Green University of Bangladesh

Department of Computer Science and Engineering (CSE)

Faculty of Sciences and Engineering

Semester: (Fall, Year: 2025), B.Sc. in CSE (Day)

Lab Report NO: 02
Course Title: Microprocessors, Microcontrollers, and Embedded System Lab
Course Code: CSE 304
Section: 232-D1

Lab Experiment Name: Implementation of conditional statement using assembly language

Student Details

Name		ID
1.	Ashab Uddin	232002274

Lab Date : 22/10/2025
Submission Date : 29/10/2025
Course Teacher's Name : Jarin Tasnim Tonvi

Lab Report Status

Marks: ………………………………… Signature: ....
Comments: .... Date: ....

TITLE: Implementation of conditional statement using assembly language.

1. INTRODUCTION

This lab focuses on implementing conditional statements in assembly language to understand how microprocessors make decisions based on flag registers. Unlike high-level languages that use simple if–else constructs, assembly requires explicit handling of status flags and jump instructions (such as JE, JNE, JG, etc.) to control program flow. Through practical exercises, this lab helps students grasp conditional branching, logical comparisons, and execution control at the hardware level.

2. OBJECTIVES

- To understand the concept of conditional branching and flag register operations in assembly language.
- To implement if, if–else, and case structures using conditional and unconditional jump instructions.
- assembly language. To write optimized and efficient 16-bit code.

3. PROCEDURE

Problem-1: Find out the largest number between two numbers using assembly language.

1. The program first displays a message asking the user to enter the first number (0–9).
2. It reads the input as an ASCII character and converts it to a numeric value by subtracting 30H, then stores it in register BL.
3. It prompts for the second number, reads it, converts it similarly, and stores it in BH.
4. The two numbers (BL and BH) are compared using the CMP instruction.
    - If both are equal → it displays “Both numbers are equal”.
    - If the first number is greater → it displays “The largest number is:” followed by the first number.
    - Otherwise, it displays the second number as the largest.
5. The result is printed using INT 21H (AH = 9) and INT 21H (AH = 2) for text and character display.
6. Finally, the program exits using INT 21H (AH = 4CH).

Problem-2: Take a number input from user, check whether the given number is divisible by 5 or not.

Input:

- The program prompts the user to enter a single-digit number (0–9).
- The entered ASCII character is read through INT 21H (AH = 1) and converted into its numeric value by subtracting 30H.
- This numeric value is stored in register AL.

Calculation:

- The value in AL is divided by 5 using the DIV BL instruction, where BL = 5.
- After division, the remainder is automatically stored in register AH.
- If the remainder (AH) = 0, the number is exactly divisible by 5; otherwise, it is not divisible.

Decision & Output:

- The program compares AH with 0 using the CMP instruction.
- If equal, it displays the message “Divisible by 5.”
- If not equal, it displays “Not divisible by 5.”
- The message is printed to the console using INT 21H (AH = 9) with the corresponding string offset in DX.

4. IMPLEMENTATION

Problem-1: Source Code:

```
.MODEL SMALL
.STACK 100H

.DATA
   STR1 DB 'Enter first number (0-9): $'
   STR2 DB 0DH,0AH, 'Enter second number (0-9): $'
   STR3 DB 0DH,0AH, 'The largest number is: $'
   STR4 DB 0DH,0AH, 'Both numbers are equal: $'
   NL   DB 0DH,0AH,'$'

.CODE
MAIN PROC
   ; init DS
   MOV AX, @DATA
   MOV DS, AX

   ; ----- Read first digit -----
   MOV DX, OFFSET STR1
   MOV AH, 9
   INT 21H

READ_FIRST:
   MOV AH, 1           ; read char
   INT 21H
   CMP AL, '0'
   JB  READ_FIRST      ; not a digit, read again
   CMP AL, '9'
   JA  READ_FIRST      ; not a digit, read again
   SUB AL, '0'        ; ASCII -> value 0..9
   MOV BL, AL         ; BL = first number
```

```
    ; newline
    MOV DX, OFFSET NL
    MOV AH, 9
    INT 21H

    ; ----- Read second digit -----
    MOV DX, OFFSET STR2
    MOV AH, 9
    INT 21H

READ_SECOND:
    MOV AH, 1
    INT 21H
    CMP AL, '0'
    JB  READ_SECOND
    CMP AL, '9'
    JA  READ_SECOND
    SUB AL, '0'
    MOV BH, AL          ; BH = second number

    ; newline
    MOV DX, OFFSET NL
    MOV AH, 9
    INT 21H

    ; ----- Compare and print result -----
    CMP BL, BH
    JE  EQUAL_CASE
    JG  FIRST_IS_LARGER

    ; BH is larger
    MOV DX, OFFSET STR3
    MOV AH, 9
    INT 21H

    MOV DL, BH          ; value 0..9
    ADD DL, '0'         ; to ASCII
    MOV AH, 2
    INT 21H
    JMP DONE

FIRST_IS_LARGER:
    MOV DX, OFFSET STR3
    MOV AH, 9
    INT 21H

    MOV DL, BL
    ADD DL, '0'
    MOV AH, 2
    INT 21H
    JMP DONE

EQUAL_CASE:
    MOV DX, OFFSET STR4
    MOV AH, 9
    INT 21H

    MOV DL, BL
    ADD DL, '0'
    MOV AH, 2
    INT 21H

DONE:
    ; final newline
    MOV DX, OFFSET NL
    MOV AH, 9
    INT 21H

    ; exit
    MOV AH, 4CH
    INT 21H
MAIN ENDP
END MAIN
```

