



# Khulna University of Engineering & Technology, Khulna

Department of Biomedical Engineering

SESSIONAL REPORT

Course No: **BME 2152**

Experiment No : **02**

Name of the Experiment: **Performing loop and built-in operations using MATLAB.**

Remarks:

Date of performance: 18 December 2024

Date of submission : 1 January 2025

Objectives:

Name: **Md. Fuad Hasan Hamim**

Roll : **2215029**

Year : **2<sup>nd</sup>**

Term : **1<sup>st</sup>**

## Objectives:

The main objectives of this sessional are-

- To learn about loop operation in MATLAB
- To get familiar with built-in operations in MATLAB
- To know how to find standard deviation and variance using built-in command
- To learn the syntax of performing loop operations, built-in operations and printing results in command window

## Introduction:

Loops or Iteration Statements in Programming are helpful when we need a specific task in repetition. They're essential as they reduce hours of work to seconds. Loops are fundamental to the concept of iteration in programming, enhancing code efficiency, readability and promoting the reuse of code logic. There are mainly two types of loops – Entry Controlled Loop & Exit controlled Loop. Entry Controlled Loop mean the test condition is checked before entering the main body of the loop (e.g. For loop and While loop). On the other hand, Exit controlled loop mean the test condition is checked at the end of the loop body (e.g. – Do While Loop).

For Loop:

For loop is a control flow statement for specifying iteration. Specifically, a for-loop functions by running a section of code repeatedly until a certain condition has been satisfied.

Syntax of For Loop for MATLAB:

```
for variable = starting value: ending value  
end
```

Built-in Operations:

MATLAB has some built-in functions or operations like other programming software. For example, mean() to find the mean value, sum() to find sum of variables, std() for standard deviation, var() for variance.

Standard Deviation:

A standard deviation is a measure of how dispersed the data is in relation to the mean. Low, or small, standard deviation indicates data are clustered tightly around the mean, and high, or large, standard deviation indicates data are more spread out. A standard deviation close to zero indicates that data points are very close to the mean, whereas a larger standard deviation indicates data points are spread further away from the mean. In MATLAB, there is std() function to determine standard deviation of data.

Variance:

The measure of how far the set of data is dispersed from their mean value. In MATLAB, var() is used to find variance of data.

**Task 1 || Performing addition from 1 to 100 using for loop.****Solution:**

<b>Coding</b>
<pre>%Performing addition from 1 to 100 using for loop. % clc; clear all; sum=0; a = 1:1:100; for i = 1:100     sum = sum + i; end fprintf("Sum is %d", sum);</pre>
<b>Output</b>
Sum is 5050 >>
<b>Comment</b>
Thus, we find the sum using for loop. In line 5, we could write 1:100 instead of 1:1:100.

**Task 2 || Performing addition of even number from 1 to 100 using for loop.****Solution:**

<b>Coding</b>
<pre>%Performing addition of even number from 1 to 100 using for loop.% clc; clear all; sum=0; a = 2:2:100; for i = 2:2:100;     sum = sum + i; end fprintf("Sum is %d", sum);</pre>
<b>Output</b>
Sum is 2550 >>
<b>Comment</b>
The even numbers start from 2 and the difference between two consecutive even number is 2. So, we write the condition for increment which is 2.

### Task 3 || Performing addition of odd number from 1 to 100 using for loop.

#### Solution:

Coding
<pre>%Performing addition of odd number from 1 to 100 using for loop.% clc; clear all; sum=0; a = 1:2:100; for i = 1:2:100;     sum = sum + i; end fprintf("Sum is %d", sum);</pre>
Output
<pre>Sum is 2500 &gt;&gt;</pre>
Comment
The odd numbers start from 1 and the difference between two consecutive odd number is 2. So, we write the condition for increment which is 2.

