Lab 7
Program Flow Control
Unconditional Jumps
Conditional Jumps

أعداد: م.م. سمر أميل يوسف

1. Unconditional Jumps

It is means where your program **can make decisions** according to **certain conditions.**

1. Unconditional Jumps

The basic instruction that transfers control to another point in the program is **JMP**.

The basic syntax of **JMP** instruction:

JMP label

To declare a *label* in your program, just type its name and add "**:**" to the end, label can be any character combination but it cannot start with a number, for example here are 3 legal label definitions:

```
label1:
label2:
a:
```

Label can be declared on a separate line or before any other instruction, for example:

```
x1:
MOV AX, 1

x2: MOV AX, 2
```

JMP instruction

EX1: compute the value of Y=5+2

```
MOV    AX, 5          ; set AX to 5.
MOV    BX, 2          ; set BX to 2.

JMP    calc           ; go to 'calc'.

back:  JMP stop       ; go to 'stop'.

calc:
ADD    AX, BX         ; add BX to AX.
JMP    back           ; go 'back'.

stop:

RET                   ; return to operating system.
```

2. Conditional Jumps

- **Short Conditional Jumps**

  Unlike **JMP** instruction that does an unconditional jump, there are instructions that do a conditional jumps (jump only when some conditions are in act). These instructions are divided in three groups, first group just test single flag, second compares numbers as signed, and third compares numbers as unsigned.

  Jump instructions that test single flag

Jump instructions that test single flag

Instruction	Description	Condition	Opposite Instruction
JZ , JE	Jump if Zero (Equal).	ZF = 1	JNZ, JNE
JC , JB, JNAE	Jump if Carry (Below, Not Above Equal).	CF = 1	JNC, JNB, JAE
JS	Jump if Sign.	SF = 1	JNS
JO	Jump if Overflow.	OF = 1	JNO
JPE, JP	Jump if Parity Even.	PF = 1	JPO

Conditional Jumps JZ, JE instruction

Jump if Zero or (Equal)

```
MOV AL, 5H
MOV BL, 5H
CMP AL,BL
JZ COMPUTE        ; Try JE
SUB AL,BL

COMPUTE:
ADD BL,AL
RET
```

JC instruction

Jump if Carry =1

```
MOV AL, 0FFH
MOV BL, 01H
ADD AL,BL
JC cal
SUB AL,BL

cal:
MOV CL,AL
RET
```

JS instruction

Jump if Sign =1

```
MOV AL, 9H
MOV BL, 8H
CMP BL,AL
JS cal
SUB AL,BL

cal:
ADD CL,AL
RET
```

JO instruction

```
    MOV AL, 11111111B
    MOV BL, 11111111B
    MUL BL
    JO NEW
JMP STOP
BACK: JMP STOP
NEW:MOV DX, AX
JMP BACK
STOP: RET
```

Jump instructions for unsigned numbers

Instruction	Description	Condition	Opposite Instruction
JA , JNBE	Jump if Above (>). Jump if Not Below or Equal (not <=).	CF = 0 and ZF = 0	JNA, JBE
JB , JNAE, JC	Jump if Below (<). Jump if Not Above or Equal (not >=). Jump if Carry.	CF = 1	JNB, JAE, JNC
JAE , JNB, JNC	Jump if Above or Equal (>=). Jump if Not Below (not <). Jump if Not Carry.	CF = 0	JNAE, JB
JBE , JNA	Jump if Below or Equal (<=). Jump if Not Above (not >).	CF = 1 or ZF = 1	JNBE, JA

Compute 5!

```
MOV AL,01
MOV BL ,05
MOV CL,01
FACT: MUL BL
SUB BL, CL
CMP BL,CL
JNZ FACT
RET
```

Q: Write program in assembly language to compute

Y= 5 +10+ …25

```
MOV BX,00
MOV AX,5
MOV CX,25
SUM: ADD BX,AX
    ADD AX,05
    CMP AX,CX
    JNZ SUM
    RET
```