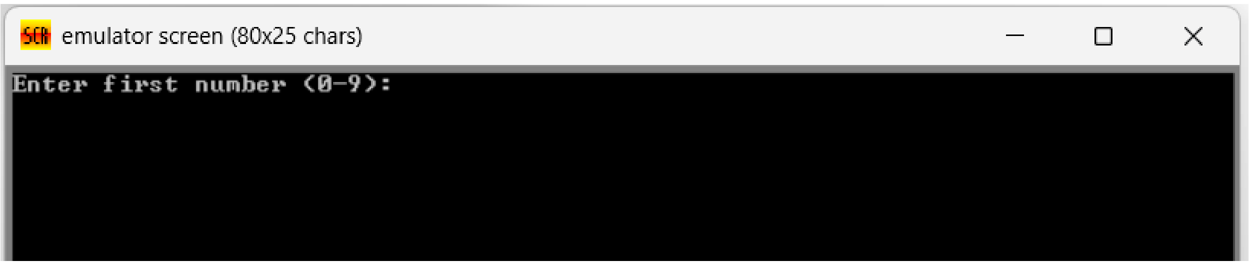
Problem-2 Source Code:

```
.MODEL SMALL
.STACK 100H

.DATA
   STR1 DB 'Enter a number (0-9): $'
   STR2 DB 0DH,0AH,'The number is divisible by 5.$'
   STR3 DB 0DH,0AH,'The number is NOT divisible by 5.$'

.CODE
MAIN PROC
   ; Initialize DS
   MOV AX, @DATA
   MOV DS, AX

   ; Ask for input
   MOV DX, OFFSET STR1
   MOV AH, 9
   INT 21H

   ; Read one character (digit)
   MOV AH, 1
   INT 21H
   SUB AL, 30H      ; Convert ASCII to number 0–9

   ; Divide the number by 5
   MOV AH, 0
   MOV BL, 5
   DIV BL       ; AL = quotient, AH = remainder

   ; Check remainder
   CMP AH, 0
   JE DIVISIBLE
   JNE NOT_DIVISIBLE

DIVISIBLE:
   MOV DX, OFFSET STR2
   MOV AH, 9
   INT 21H
   JMP EXIT

NOT_DIVISIBLE:
   MOV DX, OFFSET STR3
   MOV AH, 9
   INT 21H

EXIT:
   MOV AH, 4CH
   INT 21H

MAIN ENDP
END MAIN
```

