GREEN UNIVERSITY OF BANGLADESH

Department of Computer Science and Engineering (CSE)

Faculty of Sciences and Engineering
Semester: (Fall 2025), B.Sc. in CSE (Day)

Lab Report #3

Course Title: Microprocessors and microcontrollers

Course Code: CSE 304 **Section: 232_D4**

Lab Experiment Name: Implement a loop to find out the summation of 1^2 + 2^2 + 3^2 + ... + n^2. You can take n from user as an input. The code should be able to output a 3 digit number.

Student Details

Name		ID
1.	Rafi Shah	232002168

Submission Date : 05-11-2025
Course Teacher's Name : Jarin Tasnim Tonvi

[For Teachers use only: Don't Write Anything inside this box]

Project Report Status
Marks: …… Signature:.....
Comments:..... Date:.....

1. **TITLE OF THE LAB REPORT EXPERIMENT:**
   Implement a loop to find out the summation of 1^2 + 2^2 + 3^2 + ... + n^2. You can take n from user as an input. The code should be able to output a 3 digit number.

2. **OBJECTIVES/AIM:**
   The aim of this lab is to write an assembly language program that takes a number from the user and calculates the summation of squares from 1 to N ($1^2 + 2^2 + 3^2 \ldots + N^2$). This program helps us understand how looping, arithmetic operations, and user input/output work in 8086 microprocessor programming. It also improves our ability to handle registers, perform repeated calculations, and display a numeric result in correct decimal format.

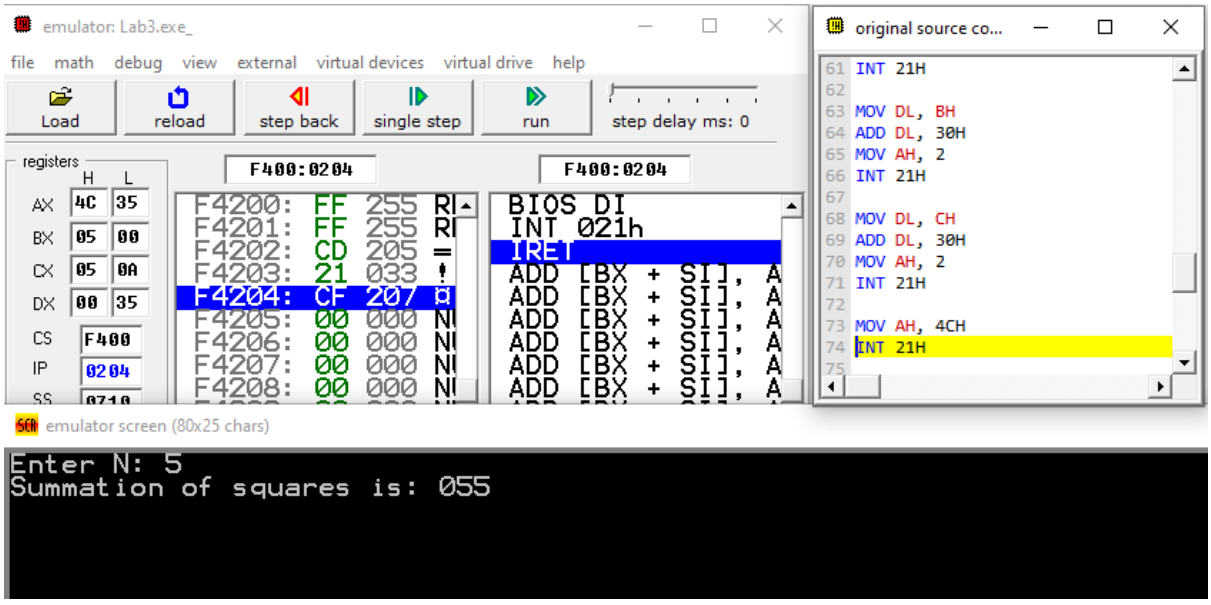
3. **IMPLEMENTATION PLAN & SYSTEM DEVELOPMENT :**
   For this program, I first identified the steps needed to take input from the user and then calculate the summation of squares from 1 to N. I initialized the data segment and displayed a prompt to enter a value. After reading the input, I stored it and used a loop to repeatedly square each number and add it to the total sum. Since assembly does not automatically handle decimal output, I manually converted the final result into hundreds, tens, and ones. Then I printed each digit one by one. Finally, I tested the program with different inputs to ensure the output was accurate and displayed correctly. This process allowed me to better understand how input/output operations, comparisons, and conditional logic work in 8086 assembly programming.

4. **IMPLEMENTATION:**

```
01 .MODEL SMALL
02 .STACK 100H
03 .DATA
04 N DB ?
05 MSG1 DB "Enter N: $"
06 MSG2 DB "Summation of squares is: $"
07 .CODE
08 MAIN PROC
09     MOV AX, @DATA
10     MOV DS, AX
11
12     ; Display "Enter N"
13     MOV DX, OFFSET MSG1
14     MOV AH, 9
15     INT 21H
```

```
    MOV AH, 1
    INT 21H
    SUB AL, 30H
    MOV N, AL
    ; New line
    MOV AH, 2
    MOV DL, 13
    INT 21H
    MOV DL, 10
    INT 21H
    ; Display output message
    MOV DX, OFFSET MSG2
    MOV AH, 9
    INT 21H
    ;Calculate sum of squares
    XOR BX, BX
    MOV CL, N

LOOP_SQ:
    MOV AL, CL
    MUL AL
    ADD BX, AX
    DEC CL
    JNZ LOOP_SQ

    ;Convert to 3 digits
    MOV AX, BX

    MOV CX, 100
    XOR DX, DX
    DIV CX
    MOV BL, AL      ; BL = hundreds
    MOV AX, DX      ; AX = remainder

    MOV CX, 10
    XOR DX, DX
    DIV CX
    MOV BH, AL
    MOV CH, DL

    ;Print 3 digits
    MOV DL, BL
    ADD DL, 30H
    MOV AH, 2
    INT 21H

    MOV DL, BH
    ADD DL, 30H
    MOV AH, 2
    INT 21H

    MOV DL, CH
    ADD DL, 30H
    MOV AH, 2
    INT 21H

    MOV AH, 4CH
    INT 21H

MAIN ENDP
END MAIN
```

5. OUTPUT:



6. ANALYSIS AND DISCUSSION:

In this experiment, I observed how assembly language handles arithmetic operations and loops at a low level. Unlike high-level languages, every step of the summation and multiplication had to be manually controlled using registers. The program successfully calculated the square of each number and accumulated the total using a loop, which helped reinforce my understanding of counter-based iteration. Converting the final result into three individual digits was a key challenge, as direct decimal output is not available in assembly. Through testing with different inputs, I confirmed that the program produces accurate results for single-digit inputs and displays them in three-digit format. Overall, the lab improved my practical understanding of microprocessor instruction flow and data handling.

7. SUMMARY:

In this lab, I developed an assembly program to calculate the summation of squares from 1 to N using a loop. The program takes a single-digit input from the user, multiplies each number by itself, and adds the results. I also learned how to break down a number and display it in a three-digit format since assembly does not print values directly. While writing this program, I practiced using registers, loops, and arithmetic instructions. Overall, the experiment improved my understanding of how low-level operations are executed inside the 8086 microprocessor.