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 Lab

Course Code: CSE 304 Section:232_D2

Lab Experiment Name: Practice of Loop in Emu8086

Student Details

	Name	ID
1.	Md. Sohan Millat Sakib	222902036

Lab Date : **27-11-2025**
Submission Date : **03-12-2025**
Course Teacher's Name : **Sagufta Sabah Nakshi**

Lab Report Status

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

1st problem

1. TITLE OF THE LAB REPORT EXPERIMENT

Write a loop in Assembly to calculate $1^2 + 2^2 + 3^2 + \ldots + n^2$.

2. OBJECTIVES

The main objectives of this experiment are:

- To learn how to take input and use loops in Assembly language.
- To understand how conditions and jumps work in Assembly.
- To write a simple program that calculates summation using a loop.

3. PROCEDURE

1. Declare a variable **A** to store numbers.

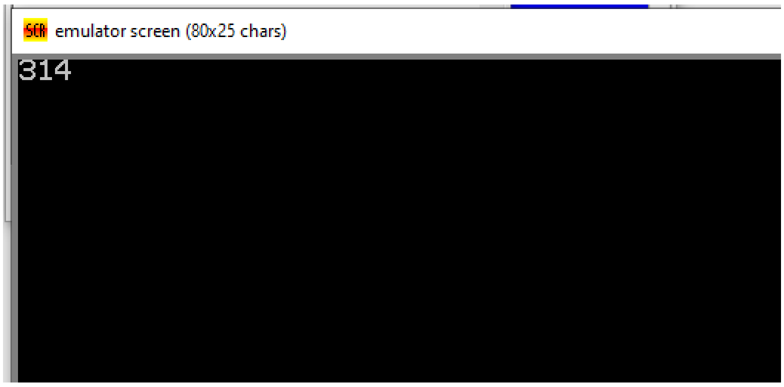
2. Take an input from the user.
3. Convert the character input to actual number by subtracting 48.
4. Move the input value into **CX** because CX controls the loop.
5. Set **BX = 0**, **AX = 0**, **A = 0** before starting the loop.
6. Inside the loop:
7. Increase A by 1
8. Move A to AX
9. Multiply AX × AX (square) and store result in BX
10. After the loop completes, divide the result and print the digits.
11. Print the output.

4. IMPLEMENTATION

```
ORG 100H

.data
A DW ?

.code
MOV AX, @data
MOV DS, AX

MOV AH, 01
INT 21H
SUB AL, 48
```

```
MOV AH, 0
MOV CX, AX

MOV BX, 0
MOV AX, 0
MOV A, 0

START:
INC A
INC AX
MUL AX
ADD BX, AX
MOV AX, A

LOOP START

MOV AX, BX
MOV BX, 10
MOV DX, 0
DIV BX
MOV BX, DX
MOV DX, AX
ADD DX, 48
MOV AH, 02
INT 21H

MOV DX, BX
ADD DX, 48
MOV AH, 02
INT 21H

RET
```

5. TEST RESULT



```
314
```

6. ANALYSIS AND DISCUSSION

- I can improve the program by adding strings for input and output messages.
- I can also print a new line to separate input and result more clearly.

2nd Problem

1. TITLE OF THE LAB REPORT EXPERIMENT

Use a loop to calculate the sum of all odd numbers from 1 to 99.
Also calculate the same result using formula.

2. OBJECTIVES

The main objectives of this experiment are:

- To write Assembly code using loops.
- To learn how odd numbers change inside a loop.
- To understand jump statements and simple conditions.

3. PROCEDURE

- There are **50 odd numbers** from 1 to 99, so CX is set to 50.
- Set **BX = 0** for storing sum.
- Set **AX = 1** because odd numbers start from 1.
- Each loop cycle:
- Add AX to BX (sum)
- Increase AX two times to reach next odd number.

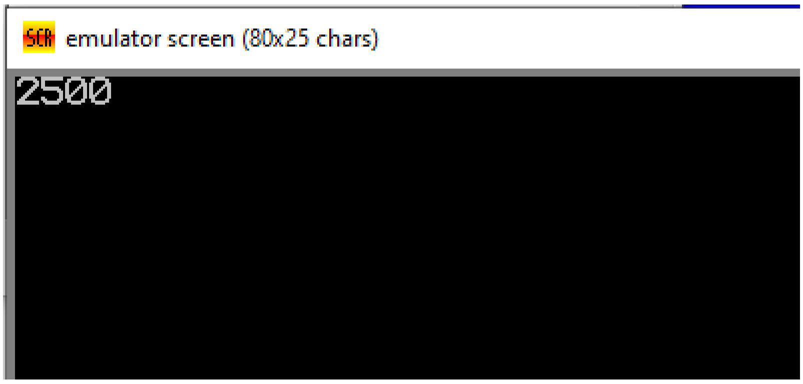
- After 50 loops, BX contains the total sum.
- Divide the sum by 1000, 100, and 10 to print each digit separately.

