UITS

UNIVERSITY OF INFORMATION TECHNOLOGY AND SCIENCES

Assignment on

Lab Report- 02

Course Title

Microprocessor and MicroControllers

Course Code

CSE 360

Submitted by

KM Jakaria

Section: 6A

Batch: 53

ID: 043231000510103 7

Submitted to

Md. Ismail

Lecturer

Department of CSE

UITS

Date of Submission

27 Aug, 2025

Problem No: 01

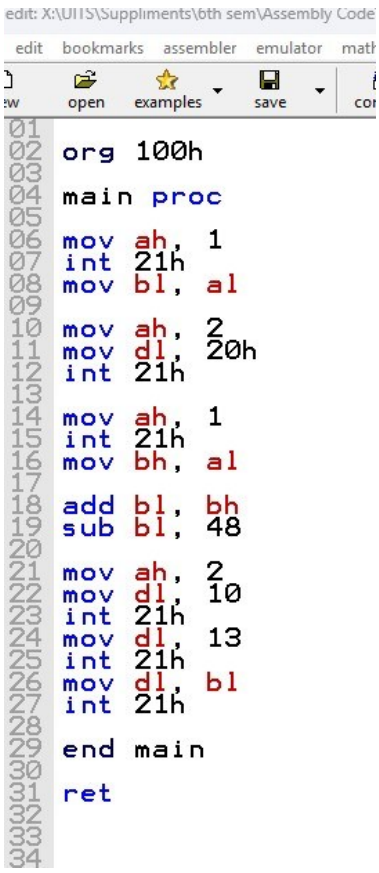
Experiment No: 01

Experiment Name: Take two numbers from user and print their summation as an output

Process:

The operations carried out by the program are consecutive and linear. For input and output, it first makes use of the DOS interrupt service INT 21H. The input function AH=01H is used to read a character from the user in the first phase, and the character is automatically echoed to the screen. After being saved in the AL register, this character is then transferred to BL for short-term storage. The program then uses the output function AH=02H to separate the two inputs and output a space character (ASCII 20H). The second character, which is saved in BH, is read using the same input function once more. The second character, which is saved in BH, is read using the same input function once more. To convert the total from an ASCII-based value to the correct numerical result in ASCII form, the arithmetic operation is carried out by adding the values in BL and BH, then subtracting 48. Prior to handing over control back to the operating system, the program shows the result character after finally producing a newline sequence (carriage return and line feed).

Implementation:



Result:

When the program is run, it interacts with the user in the following way: the user is prompted to enter a first digit (e.g., '3'), which is immediately displayed on the screen due to the echo feature of the input function. The program then outputs a space, and the user enters a second digit (e.g., '5'), which is also echoed. After the user presses Enter, the program calculates the sum (e.g., 3 + 5 = 8), moves the cursor to a new line, and prints the result. The final output on the screen appears as: When the program is run, it interacts with the user in the following way: the user is prompted to enter a first digit (e.g., '3') Which, thanks to the input function's echo functionality, is shown on the screen right away. The user inputs a second digit (such as "5"), which is likewise echoed, after the computer generates a space. The application computes the sum (for example, 3 + 5 = 8), advances the cursor to a new line, and prints the outcome whenever the

user hits Enter. On the screen, the finished product looks like this:

Conclusion:

The application effectively illustrates 8086 assembly language input and output processes. A simple way for character input and output is provided by the INT 21H interrupt with the proper function codes in the AH register. When utilizing function 01H for input, the character is automatically echoed to the output, which explains why we see the input character twice - once from the automatic echo and once from our explicit output command. Our comprehension of fundamental I/O operations in 8086 assembly programming is validated by this experiment.

