Assignment 3

Name: Om Gore

PRN no.:12110548

Roll no. : 56

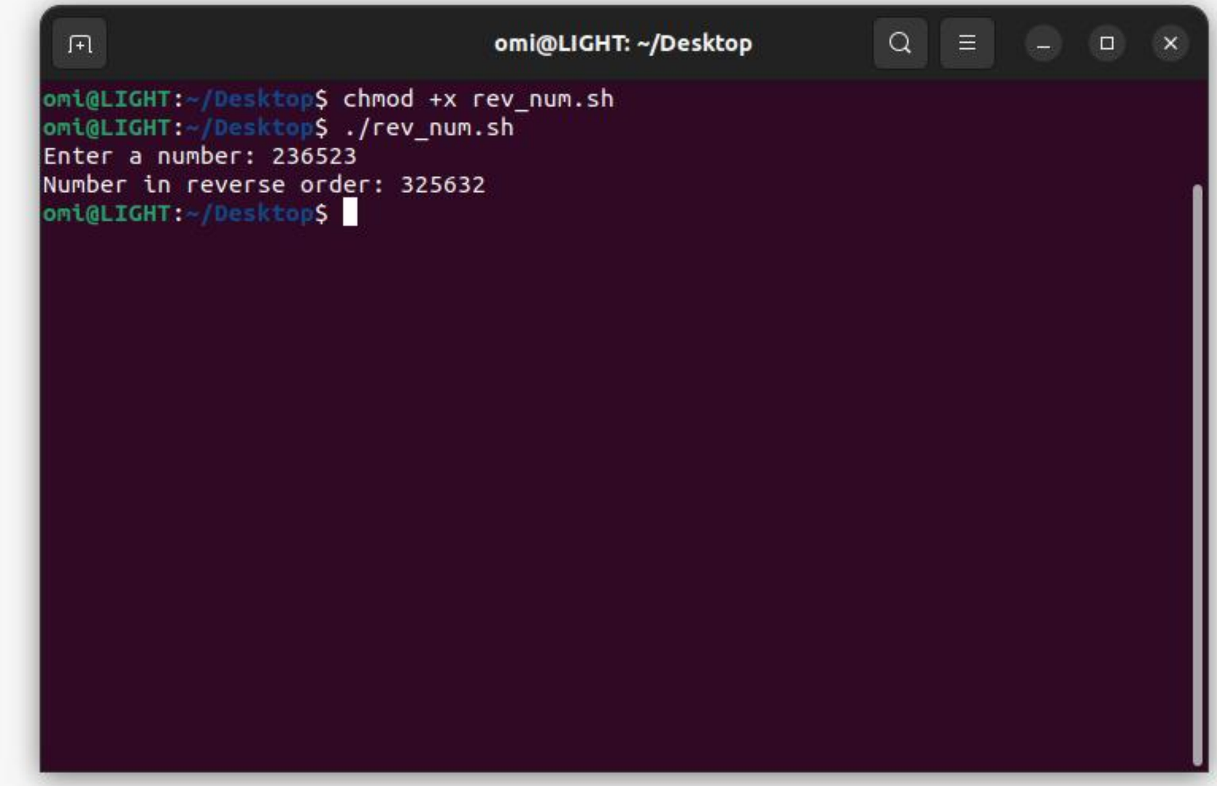
Course Name: Operating System

Problem Statement: Write Shell Programs

Code 1: Write a shell program to print given number in reverse order

```
#!/bin/bash

# Function to reverse a number
reverse_number() {
    num=$1
    reversed=0

    while [ $num -gt 0 ]; do
        remainder=$((num % 10))
        reversed=$((reversed * 10 + remainder))
        num=$((num / 10))
    done

    echo $reversed
}

# Main script
echo -n "Enter a number: "
read input_number

reversed_result=$(reverse_number $input_number)
echo "Reversed number: $reversed_result"
```

Output:

