

ASSIGNMENT – I

Name: Athul K Jose

Email: athulkjose001@gmail.com

1. Check Whether a Number is Positive, Negative, or Zero:

Script:

```
check_number() {  
    if [ $1 -gt 0 ]; then  
        echo "Number is Positive"  
    elif [ $1 -lt 0 ]; then  
        echo "Number is Negative"  
    else  
        echo "Number is Zero"  
    fi  
}
```

```
read -p "Enter a number: " num  
check_number $num
```

Output:

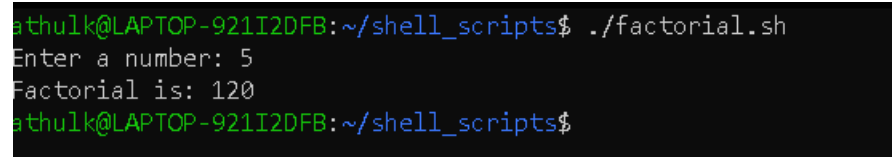
```
athulk@LAPTOP-921I2DFB:~/shell_scripts$ ./checknumber.sh  
Enter a number: 19  
Number is Positive  
athulk@LAPTOP-921I2DFB:~/shell_scripts$
```

2. Factorial of a Number:

Script:

```
factorial() {  
    n=$1  
    fact=1  
  
    while [ $n -gt 0 ]  
    do  
        fact=$((fact * n))  
        n=$((n - 1))  
    done  
  
    echo "Factorial is: $fact"  
}  
  
read -p "Enter a number: " num  
factorial $num
```

Output:



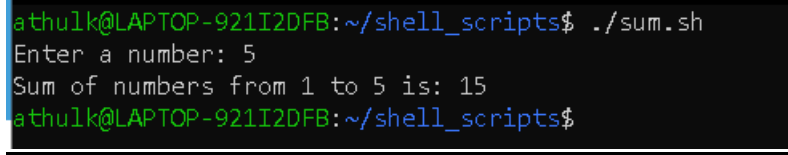
```
athul@LAPTOP-921I2DFB:~/shell_scripts$ ./factorial.sh  
Enter a number: 5  
Factorial is: 120  
athul@LAPTOP-921I2DFB:~/shell_scripts$
```

3. Sum of N Numbers:

Script:

```
sum_n() {  
    n=$1  
    sum=0  
  
    for ((i=1; i<=n; i++))  
    do  
        sum=$((sum + i))  
    done  
  
    echo "Sum of numbers from 1 to $n is: $sum"  
}  
  
read -p "Enter a number: " num  
sum_n $num
```

Output:



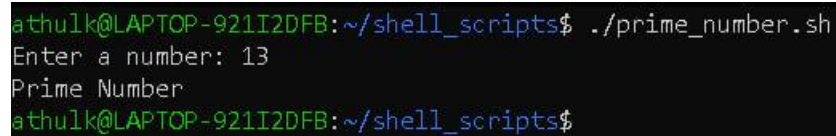
```
athul@LAPTOP-921I2DFB:~/shell_scripts$ ./sum.sh  
Enter a number: 5  
Sum of numbers from 1 to 5 is: 15  
athul@LAPTOP-921I2DFB:~/shell_scripts$
```

4. Prime Number Check:

Script:

```
is_prime() {  
    n=$1  
  
    if [ $n -le 1 ]; then  
        echo "Not a Prime Number"  
        return  
    fi  
  
    for ((i=2; i<=n/2; i++))  
    do  
        if [ $((n % i)) -eq 0 ]; then  
            echo "Not a Prime Number"  
            return  
        fi  
    done  
  
    echo "Prime Number"  
}  
  
read -p "Enter a number: " num  
is_prime $num
```

Output:



```
athulk@LAPTOP-921I2DFB:~/shell_scripts$ ./prime_number.sh  
Enter a number: 13  
Prime Number  
athulk@LAPTOP-921I2DFB:~/shell_scripts$
```

