Assembly Practice-01

Basics
Branches
Array

Recap: Code Structure

```
.MODEL SMALL
.STACK 100H
.DATA
;VARS AND ARRAYS
.CODE
MAIN PROC
      ;CODE HERE
MAIN ENDP
      END MAIN
```

Problem-01

Our first program will read a alphabet character from the keyboard and display it **at the beginning** of the next line.

- **Step-01: Prompt a Message for Input**

```
MOV AH, 09h
LEA DX, STR_VAR
INT 21h
```

- **Step-02: Input a character**

```
MOV AH, 01h
INT 21h
```

- **Step-03: Store the Character**

```
MOV VAR, AL
```

- **Step-04: Output new line**

```
MOV AH, 02h
MOV DL,0DH      ;carriage return (go to beginning)
INT 21h
MOV DL,0AH      ;line feed (move by one line)
INT 21h
```

- **Step-05: Output the character**

```
MOV DL,VAR
INT 21h
```

Problem-02

Edit the previous program, read a lowercase alphabet and display the alphabet in uppercase.

$$A \rightarrow 41h \quad \leftrightarrow \quad Z \rightarrow 5Ah$$

$$a \rightarrow 61h \quad \leftrightarrow \quad z \rightarrow 7Ah$$

- **Step-05: Output the character**

```
MOV DL,VAR
SUB DL, 20h
INT 21h
```

Branching Structures

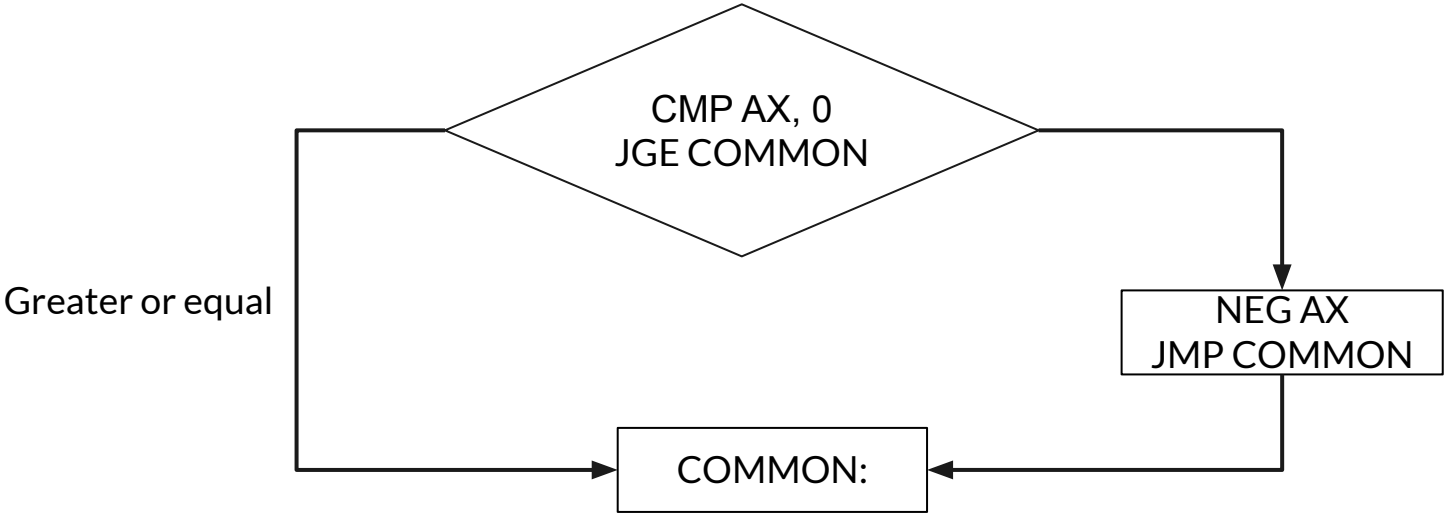
- In C we have **`if-else`** and **`conditions`**. But in Assembly we have **`JXX/ JMP`** and **`CMP`**.
- Let's look at this pdf to see the varieties of Jump instruction.

Problems: Branching Structures

1. **If Then:** Replace the number in AX by its absolute value.
2. **If Then Else:** Suppose AL and BL contain extended ASCII characters. Display the one that comes first in the character sequence.
3. **Multiple Case:** 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.
4. **Multiple Case Common Branch:** If AL contains 1 or 3, display "o"; if AL contains 2 or 4, display "e".
5. **AND Relationship:** Read a character, and if it's an uppercase letter, display it.
6. **OR Relationship:** Read a character. If it's "y" or "Y", display it; otherwise, terminate the program.

