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: 04

Course Title: Microprocessor & Microcontroller Lab

Course Code: CSE 304 **Section:** 213D1

Lab Experiment Name: Implementation of loop using assembly language

<u>Student Details</u>

Name		ID
1.	Md Jabed Hossen	213902046

Lab Date : 20-10-2023
Submission Date : 21-12-2023
Course Teacher's Name : Sudip Ghoshal

<u>Lab Report Status</u>

Marks: .... **Signature:.....................**
Comments:... **Date:..............................**

1. TITLE OF THE LAB REPORT EXPERIMENT

Implementation of loop using assembly language.

2. OBJECTIVES/AIM

- To understand basic loop
- To simplify repetitive tasks and reduce code redundancy.
- To enable the execution of specific operations for a specified number of iterations.
- To efficiently repeat a block of instructions.

3. PROCEDURE

Problem-1: Sum of 1+2+3+4+.....+99 natural number series

Step 1: Set the data segment.
Step 2: Initialize AX to 0 and set the initial value of i to 1.
Step 3: Start the loop 'doSum.'
Step 4: Add the value of i to AX and increment i by 2.
Step 5: Check if i is less than or equal to 99. If true, continue the loop.
Step 6: If i is greater than 99, exit the program.
Step 7: store ax result to ‘sum’ variable

Problem-2: Find factorial of a given number

Step 1: Set the data segment and stack.
Step 2: Take input from the user using DOS interrupt 21h.
Step 3: Subtract 48 from the ASCII value to convert it to a numerical digit.
Step 4: Set up the counter CX with the input value and initialize AX to 1.
Step 5: Begin the loop 'doFact' to calculate the factorial iteratively.
Step 6: Store the computed factorial result in the variable 'fact' defined in the data segment.

Problem-3: Check a number Odd or Even & if input N then exit the program

Step 1: Set the data segment and stack.
Step 2: Display the message to prompt the user for input.
Step 3: Take a number from the user using DOS interrupt 21h and store the input in register AL.
Step 4: Store the input value into BL for checking if it is 'N' to exit the program.
Step 5: Divide the input by 2 and store the remainder in AH to check if the number is even or odd.
Step 6: Compare the input with 'N' and exit the program if it matches.
Step 7: If the input is even, display the message 'Even'; otherwise, display 'Odd' and repeat the process.

4. IMPLEMENTATION

Problem-1: Sum of 1+3+5+.....+99 odd natural number series [using loop]

```
;summation of 1+3+5+7..+99 odd number
org 100h
.model small
.stack 100h
.data
   i dw 1
   sum dw ?

.code
    main proc
       ;set data segment
       mov ax, @data
       mov ds, ax

       mov ax, 0
       doSum:
       add ax, i
       inc i   ;increament i by 2 times
       inc i
       ;condition check to stop loop or continue
       cmp i, 99
       JLE doSum
       JG exit

       exit:
       ;before exit program, set ax to sum
       mov sum, ax
       mov ah, 4ch
       int 21h

       main endp
    end main
```

Problem-2: Find factorial of a given number

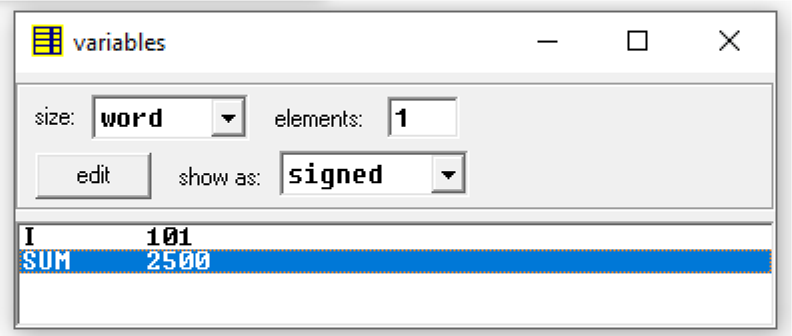
```
;factorial of a number
org 100h
.model small
.stack 100h
.data
    msg dw 'Enter a number: $'
    fact dw ? ;sum may big so taken 16bit

.code
    main proc
        mov ax, @data
        mov ds, ax

        mov ah, 9
        lea dx, msg
        int 21h
        mov ah, 1
        int 21h
        sub al, 48
        mov cx, 0
        mov cl, al ;i have used cl as counter

        ;since need to multiply like 1x2x3x4x.....
        mov ax, 1

        doFact:
        mul cx ;ax= ax*cx
        loop doFact

        mov fact, ax ;transfer b to sum variable

        main endp
    end main
```