4. IMPLEMENTATION

```
ORG 100H

.data
.code
MOV AX,@data
MOV DS,AX

MOV CX,50

MOV BX,0
MOV AX,1

START:
ADD BX,AX
INC AX
INC AX

LOOP START

MOV AX,BX
MOV BX,1000
MOV DX,0
DIV BX
MOV BX,DX
MOV DX,AX
ADD DX,48
MOV AH,02
INT 21H

MOV AX,BX
MOV BX,100
MOV DX,0
DIV BX
MOV BX,DX
MOV DX,AX
ADD DX,48
MOV AH,02
INT 21H

MOV AX,BX
MOV BX,10
MOV DX,0
DIV BX
MOV BX,DX
MOV DX,AX
ADD DX,48
```

```
MOV AH,02
INT 21H

MOV DX,BX
ADD DX,48
MOV AH,02
INT 21H

RET
```

5. TEST RESULT



```
2500
```

6. ANALYSIS AND DISCUSSION

Can add an output string to clearly show result message. No problem at all

3rd report:

1. TITLE OF THE LAB REPORT EXPERIMENT

Take a number from the user and calculate its factorial.
Example: For n = 5 → factorial = 1 × 2 × 3 × 4 × 5 = 120

2. OBJECTIVES

The main objectives of this experiment are:

- To write a simple factorial program using Assembly.
- To understand loops and multiplication in Assembly.
- To practice taking user input and printing output.

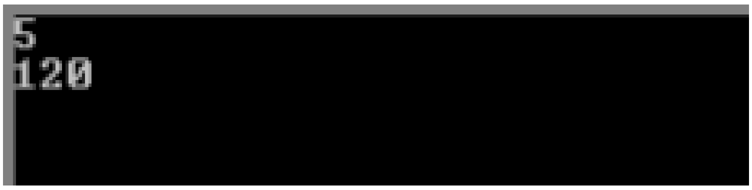
3. PROCEDURE

12. Declare variable A to store the number.
13. Take user input and store it in CX (loop counter).
14. Print a new line before showing output.
15. Set **DX = 0**, **BX = 0**, **AX = 1** to start factorial calculation.
16. Each loop:
17. Increase A
18. Increase BX
19. Multiply AX by A and store result
20. After loop ends, divide and print digits one by one.
21.

4. IMPLEMENTATION

```
ORG 100H

.data
A DW ?

.code
MOV AX, @data
MOV DS, AX

MOV AH, 01
INT 21H
SUB AL, 48
MOV AH, 0
MOV CX, AX

MOV AH, 02
MOV DL, 10
INT 21H

MOV DX, 0
MOV BX, 0
MOV AX, 1
MOV A, 0

START:
INC A
INC BX
MOV AL, A
MUL AL
MOV A, BX

LOOP START

MOV AX, A
MOV BX, 1000
MOV DX, 0
DIV BX
MOV BX, DX
MOV DX, AX
ADD DX, 48
MOV AH, 02
INT 21H

MOV AX, BX
MOV BX, 100
MOV DX, 0
DIV BX
MOV BX, DX
MOV DX, AX
ADD DX, 48
MOV AH, 02
INT 21H

MOV AX, BX
MOV BX, 10
MOV DX, 0
DIV BX
MOV BX, DX
MOV DX, AX
ADD DX, 48
MOV AH, 02
INT 21H

MOV DX, BX
ADD DX, 48
MOV AH, 02
INT 21H

RET
```

5. TEST RESULT



```
5
120
```

6. ANALYSIS AND DISCUSSION

- The output can be improved by adding simple text like “Factorial is: ”.
- Input message can also be added to make program easier to understand.

SUMMARY:

In this lab, I practiced basic Assembly language by writing three programs: summation of squares, summation of odd numbers, and factorial calculation. I used loops, conditions, and different registers to perform the calculations. Through these tasks, I learned how Assembly handles input, arithmetic operations, and output. This lab helped me understand fundamental Assembly concepts in a practical way.