Green University of Bangladesh
Department of Computer Science and Engineering (CSE)

Faculty of Sciences and Engineering

Semester: (Fall, Year:2023), B.Sc. in CSE (Day)

LAB REPORT NO: 01

Course Title: Compiler Lab

Course Code: CSE-304 Section:213-D1

Lab Experiment Name: Introduction to assembly program structure and various arithmetic operations.

Student Details

Name		ID
1.	Irteja Mahmud	213902016

Lab Date : 12-10-2023

Submission Date : 19-10-2023

Course Teacher's Name : Sudip Ghoshal

Lab Report Status

Marks: …… **Signature:.....**

Comments:..... **Date:.....**

1. TITLE OF THE LAB EXPERIMENT

Introduction to assembly program structure and various arithmetic operations.

2. OBJECTIVES/AIM

The objective of this program is to know about assembly program structure and various arithmetic operations.

3.PROCEDURE / ANALYSIS / DESIGN

Problem 1: Take a double-digit number input from the user.

Step 1: Initialize the program and memory.
- Set the program's origin to 100h.
- Define the memory model as "small."
- Allocate a stack of 100h bytes.

Step 2: Define data section.

Step 3: Start of the code section.
- Begin the 'code' section.
- Define the 'main' procedure.

Step 4: Setup data segment and input prompt.
- Load the data segment address into AX from '@DATA'.
- Set DS (data segment) to the value in AX.
- Set AH to 9 (print string function).
- Load the offset address of 'msg' into DX.
- Trigger an interrupt 21H to display the message.

Step 5: Read a character from the user (1st digit).
- Set AH to 1 (read character function).
- Trigger interrupt 21H to read a character from the user.
- Store the read character in BL.

Step 6: Read a character from the user (2nd digit).
- Set AH to 1 (read character function).
- Trigger interrupt 21H to read a character from the user.
- Store the read character in BH.

Step 7: Output newline characters.
- Set AH to 2 (display character function).
- Load DL with ASCII code 13 (carriage return).
- Trigger interrupt 21H to display a carriage return.

Step 8: Output newline characters.
- Set AH to 2 (display character function).
- Load DL with ASCII code 10 (line feed).
- Trigger interrupt 21H to display a line feed.

Step 9: Display the 1st digit entered by the user.
- Set AH to 2 (display character function).
- Load DL with the value in BL.
- Trigger interrupt 21H to display the character.

Step 10: Display the 2nd digit entered by the user.
- Set AH to 2 (display character function).
- Load DL with the value in BH.
- Trigger interrupt 21H to display the character.

Step 11: End of the main procedure.
- End the 'main' procedure using 'main endp'.

Step 12: End of the program.
- Mark the end of the program using 'end main'.

Problem 2: Convert 260°C to Fahrenheit using the following expression and store in a F variable: °F = °C 9/5 + 32

Step 1: Initialize the program and memory.

Step 2: Define data section.
- Declare variables:

Step 3: Start of the code section.
- Begin the 'code' section.
- Define the 'main' procedure.

Step 4: Setup data segment and input prompt.
- Load the data segment address into AX from '@DATA'.
- Set DS (data segment) to the value in AX.
- Set AH to 9 (print string function).
- Load the offset address of 'msg' into DX.
- Trigger an interrupt 21H to display the message.

Step 5: Display the initial temperature in Celsius.
- Set AH to 2 (display string function).
- Load DX with the address of 'tc'.

- Trigger interrupt 21H to display the value of 'tc'.

Step 6: Convert Celsius to Fahrenheit.
- Perform the Fahrenheit conversion using the formula: f = c * 9 / 5 + 32.
- Store the result in 'tf'.

Step 7: Output a newline and carriage return.
- Set AH to 9 (print string function).
- Load DX with the address of 'next'.
- Trigger interrupt 21H to display a newline and carriage return.