5. Reverse a Number:

Script:

```
read -p "Enter a number: " num
```

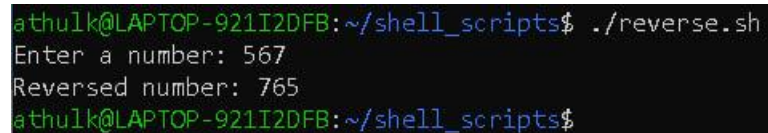
```
rev=0
```

```
if [ $num -lt 0 ]; then  
    num=$(( -num ))  
fi
```

```
while [ $num -gt 0 ]  
do  
    digit=$((num % 10))  
    rev=$((rev * 10 + digit))  
    num=$((num / 10))  
done
```

```
echo "Reversed number: $rev"
```

Output:



```
athulk@LAPTOP-921I2DFB:~/shell_scripts$ ./reverse.sh  
Enter a number: 567  
Reversed number: 765  
athulk@LAPTOP-921I2DFB:~/shell_scripts$
```

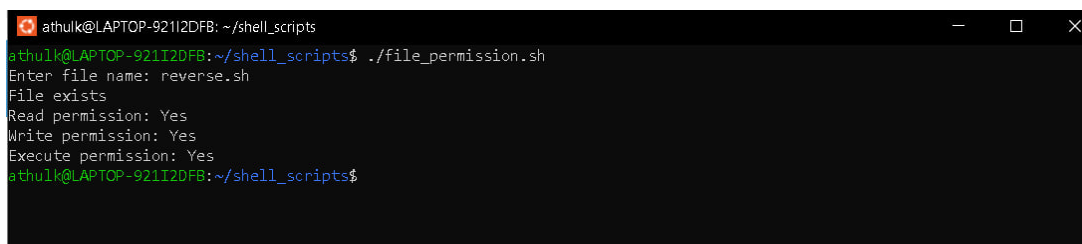
6. File Existence & Permission Check:

Script:

```
check_file() {  
    file=$1  
  
    if [ -e "$file" ]; then  
        echo "File exists"  
  
        if [ -r "$file" ]; then  
            echo "Read permission: Yes"  
        else  
            echo "Read permission: No"  
        fi  
  
        if [ -w "$file" ]; then  
            echo "Write permission: Yes"  
        else  
            echo "Write permission: No"  
        fi  
  
        if [ -x "$file" ]; then  
            echo "Execute permission: Yes"  
        else  
            echo "Execute permission: No"  
        fi  
    else  
        echo "File does not exist"  
    fi  
}
```

```
read -p "Enter file name: " fname  
check_file "$fname"
```

Output:

A terminal window with a black background and white text. The window title is 'athulk@LAPTOP-921I2DFB: ~/shell_scripts'. The prompt is 'athulk@LAPTOP-921I2DFB:~/shell_scripts\$'. The user enters './file_permission.sh'. The script prompts 'Enter file name: reverse.sh'. The output shows 'File exists', 'Read permission: Yes', 'Write permission: Yes', and 'Execute permission: Yes'. The prompt returns to 'athulk@LAPTOP-921I2DFB:~/shell_scripts\$'.

```
athulk@LAPTOP-921I2DFB: ~/shell_scripts  
athulk@LAPTOP-921I2DFB:~/shell_scripts$ ./file_permission.sh  
Enter file name: reverse.sh  
File exists  
Read permission: Yes  
Write permission: Yes  
Execute permission: Yes  
athulk@LAPTOP-921I2DFB:~/shell_scripts$
```

