZ Ahmed
        INC SI
        DEC CL
        JNZ Ali
        JMP Omar
Ahmed:  MOV BX, SI
Omar:   ret
```