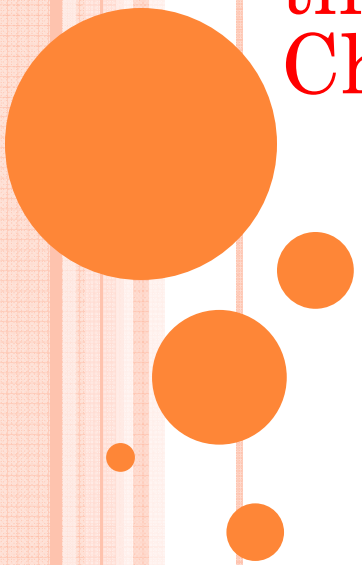
Branching Structure

Reference: Assembly Language Programming and Organization of the IBM PC – Charles Marut – Chapter 6



Branching Structure

Branching structure enables a program to take different paths, depending on conditions.

Three types of branching structures:

- IF-THEN
- IF-THRN-ELSE
- Case

Branching Structure: IF-THEN

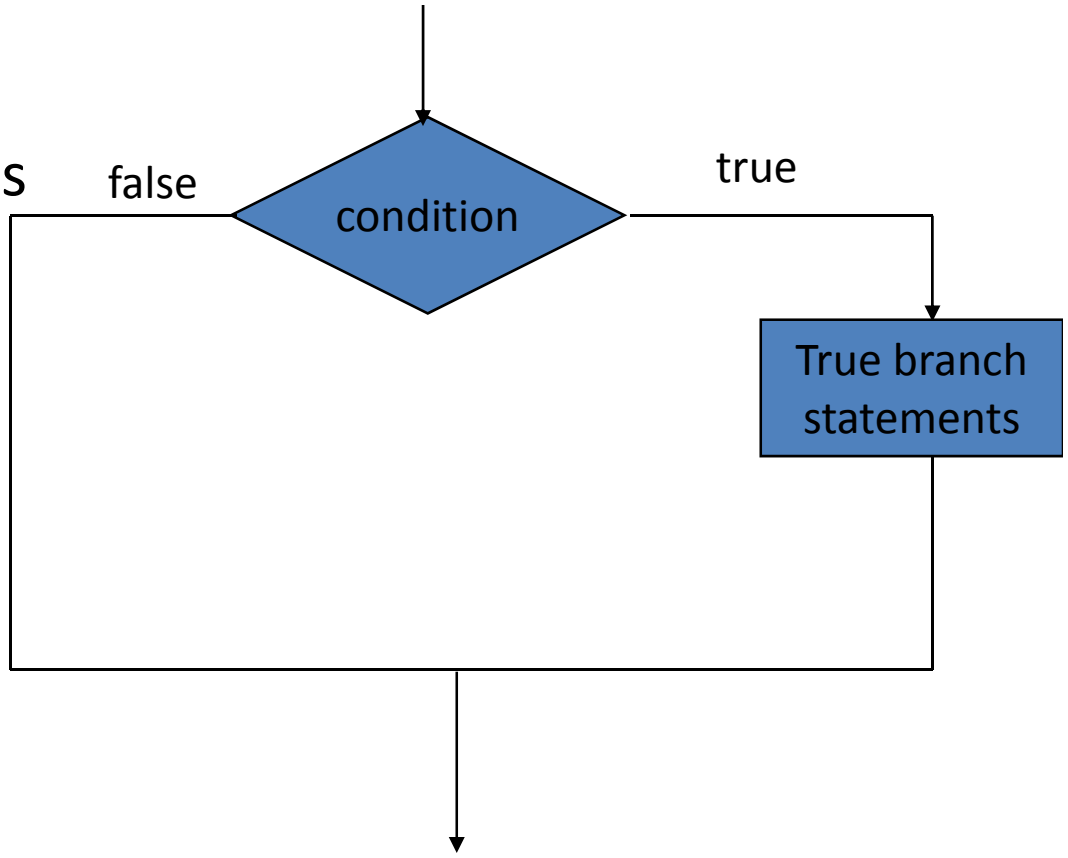
- ***Pseudocode in HLL:***

IF condition is true

THEN

execute true-branch statements

End_if



Example

Replace the number in AX by its absolute value.