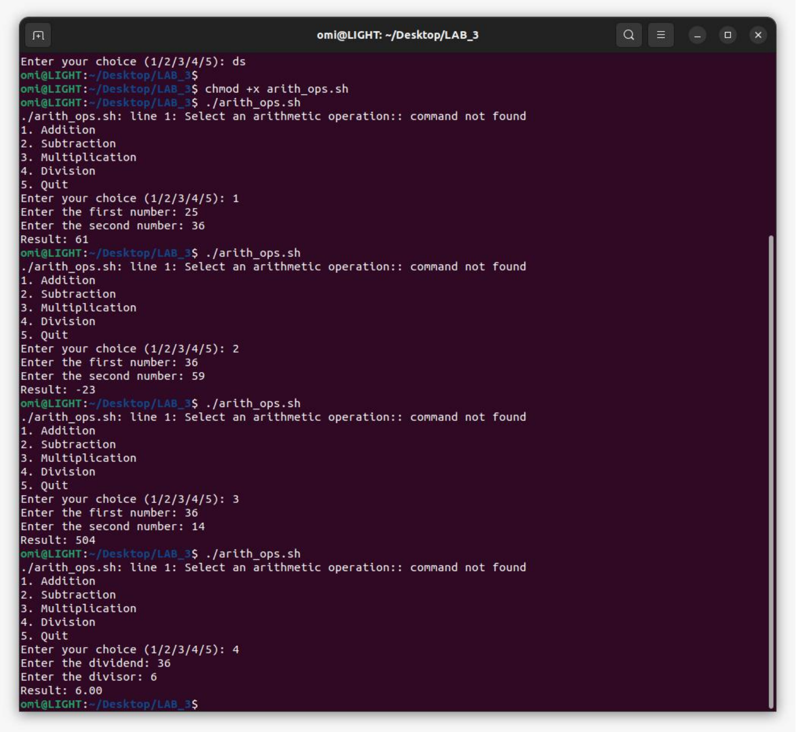


```
omi@LIGHT:~/Desktop$ chmod +x rev_num.sh
omi@LIGHT:~/Desktop$ ./rev_num.sh
Enter a number: 236523
Number in reverse order: 325632
omi@LIGHT:~/Desktop$
```

Code 2: Write a shell program to perform arithmetic operations using case

```
#!/bin/bash

# Function to perform addition
addition() {
   result=$(($1 + $2))
   echo "Result: $result"
}

# Function to perform subtraction
subtraction() {
   result=$(($1 - $2))
   echo "Result: $result"
}

# Function to perform multiplication
multiplication() {
    result=$(($1 * $2))
    echo "Result: $result"
}

# Function to perform division
division() {
    if [ $2 -ne 0 ]; then
        result=$(($1 / $2))
        echo "Result: $result"
    else
        echo "Error: Division by zero is not allowed"
    fi
}

# Main script
echo "Arithmetic Operations"
echo "1. Addition"
echo "2. Subtraction"
echo "3. Multiplication"
echo "4. Division"

# Read the operation choice from the user
echo -n "Enter your choice (1/2/3/4): "
read choice

case $choice in
    1)
        echo -n "Enter the first number: "
        read num1
        echo -n "Enter the second number: "
        read num2
        addition $num1 $num2
        ;;
    2)
        echo -n "Enter the first number: "
        read num1
        echo -n "Enter the second number: "
        read num2
        subtraction $num1 $num2
        ;;
    3)
        echo -n "Enter the first number: "
        read num1
        echo -n "Enter the second number: "
        read num2
        multiplication $num1 $num2
        ;;
    4)
        echo -n "Enter the dividend: "
        read num1
        echo -n "Enter the divisor: "
        read num2
        division $num1 $num2
        ;;
    *)
        echo "Invalid choice"
        ;;
esac
```

Output:

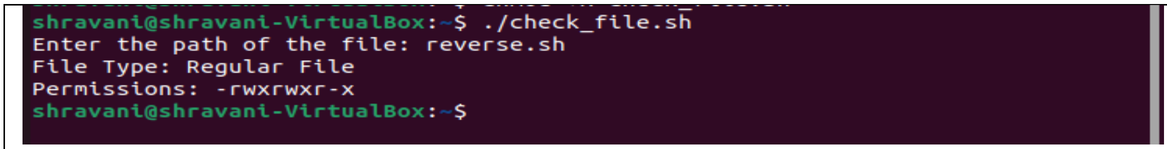


```
Enter your choice (1/2/3/4/5): ds
omi@LIGHT:~/Desktop/LAB_3$
omi@LIGHT:~/Desktop/LAB_3$ chmod +x arith_ops.sh
omi@LIGHT:~/Desktop/LAB_3$ ./arith_ops.sh
./arith_ops.sh: line 1: Select an arithmetic operation:: command not found
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Quit
Enter your choice (1/2/3/4/5): 1
Enter the first number: 25
Enter the second number: 36
Result: 61
omi@LIGHT:~/Desktop/LAB_3$ ./arith_ops.sh
./arith_ops.sh: line 1: Select an arithmetic operation:: command not found
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Quit
Enter your choice (1/2/3/4/5): 2
Enter the first number: 36
Enter the second number: 59
Result: -23
omi@LIGHT:~/Desktop/LAB_3$ ./arith_ops.sh
./arith_ops.sh: line 1: Select an arithmetic operation:: command not found
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Quit
Enter your choice (1/2/3/4/5): 3
Enter the first number: 36
Enter the second number: 14
Result: 504
omi@LIGHT:~/Desktop/LAB_3$ ./arith_ops.sh
./arith_ops.sh: line 1: Select an arithmetic operation:: command not found
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Quit
Enter your choice (1/2/3/4/5): 4
Enter the dividend: 36
Enter the divisor: 6
Result: 6.00
omi@LIGHT:~/Desktop/LAB_3$
```

