Conditional Jump

The conditional jump depends on the values of the status flags i.e. (**S,Z,O,P,C**). for this type of jump, the jump does not happen if the condition that affect the status flags does not exist. Notice that these instructions are ***short*** jump instructions and this limits the jump within +127 and -128 bytes from the loction following the conditional jump. A list of each of the conditional jump instructions is shown in the following table:

Mnemonics	Meaning	Condition
JA	Above	CF=0 and ZF=0
JAE	Above or Equal	CF=0
JB	Belo	CF=1
JBE	Belo or Equal	CF=1 or ZF=1
JC	Carry	CF=1
JCXZ	CX register is zero	CF=0 or ZF=0
JE	Equal	ZF=1
JG	Greater	ZF=0 and SF=OF
JGE	Greater or Equal	SF=OF
JL	Less	(SF xor OF)=1
JLE	Less or Equal	((SF xor OF) or ZF)=1
JNA	Not Above	CF=1 or ZF=1
JNAE	Not Above nor Equal	CF=1
JNB	Not Below	CF=0
JNBE	Not Below nor Equal	CF=0 and ZF=0
JNC	No Carry	CF=0
JNE	Not Equal	ZF=0
JNG	Not Greater	((SF xor OF) or ZF)=1
JNGE	Not Greater nor Equal	(SF xor OF)=1
JNL	Not Less	SF=OF
JNLE	Not Less nor Equal	ZF=0 and SF=OF
JNO	No Overflow	OF=0
JNP	No Parity	PF=0

JNS	No Sign	SF=0
JNZ	No Zero	Zf=0
JO	Overflow	OF=1
JP	Parity	PF=1
JPE	Parity Even	PF=1
JPO	Parity Odd	PF=0
JS	Sign	SF=1
JZ	Zero	ZF=1

Important notes:

- The terms Greater than and Less than refer to ***signed numbers***, therefore when signed numbers are compared; the **JG, JL, JGE, JLE, JE,** and **JNE** are used.
- The terms Above and Below refer to ***unsigned*** numbers, therefore when unsigned numbers are compared use **JA, JB, JAE, JBE, JE,** and **JNE** are used.

According to the nature of there working, conditional jumps can be classified as:

i. If statement:

As mentioned before, the conditional jumps do not exist if the condition does not exist.

If Condition is existed
Jump to LABEL
} the condition depends on the type of the conditional jump

Continue program

JMP END

LABEL: certain instructions

END: HLT

Ex.1:

```
MOV AX,2500h
MOV BX,2006h
ADD AX,BX
JP *
MOV CL,02h
JMP END
*: MOV CL,01h
END: HLT
```

Ex.2:

```
MOV AX,5454h
CMP AL,AH
JZ *
MOV BYTEPTR[3000h],01h
JMP END
*: MOV BYTEPTR[3000h],02
END HLT
```

Ex.3: Write an A.L.P. to check the contents of a M.L. specified by [SI] if it is below or equal (50h) then put (01h) in reg. BL. Else put (02h) in reg. BL.

Sol.

```
CMP BYTEPTR[SI],50h
JBE *
MOV BL,02h
JMP END
*: MOV BL,01h
END: HLT
```

EX. 4: Write an A.L.P. to check the number stored in a M.L. specified by [BX+DI+2673h] if it is +ve put (01h) in reg. CH, if it is equal zero put (02h) in reg. CH, else put (03h) in reg. CH.

Sol.

```
CMP BYTEPTR[BX+DI+2673],00H
JE *
JA **
MOV CH,03
JMP END
*: MOV CH,01H
JMP END
**: MOV CH,02
END HLT
```