Problem No: 02

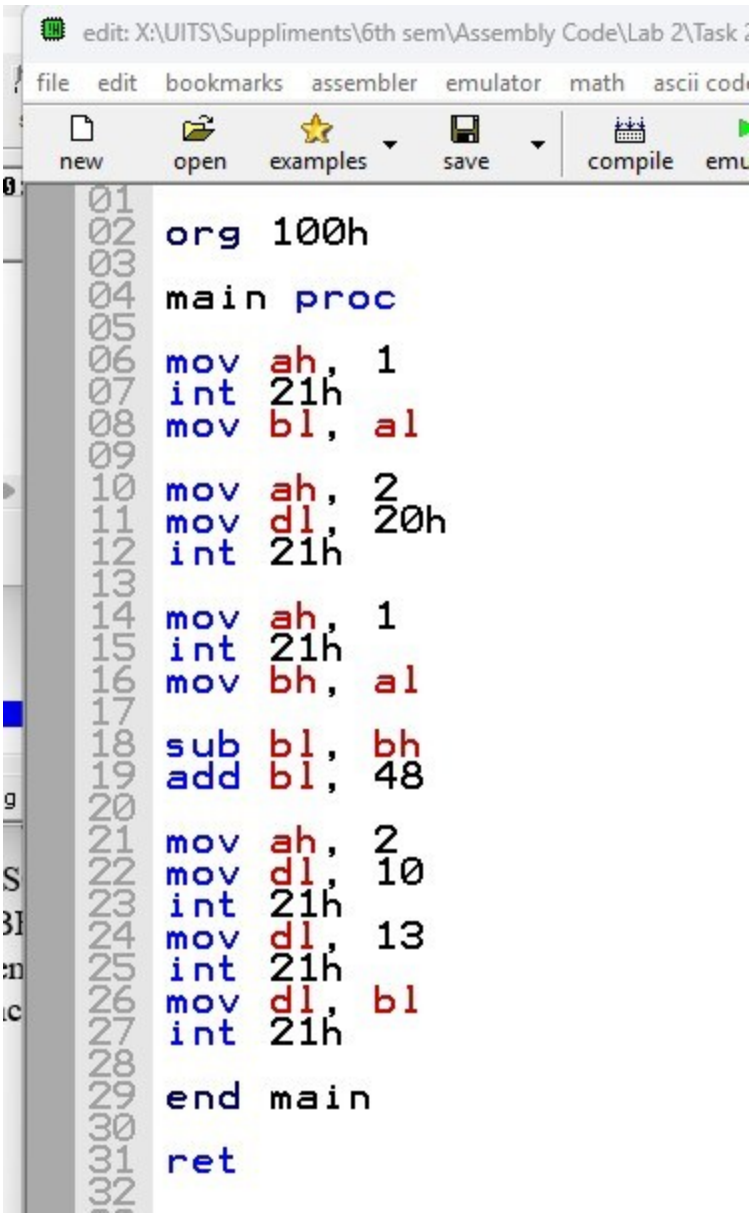
Experiment No: 02

Experiment Name: Take two numbers from user and print their difference as a output

Process:

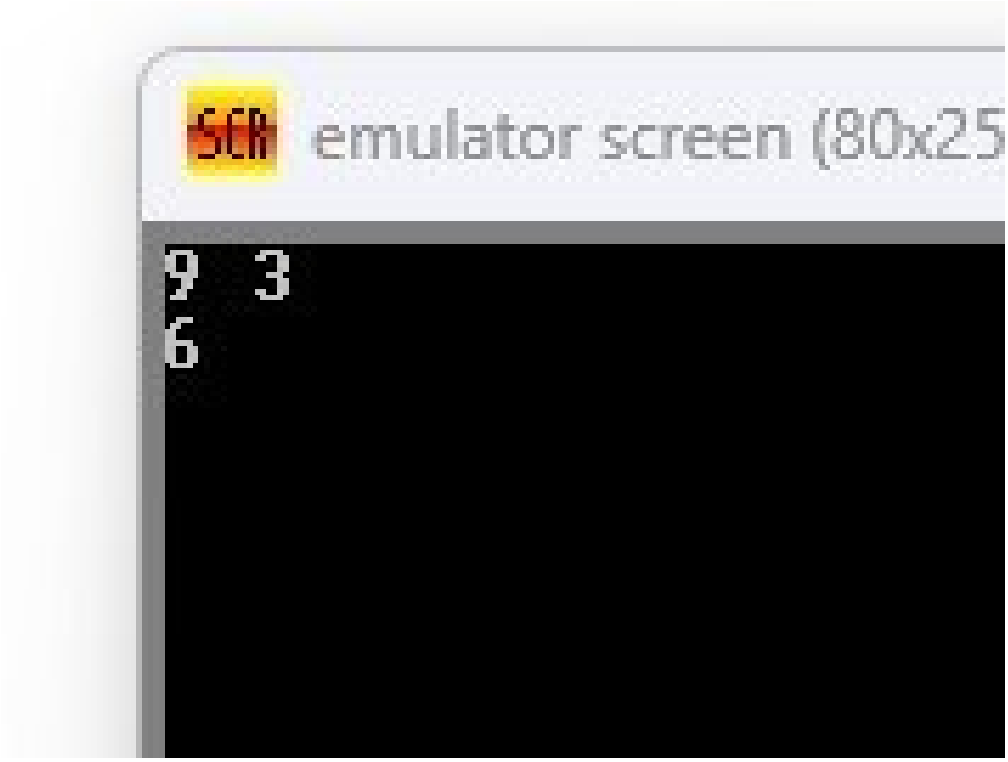
An organized series of actions is used by the program to process user input, carry out arithmetic subtraction, and provide formatted output. It starts by setting up a DOS.COM file's application environment. The DOS interrupt INT 21H with AH=01H is used to read a character from the user in the first stage. The input is automatically echoed to the screen, and the ASCII value is stored in the AL register. After then, this value is transferred to BL for short-term storage. The program prints a space character (ASCII 20H) using INT 21H with clear formatting. AH is equal to 02H. The second character is saved in BH and is read using the same input mechanism. The ASCII values are handled directly by the arithmetic operation, which subtracts the value in BH from BL. In order to convert the result back to a legal ASCII character that represents the numerical difference, the computer adds 48 since subtracting ASCII values can provide outputs that are outside the range of permissible digits. Before handing back control to the operating system, the software shows the result character and emits a newline sequence (carriage return and line feed) to advance to the next line.

Implementation:



Result:

When the program runs, it communicates with the user in this way: the user inputs the first number (for example, "9"), and the input function's echo functionality instantly displays it on the screen. After that, the software outputs a space character to provide visual distinction. Input of the second digit (for example, '3') is likewise echoed. When the user hits Enter, the application determines the difference (for example, 9 - 3 = 6), advances the cursor to a new line, and outputs the outcome. On the screen, the finished product looks like this:



Conclusion:

In 8086 assembly code, the program effectively illustrates input and output actions using formatting and arithmetic subtraction. It manages user interactions efficiently by using DOS interrupt INT 21H with function codes 01H for input and 02H for output. The output is easier to read when formatting characters like spaces and newline sequences are used. In order to accommodate for any negative outcomes, the adjustment step (adding 48 after subtraction) guarantees that the result is accurately transformed from ASCII arithmetic back to a displayable ASCII character. This exercise demonstrates a practical grasp of data manipulation, interrupt management, and basic I/O handling in 8086 assembly programming, laying the groundwork for more intricate programs including formatted output and arithmetic operations.