Problem-3: Check a number Odd or Even & if input N then exit the program

```
;conditional loop
org 100h
.model small
.stack 100h
.data
    msg1 db 10, 13,'Enter a number: $'
    msg2 db 10, 13, 'Even $'
    msg3 db 10, 13, 'Odd $'

.code
    main proc
        mov ax, @data
        mov ds, ax

        again:
        mov ah, 9
        lea dx, msg1
        int 21h
```

```
     mov ah, 1
     int 21h   ;take a number from user
     mov bl, al
     mov ah, 0
     mov dl, 2
     div dl
    ;when input is N then exit the program
     cmp bl, 'N'
     JE exit

     ;if ah equal 0 goto even level
     cmp ah, 0
     JE even
     ;else if not equal, goto odd level
     JNE odd

     even:
     mov ah, 9
     lea dx, msg2
     int 21h
     jmp again

     odd:
     mov ah, 9
     lea dx, msg3
     int 21h
     jmp again

     exit:
     mov ah, 4ch
     int 21h
     main endp
end main
```

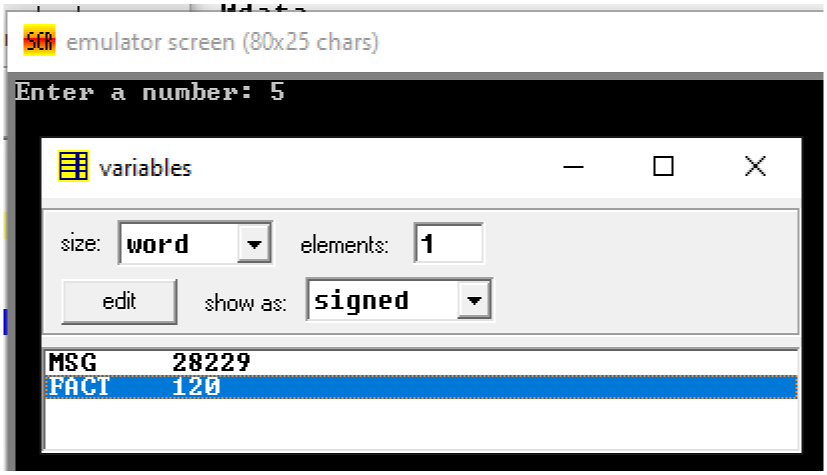
5. OUTPUT



size: word elements: 1
show as: signed

```
I          101
SUM        2500
```

Figure-1: Summation of 99th natural number using loop

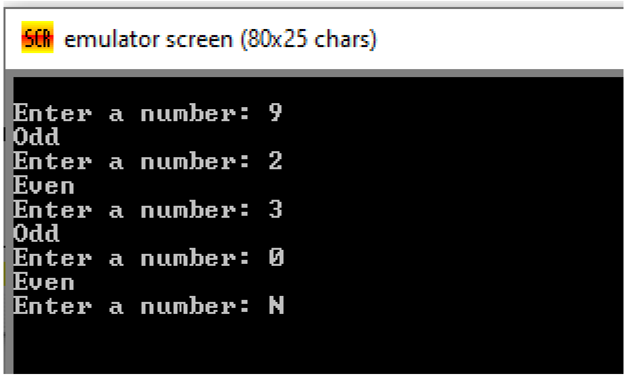


```
Enter a number: 5
```

size: word elements: 1
show as: signed

```
MSG        28229
FACT       120
```

Figure-2: Factorial of a number and showed into variabl



```
Enter a number: 9
Odd
Enter a number: 2
Even
Enter a number: 3
Odd
Enter a number: 0
Even
Enter a number: N
```

Figure-3: Check a number Odd or Even & if input N then exit the program

6. ANALYSIS AND DISCUSSION

The first problem, involving the summation of a natural odd number series, employs a loop to compute the sum from 1 to 99, demonstrating an efficient approach in iterative process management, emphasizing the importance of proper initialization and control flow within loop structures. The second problem, which involves calculating the factorial of a given number, effectively uses iterative multiplication, highlighting the adept use of the 'MUL' instruction in a loop construct for consecutive number multiplication. Meanwhile, the third problem focuses on determining whether a number is even or odd, showcasing the application of conditional branching in response to user input. This program adeptly handles different input types, integrating interrupt services and conditional statements to accurately display messages based on the input number's parity, illustrating a comprehensive approach in processing and interpreting user input.