7. Menu-Driven Calculator:

Script:

```
add() {
    echo "Result: $((a + b))"
}

sub() {
    echo "Result: $((a - b))"
}

mul() {
    echo "Result: $((a * b))"
}

div() {
    if [ $b -eq 0 ]; then
        echo "Division by zero not allowed"
    else
        echo "Result: $((a / b))"
    fi
}

while true
do
    echo "----- MENU -----"
    echo "1. Addition"
    echo "2. Subtraction"
    echo "3. Multiplication"
    echo "4. Division"
    echo "5. Exit"

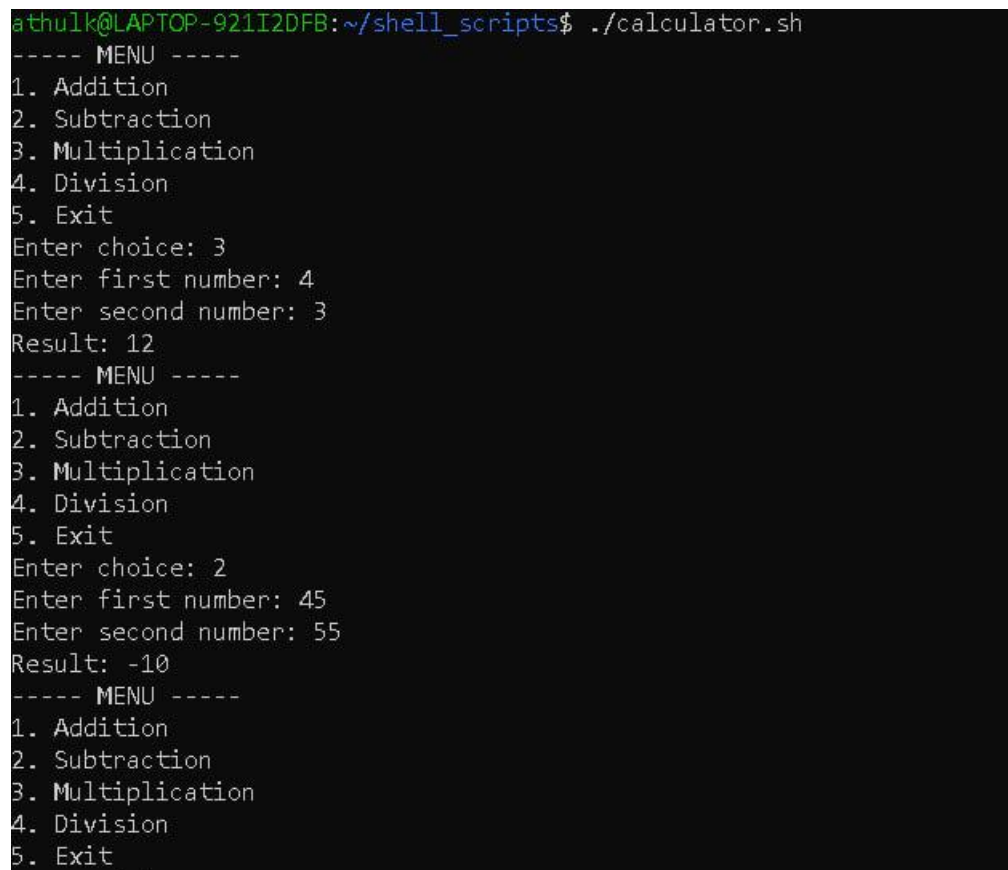
    read -p "Enter choice: " choice

    if [ $choice -eq 5 ]; then
        echo "Exiting..."
        break
    fi

    read -p "Enter first number: " a
    read -p "Enter second number: " b
```

```
case $choice in
    1) add ;;
    2) sub ;;
    3) mul ;;
    4) div ;;
    *) echo "Invalid choice" ;;
esac
done
```

Output:

A screenshot of a terminal window showing the execution of a calculator script. The prompt is 'athulk@LAPTOP-921I2DFB:~/shell_scripts\$./calculator.sh'. The script displays a menu with five options: 1. Addition, 2. Subtraction, 3. Multiplication, 4. Division, and 5. Exit. The user enters '3' for multiplication, then '4' for the first number and '3' for the second number, resulting in 'Result: 12'. The menu is shown again, and the user enters '2' for subtraction, then '45' for the first number and '55' for the second number, resulting in 'Result: -10'. The menu is shown a third time, and the user enters '5' to exit the script.

```
athulk@LAPTOP-921I2DFB:~/shell_scripts$ ./calculator.sh
----- MENU -----
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
Enter choice: 3
Enter first number: 4
Enter second number: 3
Result: 12
----- MENU -----
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
Enter choice: 2
Enter first number: 45
Enter second number: 55
Result: -10
----- MENU -----
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
```