Step 8: Display the message for Fahrenheit temperature.
- Set AH to 9 (print string function).
- Load DX with the address of 'msg1'.
- Trigger interrupt 21H to display the message.

Step 9: Display the converted temperature in Fahrenheit.
- Set AH to 2 (display string function).
- Load DX with the address of 'tf'.
- Trigger interrupt 21H to display the value of 'tf'.

Step 10: End of the main procedure.
- End the 'main' procedure using 'main endp'.

Step 11: End of the program.
- Mark the end of the program using 'end main'.

Problem 3: Convert 1000 °F (Fahrenheit) to °C (Celsius) using the following expression and store in a C variable: °C = (°F - 32) 5/9

Step 1: Initialize the program and memory.

Step 2: Define data section.
- Declare variables:

Step 3: Start of the code section.
- Begin the 'code' section.
- Define the 'main' procedure.

Step 4: Setup data segment and input prompt.
- Load the data segment address into AX from '@DATA'.
- Set DS (data segment) to the value in AX.
- Set AH to 9 (print string function).
- Load the offset address of 'msg1' into DX.
- Trigger an interrupt 21H to display the message.

Step 5: Display the initial temperature in Fahrenheit.
- Set AH to 2 (display string function).
- Load DX with the address of 'tf'.
- Trigger interrupt 21H to display the value of 'tf'.

Step 6: Convert Fahrenheit to Celsius.
- Perform the Celsius conversion using the formula: c = (f - 32) * 5 / 9.
- Store the result in 'tc'.

Step 7: Output a newline and carriage return.
- Set AH to 9 (print string function).
- Load DX with the address of 'next'.
- Trigger interrupt 21H to display a newline and carriage return.

Step 8: Display the message for Celsius temperature.
- Set AH to 9 (print string function).
- Load DX with the address of 'msg'.
- Trigger interrupt 21H to display the message.

Step 9: Display the converted temperature in Celsius.
- Set AH to 2 (display string function).
- Load DX with the address of 'tc'.
- Trigger interrupt 21H to display the value of 'tc'.

Step 10: End of the main procedure.
- End the 'main' procedure using 'main endp'.

Step 11: End of the program.
- Mark the end of the program using 'end main'.

4.IMPLEMENTATION

Problem 1:

```
org 100h
.model small
.stack 100h
.data
   var dw 100
   msg dw "Enter A Number: $"
.code
    main proc

    MOV AX, @DATA
    MOV DS, AX

    MOV AH,9
    LEA DX,MSG
    INT 21H

    MOV AH,1
    INT 21H
    MOV BL,AL

    MOV AH,1
    INT 21H
    MOV BH,AL

    MOV AH,2
    MOV DL,013
    INT 21H

    MOV AH,2
    MOV DL,010
    INT 21H

    MOV AH,2
    MOV DL,BL
    INT 21H

    MOV AH,2
    MOV DL,BH
    INT 21H


    main endp
end main
```

Problem 2:

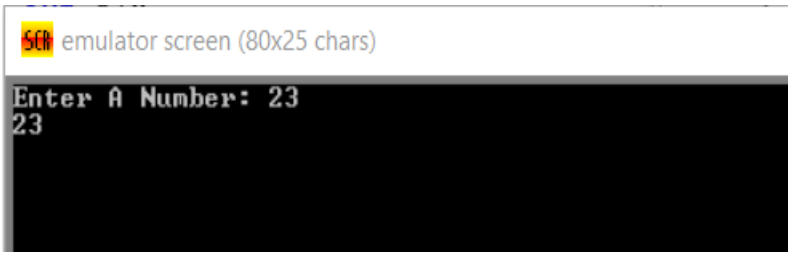
```
org 100h
.model small
.stack 100h
.data
   var dw 100
   msg dw "Temparature In Celcius: $"
   msg1 dw "Temparature In Fahrenheit: $"
   tc dw 260     ; t celsius.
   tf dw 0       ; t fahrenheit.
   next db 013,010"$"
.code
    main proc

    MOV AX, @DATA
    MOV DS, AX

    MOV AH,9
    LEA DX,MSG
    INT 21H

    MOV AH,2
    MOV DX,TC
    INT 21H

    ; convert celsius to fahrenheit according
    ; to this formula: f = c * 9 / 5 + 32

    MOV cx, tc
    MOV ax, 9
    IMUL cx
    MOV cx, 5
    IDIV cx
    ADD ax, 32
    MOV tf, ax

    MOV AH,9
    LEA DX,next
    INT 21H

    MOV AH,9
    LEA DX,MSG
    INT 21H



    MOV AH,2
    MOV DX,Tf
    INT 21H

    main endp
 end main
```