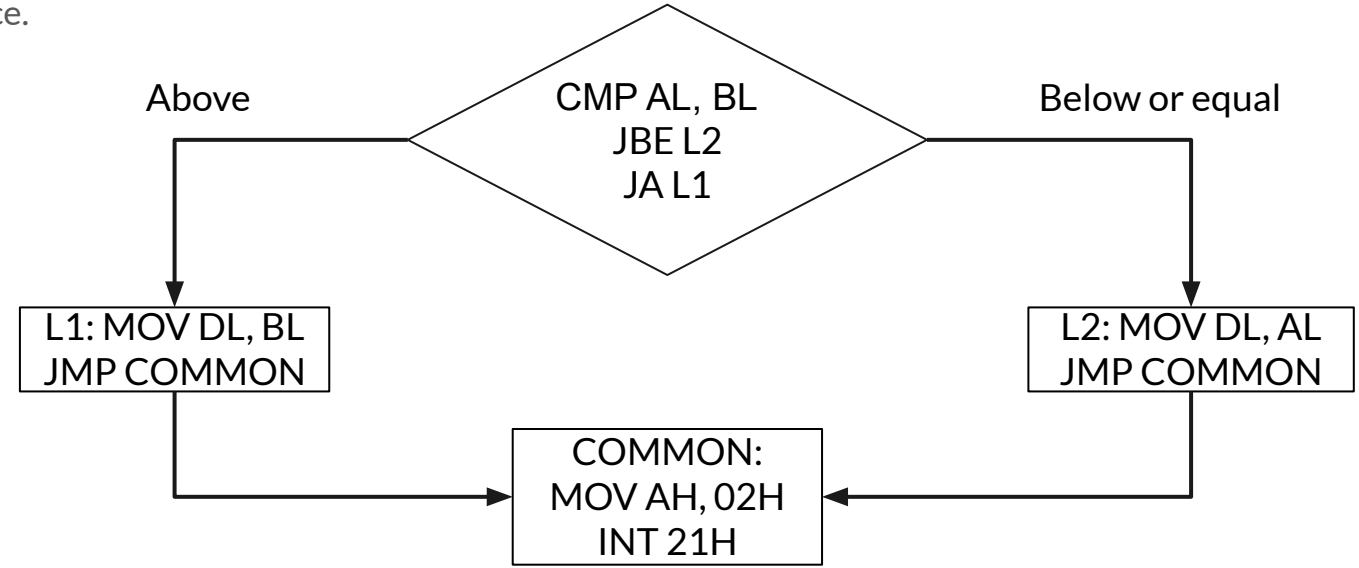
If Then

Replace the number in AX by its absolute value.