### Task 4 || Finding average value from 1 to 100.

#### Solution:

Coding
<pre>% Find Average value for 1 to 100. % clc; clear all; sum=0; a = 1:1:100; b=length(a); for i = 1:1:100;     sum = sum + i; end avg=sum/b; fprintf("Average of 1-100 is %d", avg);</pre>
Output
<pre>Average of 1-100 is 5.050000e+01 &gt;&gt;</pre>
Comment
To find the average, we need to know about the length (Total number of values). So, we used a built-in function length() to find the length of data.

**Task 5 || Finding average value of even number from 1 to 100.****Solution:**

<b>Coding</b>
<pre>% Finding average value of even number from 1 to 100. % clc; clear all; sum=0; a = 2:2:100; b=length(a); for i = 2:2:100;     sum = sum + i; end avg=sum/b; fprintf("Average of even number from 1 to 100 is %d", avg);</pre>
<b>Output</b>
<pre>Average of even number from 1 to 100 is 51 &gt;&gt;</pre>
<b>Comment</b>
Here, we used length() function like before.

**Task 6 || Finding average value of odd number from 1 to 100.****Solution:**

<b>Coding</b>
<pre>% Finding average value of odd number from 1 to 100. % clc; clear all; sum=0; a = 1:2:100; b=length(a); for i = 1:2:100;     sum = sum + i; end avg=sum/b; fprintf("Average of odd number from 1 to 100 is %d", avg);</pre>
<b>Output</b>
<pre>Average of odd number from 1 to 100 is 50 &gt;&gt;</pre>
<b>Comment</b>
Here, we used length() function like before.

### Task 7 || Performing addition from 1 to 100 using built-in commands.

#### Solution:

Coding
<pre>% Performing addition from 1 to 100 using built-in command. % clc; clear all; a = 1:1:100; b=sum(a); fprintf('Sum is %d\n', b);</pre>
Output
<pre>Sum is 5050 &gt;&gt;</pre>
Comment
There is a built-in function sum() to determine sum of values. We used it here, instead of using for loop.

### Task 8 || Finding standard deviation and variance of 1 to 100 integer number.

#### Solution:

Coding
<pre>% Performing arithmetic operations using built-in command. % clc; clear all; a = 1:1:100; b=std(a); c=var(a); fprintf('Standard Deviation is %d\n', b); fprintf('Variance is %d', c);</pre>
Output
<pre>Standard Deviation is 2.901149e+01 Variance is 8.416667e+02 &gt;&gt;</pre>
Comment
We know that, in MATLAB we can use std() function for finding standard deviation and var() to find variance of data.

**Task 9 || A = [1, 5, 10, 2,7, 100]. Find sum, mean, std, var using built-in function.**

**Solution:**

Coding
<pre>% A = [1, 5, 10, 2,7, 100]. Find sum, mean, std, var using built-in function. % clc; clear all; A = [1, 5, 10, 2,7, 100]; a=sum(A); b=mean(A); c=std(A); d=var(A); fprintf("Sum is %d\n", a); fprintf("Mean is %d\n", b); fprintf("Standard Deviation is %d\n", c); fprintf("Variance is %d\n", d);</pre>
Output
<pre>Sum is 125 Mean is 2.083333e+01 Standard Deviation is 3.892257e+01 Variance is 1.514967e+03 &gt;&gt;</pre>
Comment
<p>We used built-in functions to find sum, mean, standard deviation and variance of values from an array declared on the program.</p>

## Conclusion:

In this sessional, we've learned how to use for loop and some built-in functions like length(), sum(), mean(), std(), var() etc. on MATLAB in solving repetitive problems. Using these tools, we've solved some complex problems which are very lengthy and time consuming to solve manually. This sessional would help us to solve further complex problems.

## References:

1. <https://www.geeksforgeeks.org/loops-programming/>
2. <https://www.geeksforgeeks.org/functions-in-matlab/>
3. <https://www.nlm.nih.gov/oet/ed/stats/02-900.html>
4. <https://www.geeksforgeeks.org/variance-and-standard-deviation/>