Pseudocode	**Code**
IF AX <0	CMP AX,0
Then	JNL END_IF
replace AX by –AX	NEG AX
END_IF	END_IF:

Branching Structure: IF-THEN-ELSE

- ***Pseudocode in HLL:***

IF condition is true

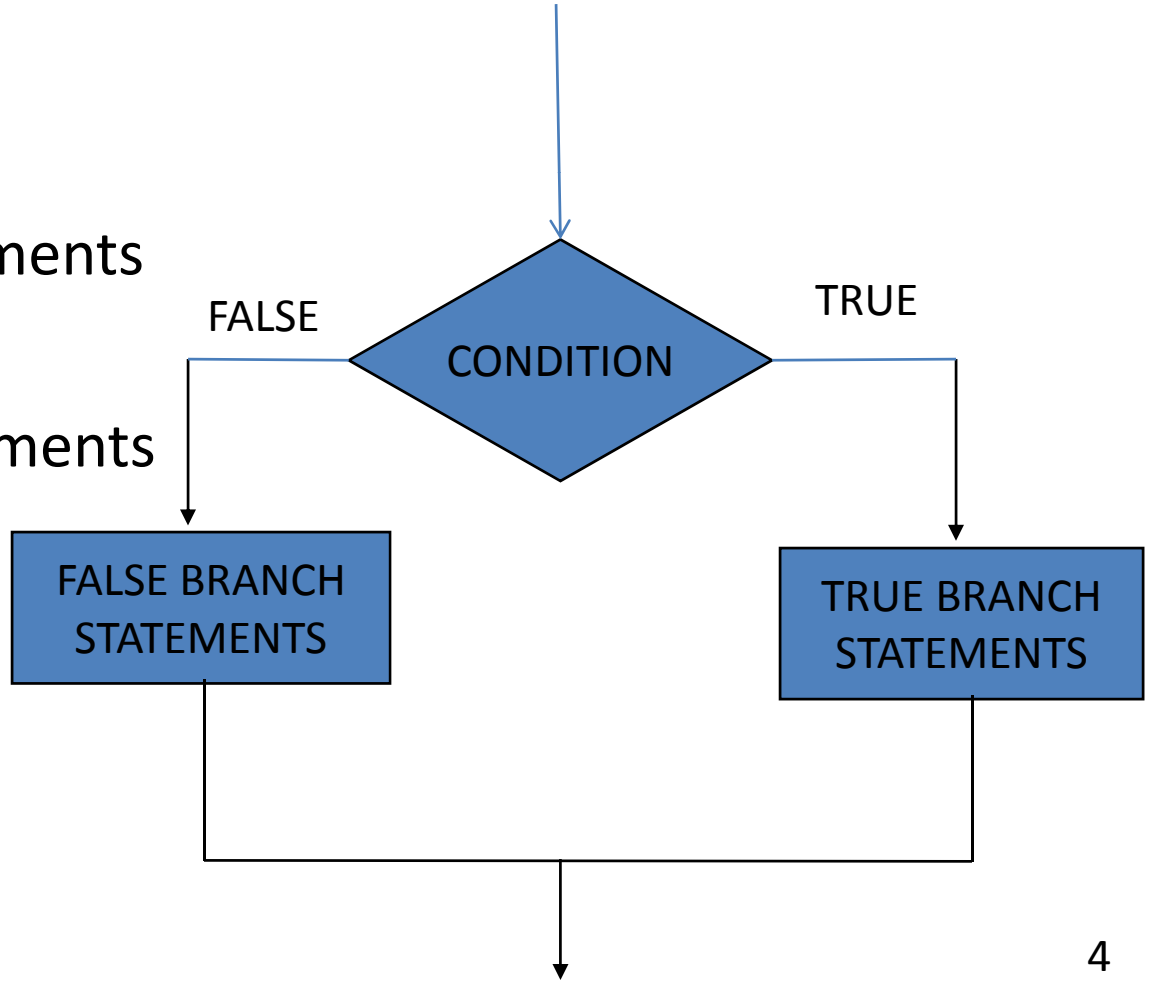
THEN

execute true-branch statements

ELSE

execute false-branch statements

End_IF



EXAMPLE

Suppose AL and BL contains extended ASCII characters. Display the one that comes first in the character sequence.

