LAB IMPLEMENTATION RECORD

OPERATING SYSTEM AND CONCEPTS

BCSC 0856

Institute of Engineering & Technology

B. TECH CS HONORS
(2022-23)

Submitted By

Shiva Srivastava

(2115800023)

Submitted To

Mr. Pappu Kumar
Bhagat

Department of Computer Engineering & Applications

Department of Computer Engineering and Applications,
GLA University, 17 km. Stone NH#2,
Mathura-Delhi Road, Chaumuha, Mathura – 281406, UP

CERTIFICATE

This is to certify that I, **Shiva Srivastava**, have done hands-on implementation of all the practical's mentioned in this Lab Implementation Record File under the supervision of **Mr. Pappu Kumar Bhagat,** I have submitted the file in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (Computer Science & Engineering) Honors.

Signature of Supervisor:

Name of Supervisor: Mr. Pappu Kumar Bhagat

Date: 02-05-2023

TABLE OF CONTENTS

Part 1

Part 2

Basic Problems on Bash Scripting Language

Basic Problems on Bash Scripting Language:

Programs:

1) Program to find the sum of square of individual digits of a number.

   Program:

```
#!/bin/bash

echo "Enter a number: "
read num

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

echo "Sum of square of digits: $sum"
```

   Result:

```
 ENTER A NUMBER:

1234

SUM OF SQUARE OF DIGITS: 30
```

2) Program to find the sum of cube of individual digits of a number.

   Program:

```
#!/bin/bash

echo "Enter a number: "
read num

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

echo "Sum of cube of digits: $sum"
```

Result:

```
ENTER A NUMBER:

1234

SUM OF CUBE OF DIGITS: 100
```

3) Program to execute various UNIX commands using case statements set of numbers.

```
#!/bin/bash

echo "Enter a number corresponding to the
command to execute:"
echo "1. List files in the current directory"
echo "2. Show the current date and time"
echo "3. Show the current user"
echo "4. Show the current working directory"
echo "5. Exit"
read num

case $num in
  1) ls -l ;;
  2) date ;;
  3) whoami ;;
  4) pwd ;;
  5) exit ;;
  *) echo "Invalid input" ;;
esac
```

Result:

```
ENTER A NUMBER CORRESPONDING TO THE
COMMAND TO EXECUTE:

1. LIST FILES IN THE CURRENT DIRECTORY

2. SHOW THE CURRENT DATE AND TIME

3. SHOW THE CURRENT USER

4. SHOW THE CURRENT WORKING DIRECTORY

5. EXIT

2

SAT MAY  6 12:45:00 PDT 2023
```

4) Program to count the number of vowels in a line of text.

```
#!/bin/bash

echo "Enter a line of text: "
read line

vowel_count=0
for (( i=0; i<${#line}; i++ )); do
   char="${line:$i:1}"
   case $char in

([aeiouAEIOU])vowel_count=$((vowel_count+1)) ;;
   esac
done

echo "Number of vowels: $vowel_count"
```

Result:

```
ENTER A LINE OF TEXT:

THE QUICK BROWN FOX JUMPS OVER THE LAZY
DOG.

NUMBER OF VOWELS: 11
```

5) Program to display student grades.

```
#!/bin/bash

echo "Enter the student's name: "
read name

echo "Enter the student's marks in 3 subjects (separated by spaces): "
read marks1 marks2 marks3

average=$(( (marks1+marks2+marks3) / 3 ))

echo "$name's grades:"
echo "Subject 1: $marks1"
echo "Subject 2: $marks2"
echo "Subject 3: $marks3"
echo "Average marks: $average"

if [ $average -ge 90 ]
then
   echo "Grade: A+"
elif [ $average -ge 80 ]
then
   echo "Grade: A"
elif [ $average -ge 70 ]
then
   echo "Grade: B"
elif [ $average -ge 60 ]
then
   echo "Grade: C"
else
   echo "Grade: F"
fi
```

Result:

```
ENTER THE STUDENT'S NAME:

SHIVA SRIVASTAVA

ENTER THE STUDENT'S MARKS IN 3 SUBJECTS
(SEPARATED BY SPACES):

85 92 89

SHIVA SRIVASTAVA'S GRADES:

SUBJECT 1: 85

SUBJECT 2: 92

SUBJECT 3: 89

AVERAGE MARKS: 88

GRADE: A
```

6) Program to find the smallest number from a set of numbers:

```
#!/bin/bash

# Read input numbers
read -p "Enter the numbers separated by space: " numbers

# Set the first number as the minimum
min=$1

# Iterate through the numbers and update the minimum value
for num in ${numbers[@]}
do
  if [ $num -lt $min ]
  then
    min=$num
  fi
done

# Print the minimum value
echo "The smallest number is: $min"
```

Result:

```
ENTER THE NUMBERS SEPARATED BY SPACE: 5 2 7 1 8

THE SMALLEST NUMBER IS: 1
```

6) Program to find the smallest number from a set of numbers.
   Program:

```
#!/bin/bash

# Read input number
read -p "Enter a number: " num

# Set the first digit as the minimum
min=${num:0:1}

# Iterate through the digits and update the minimum value
for digit in $(echo $num | fold -w1)
do
  if [ $digit -lt $min ]
  then
    min=$digit
  fi
done

# Print the minimum value
echo "The smallest digit is: $min"
```

Result:

```
ENTER THE NUMBERS SEPARATED BY SPACE: 5 2 7 1 8

THE SMALLEST NUMBER IS: 1
```

7) Program to find the smallest digit from a number.

Program:

```
#!/bin/bash

# Read input number
read -p "Enter a number: " num

# Set the first digit as the minimum
min=${num:0:1}

# Iterate through the digits and update the minimum value
for digit in $(echo $num | fold -w1)
do
  if [ $digit -lt $min ]
  then
    min=$digit
  fi
done

# Print the minimum value
echo "The smallest digit is: $min"
```

Result:

```
ENTER A NUMBER: 72385

THE SMALLEST DIGIT IS: 2
```

8) Program to find all numbers between 50 and 100, which are divisible by 3 and not divisible

Program:

```
#!/bin/bash

# Loop through numbers between 50 and 100
for (( i=50; i<=100; i++ ))
do
  # Check if the number is divisible by 3 and not divisible by 5
  if [ $(( i % 3 )) -eq 0 ] && [ $(( i % 5 )) -ne 0 ]
  then
    echo $i
  fi
done
```

Result:

```
51 54 57 63 66 69 72 78 81 84 87 93 96 99
```

9) Program to find the sum of digits of a number until a single digit is obtained.

```
#!/bin/bash

# Read input number
read -p "Enter a number: " num

# Define function to compute the digit sum
function digit_sum() {
  local sum=0
  for digit in $(echo $1 | fold -w1)
  do
    sum=$(( sum + digit ))
  done
  echo $sum
}

# Loop until the number becomes a single digit
while [ ${#num} -gt 1 ]
do
  num=$(digit_sum $num)
done

# Print the final single digit
echo "The sum of digits until a single digit is:
$num"
```

Result:

```
ENTER A NUMBER: 789

THE SUM OF DIGITS UNTIL A SINGLE DIGIT IS: 6
```

10) Program to find the second highest number from a set of numbers.

Program:

```
#!/bin/bash

# Initialize the array of numbers
nums=(23 45 11 56 89 4 76)

# Sort the array in descending order
sorted_nums=($(printf '%s\n' "${nums[@]}" | sort -n -r))

# Find the second highest number
second_highest=${sorted_nums[1]}

echo "The second highest number is: $second_highest"
```

Result:

```
THE SECOND HIGHEST NUMBER IS: 76
```