5. OUTPUT

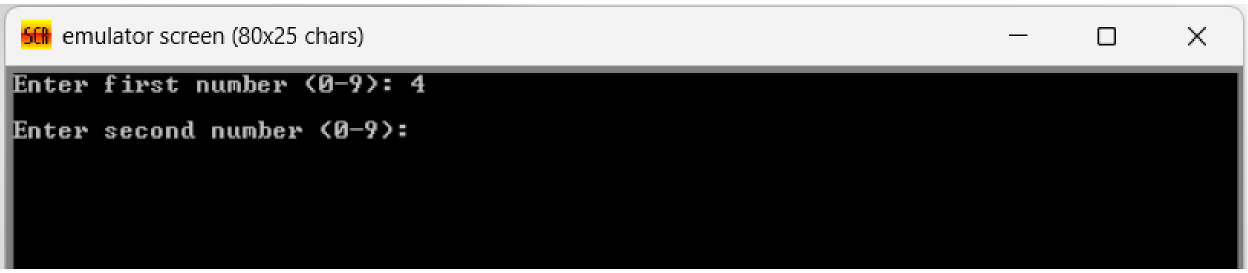
Output Problem-1

Step-1



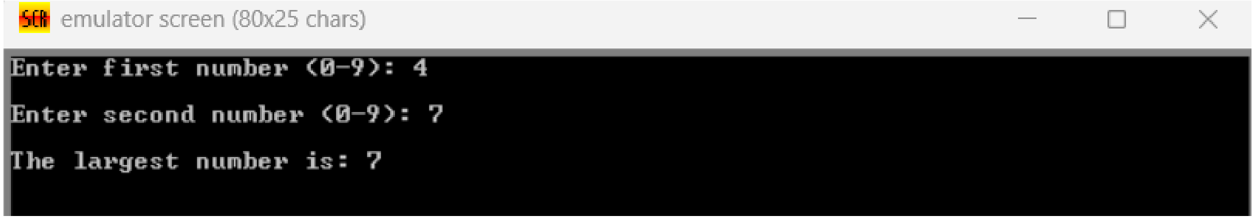
```
Enter first number (0-9):
```

Step-2



```
Enter first number (0-9): 4

Enter second number (0-9):
```

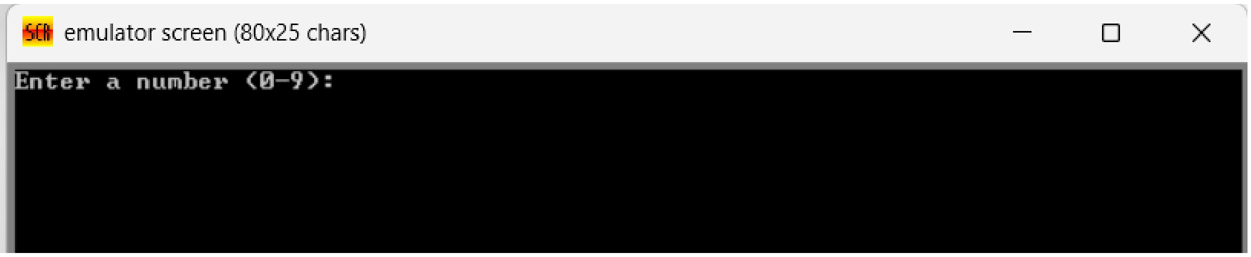
Final Step



```
Enter first number (0-9): 4

Enter second number (0-9): 7

The largest number is: 7
```

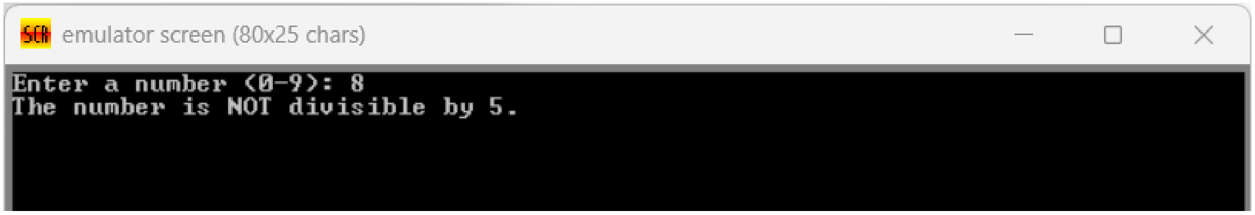
Output Problem-2

Step-1



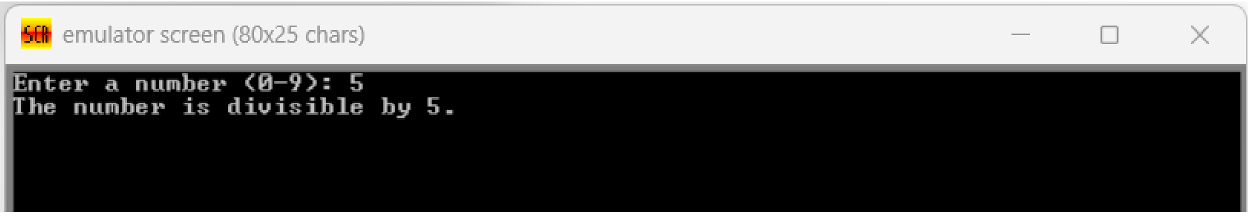
```
Enter a number (0-9):
```

Step-2



```
Enter a number (0-9): 8
The number is NOT divisible by 5.
```

Step-3



```
Enter a number (0-9): 5
The number is divisible by 5.
```

6. ANALYSIS AND DISCUSSION

This lab focuses on two conditional problems in assembly language: identifying the larger of two numbers and checking whether a number is divisible by 5. In both cases, the logic depends on flag registers and conditional jump instructions. For the first problem, the program compares two inputs using the CMP instruction and uses conditional jumps like JAE or JB to determine and display the greater value. For the second problem, the DIV instruction divides the input number by 5, and the remainder stored in the AH register determines divisibility if the remainder is zero, the number is divisible by 5.

These exercises strengthened the understanding of decision-making at the machine level. By implementing comparisons and conditional branching manually, it became clearer how high-level logic like if–else operates internally through flag manipulation. Overall, the lab enhanced practical skills in assembly programming, arithmetic operation handling, and logical control flow within the processor.

.