Pseudocode

```
IF AL < BL
THEN
display the char in AL
ELSE
display the char in BL
END_IF
```

Code:

```
        MOV AH,2        ; prepare to display
        CMP AL,BL       ; AL<=BL?
        JNBE ELSE_      ;no, display char in BL
        MOV DL,AL       ; move char to be displayed
        JMP DISPLAY     ; go to display
ELSE_:                  ; BL<AL
        MOV DL,BL
DISPLAY:
        INT 21H         ; display it
END_IF:
```

Branching Structure: CASE

A case is a multiway branch structure that tests a register, variable, or expression for particular values or range of values. The general form is as follows:

Case Expression

values_1: statements_1

values_2: statements_2

……..

values_n: statements_n

END_CASE

Case Expression:

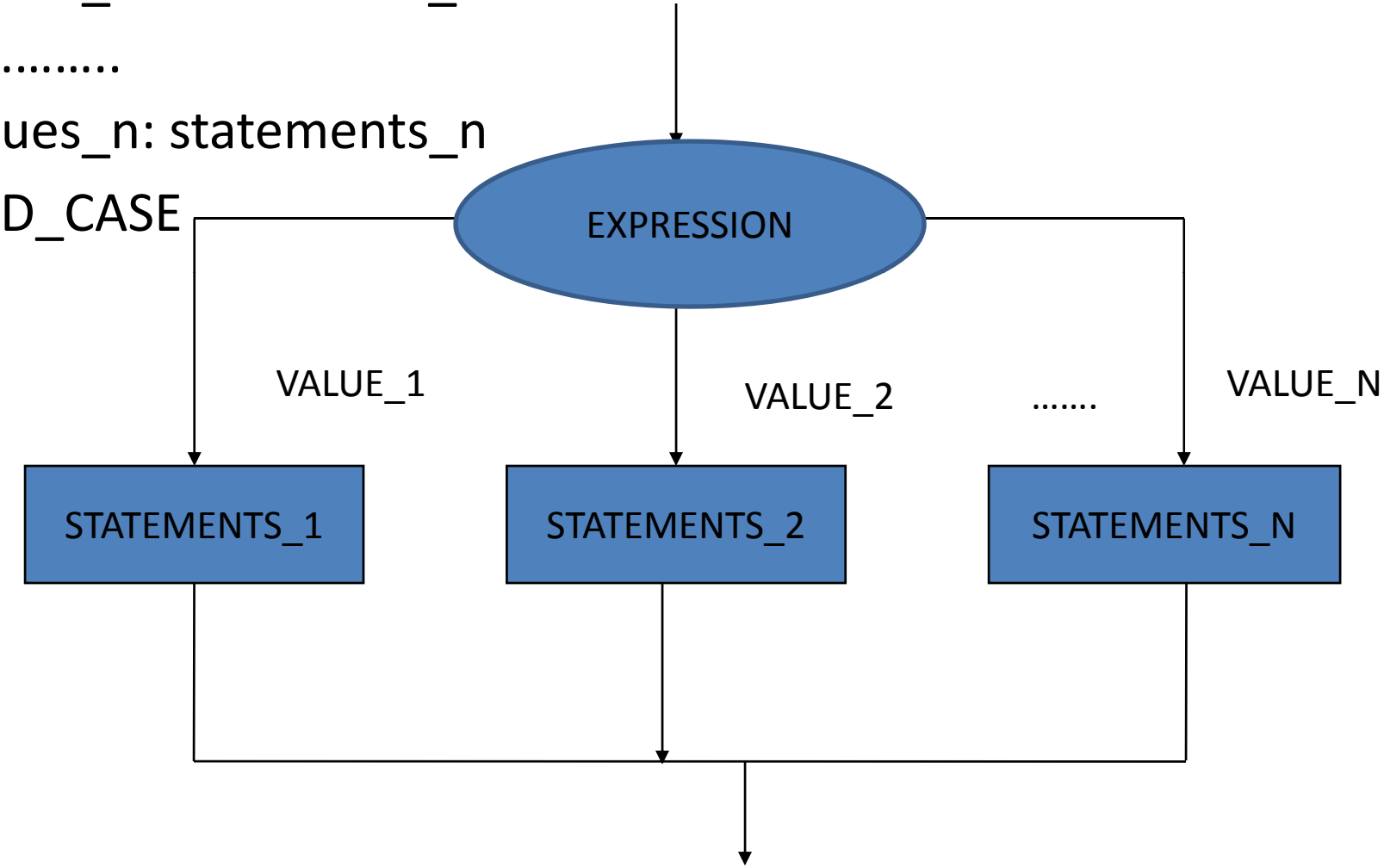
values_1: statements_1

values_2: statements_2

……..

values_n: statements_n

END_CASE



EXAMPLE

IF AX contains a negative number, put -1 in BX; if AX contains 0, put 0 in BX; if AX contains a positive number, put 1 in BX.

