



## Lab Report

Only for course Teacher						
		Needs Improvement	Developing	Sufficient	Above Average	Total Mark
Allocate mark & Percentage		25%	50%	75%	100%	5
Clarity	1					
Content Quality	2					
Spelling & Grammar	1					
Organization and Formatting	1					
Total obtained mark						
Comments						

**Semester: Spring ..... / Fall .2023..**

**Student Name: Md Rufsán Jani Shanto**

**Student ID: 221-35-1064**

**Batch: 37**

**Section: D2**

**Course Code: SE-233**

**Course Name: Operating system & system programme**

**Course Teacher Name: Ishrat Sultana**

**Designation: Lecturer**

**Submission Date: 28... / .11... / .23...**

Course Code: SE233

Course Title: Operating System &amp; System Program

Lab Shell Scripting Problem

**Problem 1.** Write a script that calculates the sum of numbers from 1 to a user-defined limit using a for loop.

```
#!/bin/bash
```

```
echo -n "Enter the limit that you need: "
```

```
read n
```

```
sum=0
```

```
for ((i=1; i<=n; i++)); # using for loop
```

```
do
```

```
    ((sum+=i))
```

```
done
```

```
echo "The sum is: $sum"
```

### Output

```
Enter the limit that you need: The sum is: 55
```

```
[Execution complete with exit code 0]
```

Comment: Here I use for loop. I input the limited value n=10. As the loop start from i=1. So, sum=1+2+.....+10=55

**Problem 2.** Write a script that takes an array of numbers as input from the user and calculates the average of those numbers.

```
#!/bin/bash

echo "Enter numbers : "
read -a num
sum=0
for num in "${num[@]}";
do
    ((sum+=num))
done
avg=$((sum / ${#num[@]}))
echo "The average of those numbers is: $avg"
```

1 2 3 4 5

### Output

```
Enter numbers :
The average of those numbers is: 3

[Execution complete with exit code 0]
```

Comment: I take 5 number (1,2,3,4,5) in array. Sum=1+2+3+4+5=15

Average =15/5=3

**Problem 3.** Write a script that takes an array of floating-point numbers as input from the user and calculates the average with precision.

```
#!/bin/bash

echo "Enter floating-point numbers :"
read -a f_n # input floating-point numbers referred by f_n
sum=0
for float_num in "${f_n[@]"; do
    sum=$(echo "$sum + $float_num" | bc -l)
done
avg=$(echo "scale=2; $sum / ${#f_n[@]}" | bc -l)
echo "The average with precision is: $avg"
```

7.55 3.40

### Output

```
Enter floating-point numbers :
The average with precision is: 5.47

[Execution complete with exit code 0]
```

Comment: The average of my numbers=5.47. As I use "scale=2" it give 2 precision number from decimal point .If I use "scale=3" it give 3 precision number from decimal point and average=5.475

7.55 3.40

### Output

```
Enter floating-point numbers :
The average with precision is: 5.475

[Execution complete with exit code 0]
```

**Problem 4.** Write a script that takes a positive integer as input from the user and uses a while loop to print all numbers from 1 to that integer, skipping the even numbers.

```
#!/bin/bash
```

```
echo -n "As your choice input a positive number : "
```

```
read n
```

```
i=1
```

```
while [ $i -le $n ];# -le is less than mean i less then my input value n
```

```
do
```

```
    if [ $((i % 2)) -ne 0 ];
```

```
    then
```

```
        echo $i
```

```
    fi
```

```
    ((i++))
```

```
done
```

#### Output

```
As your choice input a positive number : 1
```

```
3
```

```
5
```

```
[Execution complete with exit code 0]
```

Comment: I take a positive number 5. As I take i=1 so, my code takes 1 to 5. After skipping the even number from 1 to 5 it gives the output 1,3,5 (skipping 2&4)

**Problem 5.** Write a script that defines a function to greet the user by printing a personalized message. The script should take the user's name as input and use the function to greet the user.

```
#!/bin/bash

greet_user() {
    echo "Hi, $1! Welcome to my programme!"
}
```

```
echo -n "Input the name: "
read username
greet_user "$username"
```

Md Rufsán Jani Shantó

#### Output

```
Input the name: Hi, Md Rufsán Jani Shantó! Welcome to my programme!

[Execution complete with exit code 0]
```

Comment: I make a function → greet user ()

After input my name here call the function **greet user "\$username"** & Give output.