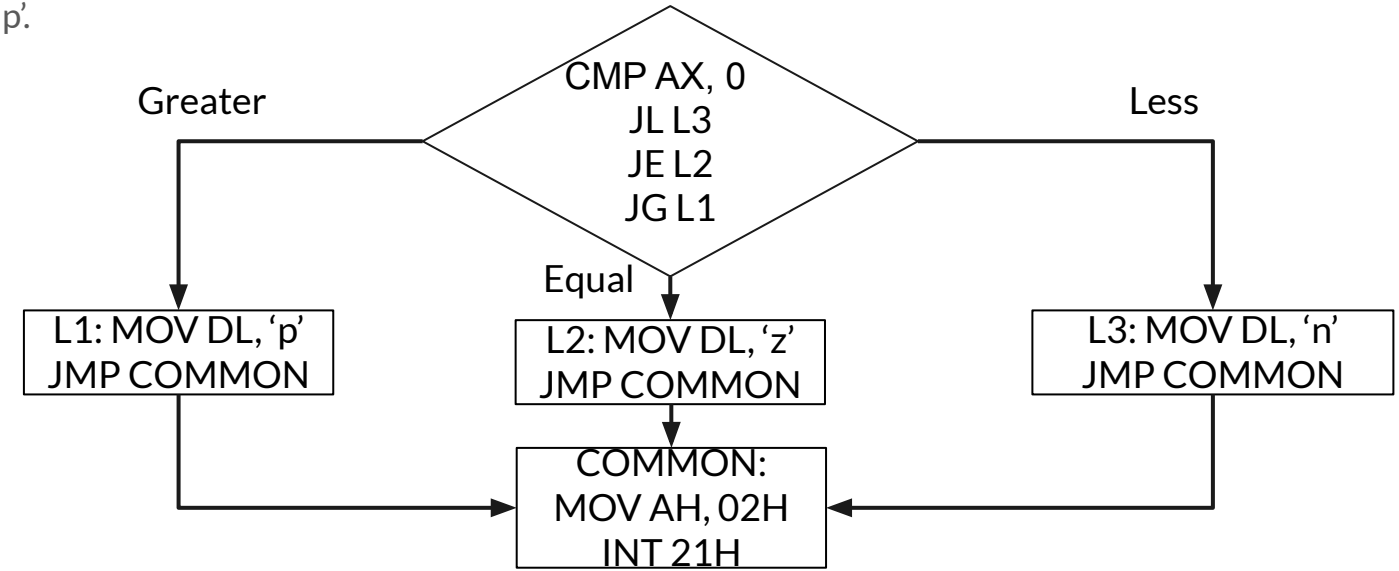
If Then Else

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



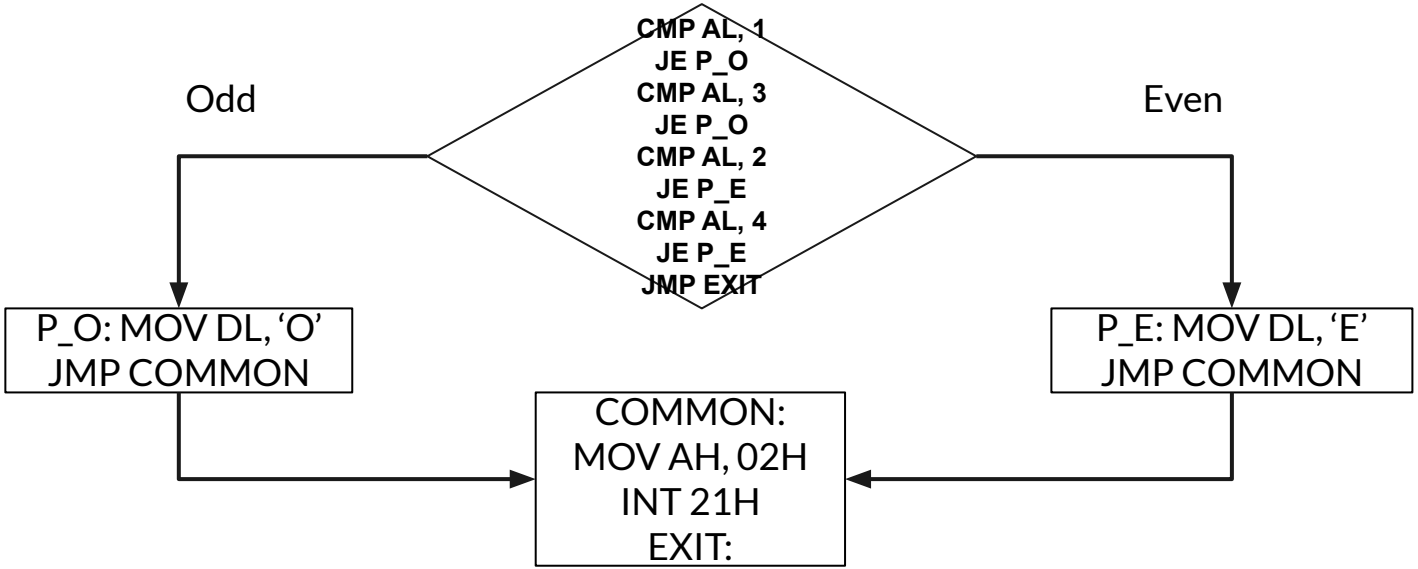
Multiple Case

If AX contains a negative number, display ‘n’; if AX contains 0, display ‘z’; if AX contains a positive number, display ‘p’.