Pseudocode

Case AX

< 0:put -1 in BX

= 0:put 0 in BX

> 0:put 1 in BX

END_CASE

Code:

```
CMP AX,0          ; test AX
JL   NEG          ; AX<0
JE   ZERO         ;AX=0
JG   POS          ;AX>0
NEG:  MOV BX,-1   ; put -1 in BX
      JMP END_CASE ; and exit

ZERO:  MOV BX,0   ; put 0 in BX
       JMP END_CASE ; and exit

POS: MOV BX,1     ; put -1 in BX

END_CASE:
```

•***NOTE*** *: only* one CMP is needed, because jump instructions do not affect the flags.

Branches with compound conditions

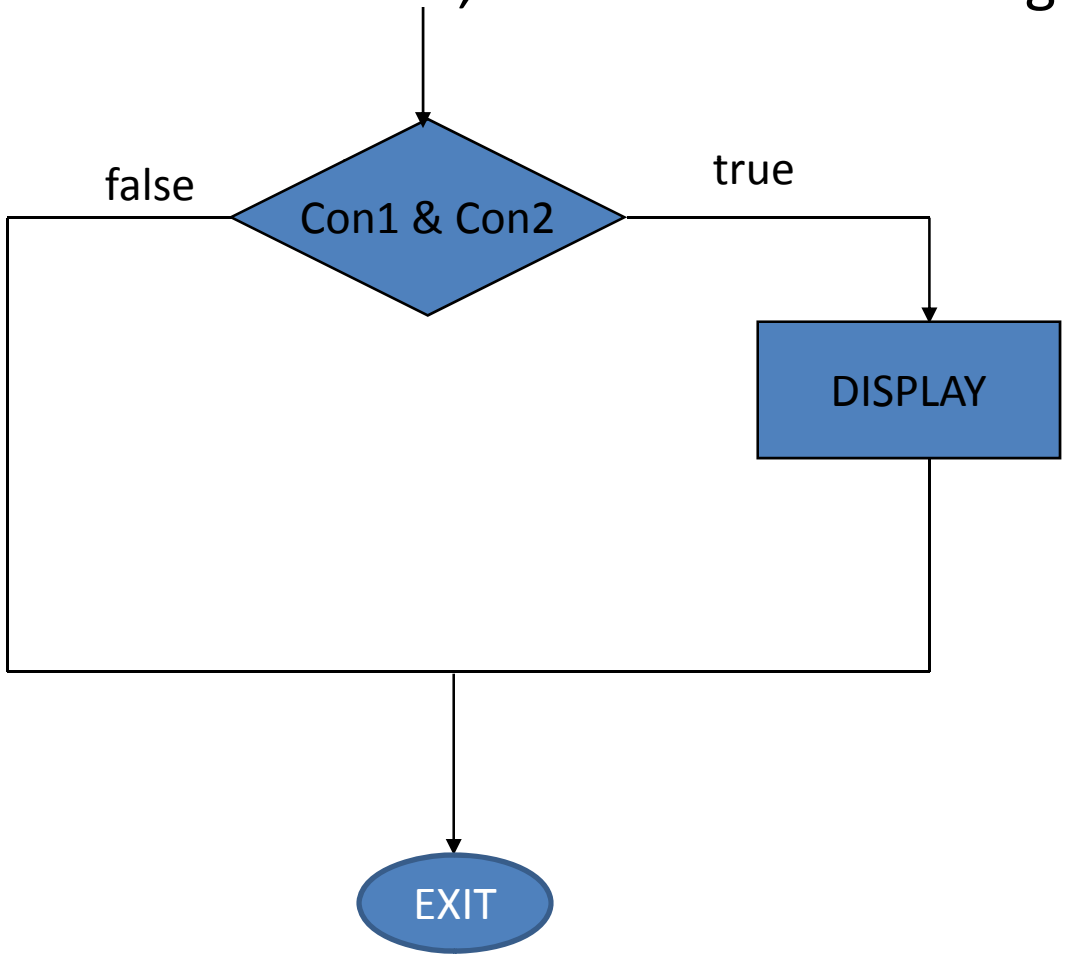
- Sometimes the branching condition in an IF or CASE takes the forms

  →Condition_1 and condition_2

  →Condition_1 or condition_2

AND conditions

- An AND condition is true if and only if condition_1 and condition_2 are both true.
- If any condition is false, then the whole thing is false.