Code 3: Write a shell script to check file type and permissions of a given input by user

```
#!/bin/bash

# Read the file path from the user
echo -n "Enter the path of the file: "
read file_path

# Check if the file exists
if [ -e "$file_path" ]; then
    # Determine file type
    if [ -f "$file_path" ]; then
        file_type="Regular File"
    elif [ -d "$file_path" ]; then
        file_type="Directory"
    elif [ -L "$file_path" ]; then
        file_type="Symbolic Link"
    else
        file_type="Other"
    fi

    # Get file permissions
    permissions=$(ls -l "$file_path" | awk '{print $1}')

    echo "File Type: $file_type"
    echo "Permissions: $permissions"
else
    echo "Error: File not found"
fi
```

Output:

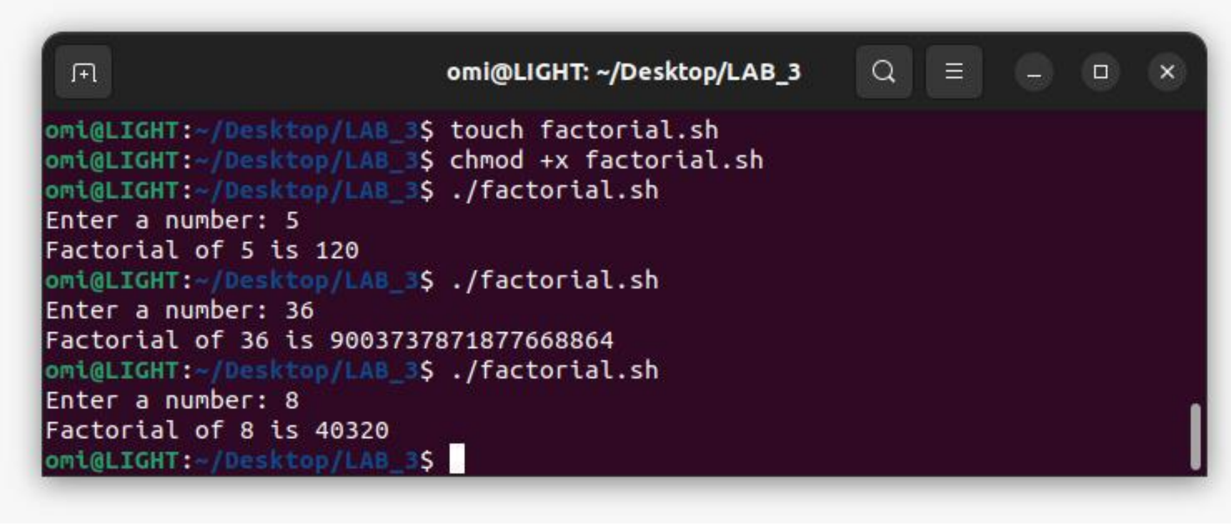


```
shravani@shravani-VirtualBox:~$ ./check_file.sh
Enter the path of the file: reverse.sh
File Type: Regular File
Permissions: -rwxrwxr-x
shravani@shravani-VirtualBox:~$
```

Code 4: Write a shell script to Find factorial of a given number using function

```
#!/bin/bash

# Function to calculate factorial
factorial() {
    if [ $1 -eq 0 ]; then
        echo 1
    else
        prev_factorial=$(factorial $(($1 - 1)))
        echo $(($1 * $prev_factorial))
    fi
}

# Main script
echo -n "Enter a number: "
read num

if [ $num -lt 0 ]; then
    echo "Factorial is not defined for negative numbers."
else
    result=$(factorial $num)
    echo "Factorial of $num is $result"
fi
```