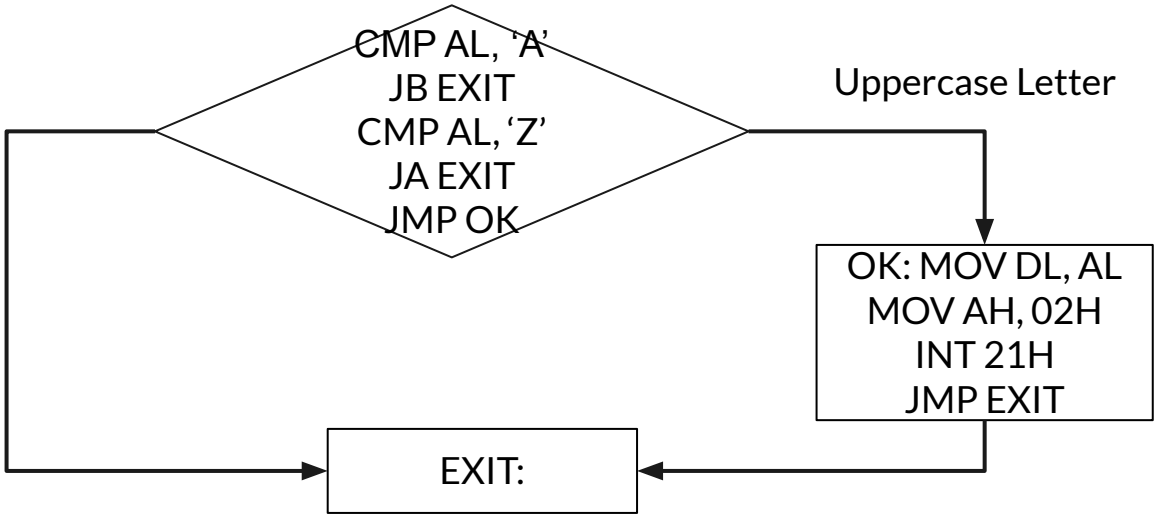
Multiple Case Common Branch

If AL contains 1 or 3, display "o"; if AL contains 2 or 4, display "e".



AND Relationship

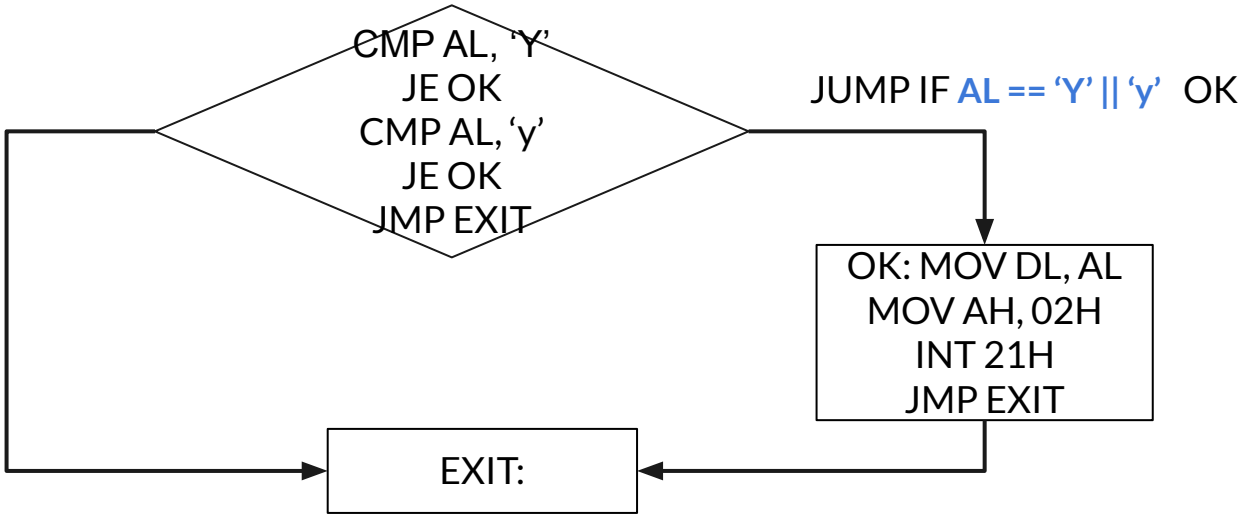
Read a character, and if it's an uppercase letter, display it. **Relation→ 'A' ≤ AL ≤ 'Z'**



Not uppercase letter:
'A' > AL > 'Z'==1

OR Relationship

Read a character. If it's "y" or "Y", display it; otherwise, terminate the program. **Relation→ AL == ‘Y’ || ‘y’**



Practice

Write a code to input signed number in A and B. And display the biggest one in the terminal. Assume A and B both are byte.

Simple Input-Output:

Enter A: 5

Enter B: 7

The larger value is 7.

Thank You

QUESTION?