Problem 3:

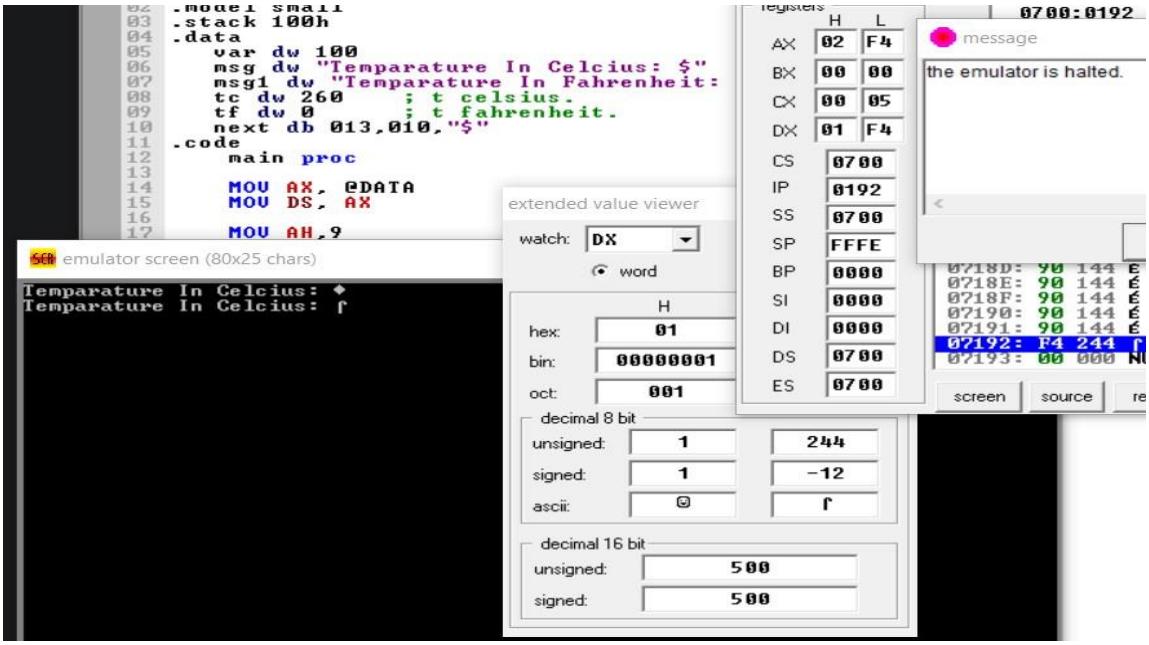
```
org 100h
.model small
.stack 100h
.data
   var dw 100
   msg dw "Temparature In Celcius: $"
   msg1 dw "Temparature In Fahrenheit: $"
   tc dw 0     ; t celsius.
   tf dw 1000      ; t fahrenheit.
   next db 013,010,"$"
.code
    main proc

    MOV AX, @DATA
    MOV DS, AX

    MOV AH,9
    LEA DX,MSG1
    INT 21H

    MOV AH,2
    MOV DX,Tf
    INT 21H

    ; convert fahrenheit to celsius according
    ; to this formula: c = (f - 32) * 5 / 9

    MOV cx, tf
    SUB cx, 32
    MOV ax, 5
    IMUL cx
    MOV cx, 9
    IDIV cx
    MOV tc, ax

    MOV AH,9
    LEA DX,next
    INT 21H

    MOV AH,9
    LEA DX,MSG
    INT 21H


    MOV AH,2
    MOV DX,Tc
    INT 21H

    main endp
 end main
```