Output:

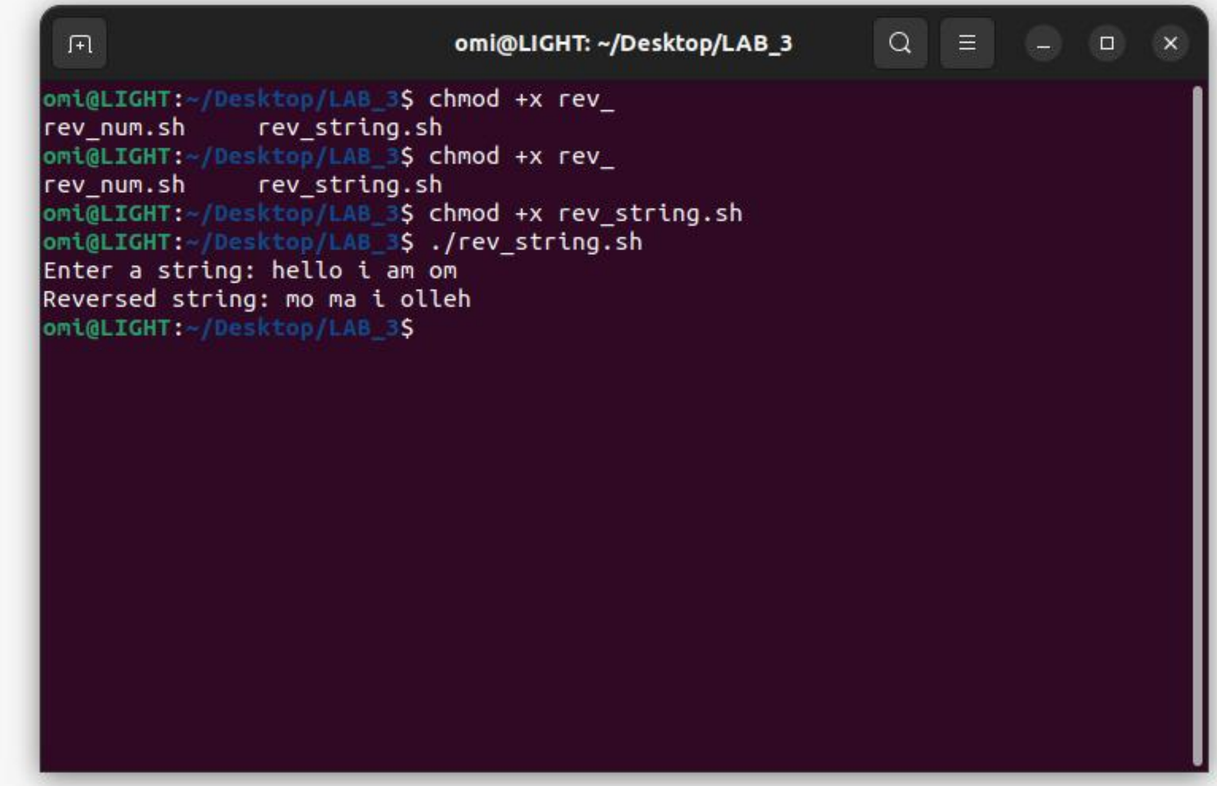


```
omi@LIGHT:~/Desktop/LAB_3$ touch factorial.sh
omi@LIGHT:~/Desktop/LAB_3$ chmod +x factorial.sh
omi@LIGHT:~/Desktop/LAB_3$ ./factorial.sh
Enter a number: 5
Factorial of 5 is 120
omi@LIGHT:~/Desktop/LAB_3$ ./factorial.sh
Enter a number: 36
Factorial of 36 is 9003737871877668864
omi@LIGHT:~/Desktop/LAB_3$ ./factorial.sh
Enter a number: 8
Factorial of 8 is 40320
omi@LIGHT:~/Desktop/LAB_3$
```