EXAMPLE

Read a character and if it's an uppercase letter, display it.

Pseudocode

If ('A' >= character) and (character <='Z')

Then

Display character

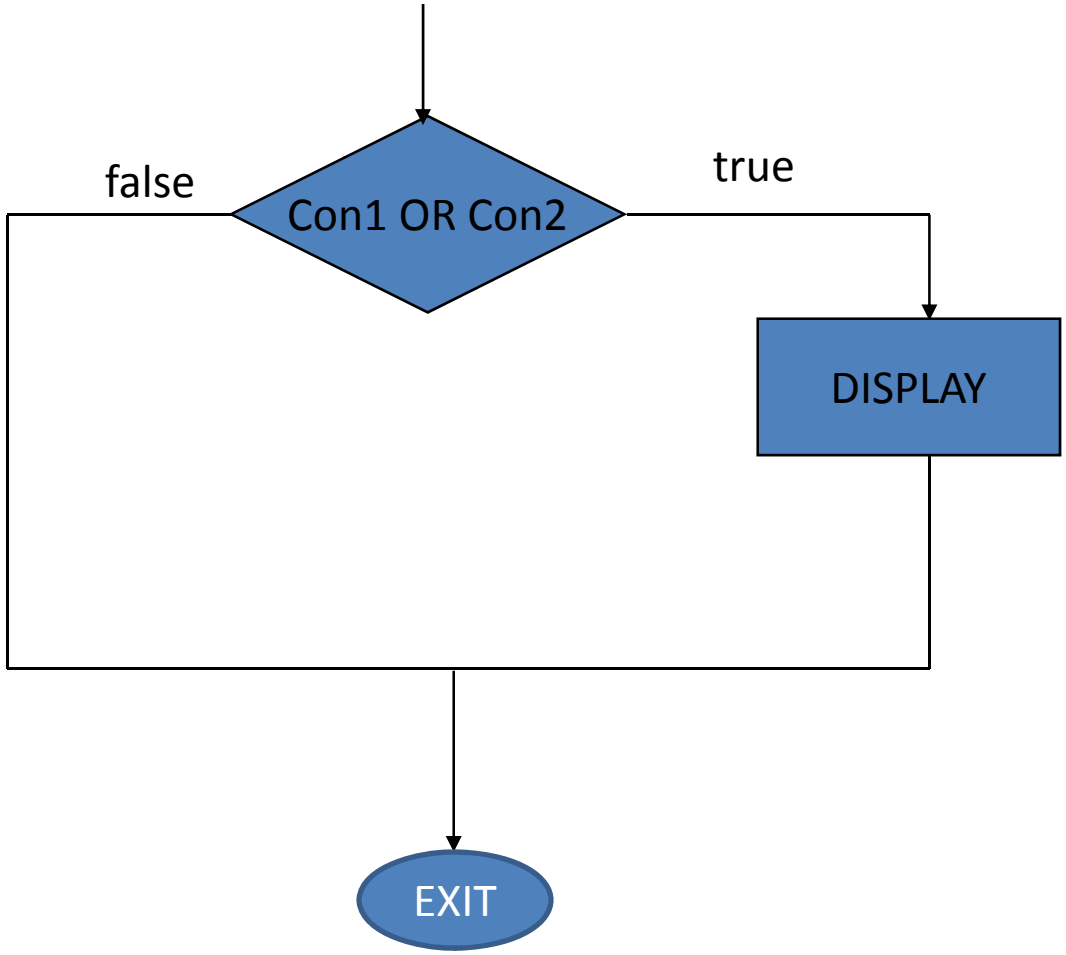
End_if

Code:

```
MOV AH,1
INT  21H
CMP AL,'A'
JL END_IF
CMP AL,'Z'
JG END_IF
MOV DL,AL
MOV AH,2
INT 21H
END_IF:
```

OR conditions

- If condition_1 OR condition_2 is true then an OR conditions is true.
- if both conditions are false, then the whole thing is false.



Example

Read a character. If it's "y" or "Y", display it; otherwise, terminate the program.

Pseudocode

if (character = "y" or character= "Y")
Then
Display character
Else
Terminate the program
End_if

Code:

```
MOV  AH,1
INT  21H
CMP AL, 'y'
JE THEN
CMP AL,'Y'
JE THEN
JMP ELSE_
THEN:
 MOV AH,2
 MOV DL,AL
 INT 21H
 JMP END_IF
  ELSE_:
END_IF:
```

THANK YOU