5.TEST RESULT / OUTPUT

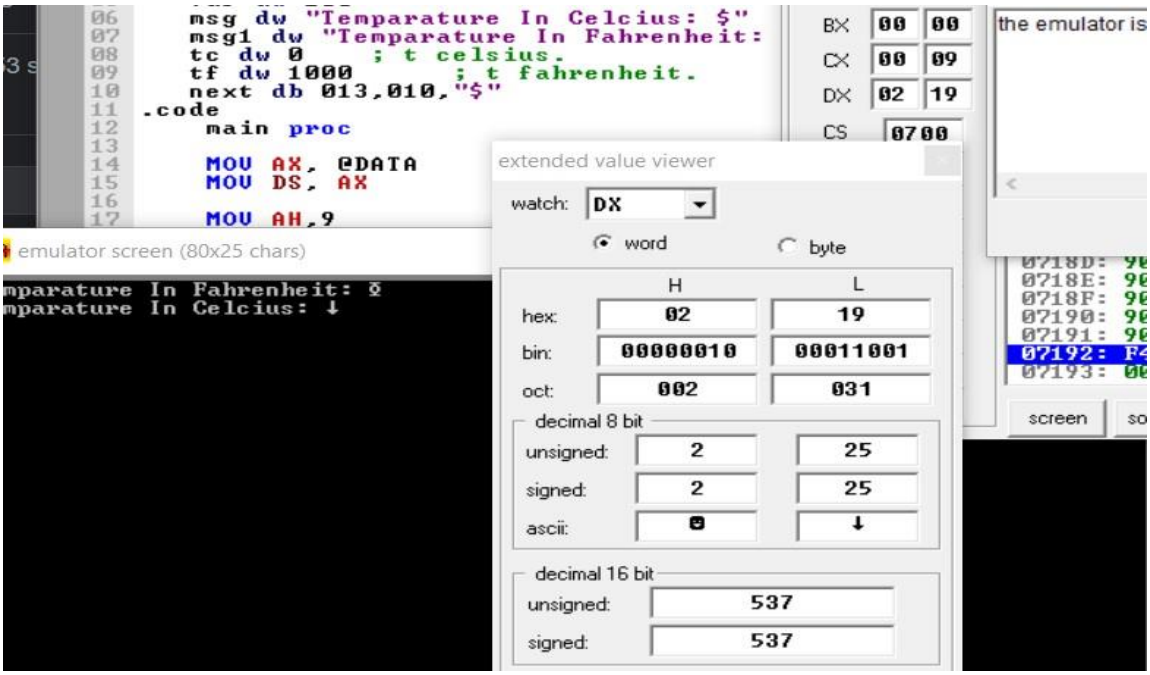
Problem 1 Output:



Problem 2 Output:



Problem 3 Output:



6.ANALYSIS AND DISCUSSION

In this lab report, the provided assembly code problem is a simple program that demonstrates temperature conversion between Fahrenheit to Celsius. The program is written in 8086 assembly language and is run in emu8086. The program begins by setting the origin point to 100h, indicating the memory location where it will start executing. It defines a small memory model and allocates a stack of 100h bytes for program operations. The data section defines various data items. The code section starts by defining the 'main' procedure. It initializes the data segment and displays a prompt message.

7. SUMMARY:

This code is designed for execution in a emu8086 environment, and it can serve as an educational example of data manipulation and arithmetic operations in assembly language. It demonstrates the conversion formula and the use of data segments and interrupts for input and output operations.