Code 5: Write a shell script to reverse a string

```
#!/bin/bash

# Read the input string from the user
echo -n "Enter a string: "
read input_string

# Calculate the length of the input string
length=${#input_string}

# Initialize the reversed string
reversed_string=""

# Loop through each character in the input string in reverse order and append it to the reversed string
for (( i=$length-1; i>=0; i-- ))
do
    reversed_string="${reversed_string}${input_string:$i:1}"
done

# Print the reversed string
echo "Reversed string: $reversed_string"
```

Output:

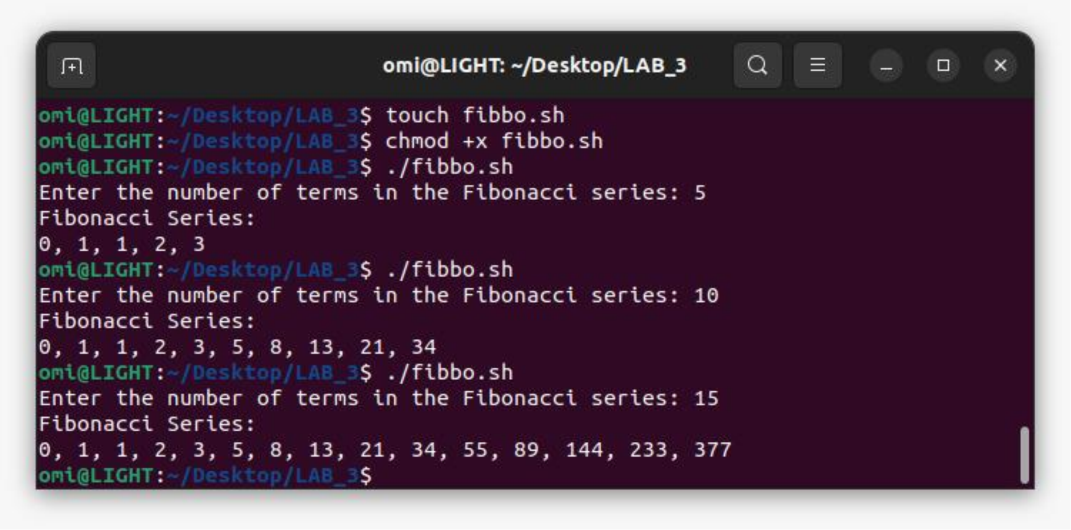


```
omi@LIGHT:~/Desktop/LAB_3$ chmod +x rev_
rev_num.sh     rev_string.sh
omi@LIGHT:~/Desktop/LAB_3$ chmod +x rev_
rev_num.sh     rev_string.sh
omi@LIGHT:~/Desktop/LAB_3$ chmod +x rev_string.sh
omi@LIGHT:~/Desktop/LAB_3$ ./rev_string.sh
Enter a string: hello i am om
Reversed string: mo ma i olleh
omi@LIGHT:~/Desktop/LAB_3$
```

Code 6: Write a shell script to print fibonacci series

```
#!/bin/bash

# Function to generate Fibonacci series
fibonacci_series() {
    num_terms=$1

    # Initialize the first two terms
    term1=0
    term2=1

    # Print the first two terms
    echo -n "$term1 $term2 "

    # Generate and print the rest of the terms
    for ((i=2; i<num_terms; i++))
    do
        next_term=$((term1 + term2))
        echo -n "$next_term "
        term1=$term2
        term2=$next_term
    done

    echo # Print a newline at the end
}

# Main script
echo -n "Enter the number of terms in the Fibonacci series: "
read num_terms

if [ $num_terms -le 0 ]; then
    echo "Number of terms should be greater than zero."
else
    echo "Fibonacci Series:"
    fibonacci_series $num_terms
fi
```

Output:



```
omi@LIGHT:~/Desktop/LAB_3$ touch fibbo.sh
omi@LIGHT:~/Desktop/LAB_3$ chmod +x fibbo.sh
omi@LIGHT:~/Desktop/LAB_3$ ./fibbo.sh
Enter the number of terms in the Fibonacci series: 5
Fibonacci Series:
0, 1, 1, 2, 3
omi@LIGHT:~/Desktop/LAB_3$ ./fibbo.sh
Enter the number of terms in the Fibonacci series: 10
Fibonacci Series:
0, 1, 1, 2, 3, 5, 8, 13, 21, 34
omi@LIGHT:~/Desktop/LAB_3$ ./fibbo.sh
Enter the number of terms in the Fibonacci series: 15
Fibonacci Series:
0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377
omi@LIGHT:~/Desktop/LAB_3$
```