EXPERIMENT NO.: 2 **DATE:** 22/08/2023

TITLE: Shell Script

AIM: To implement the following programs using shell script:

i. To find the difference of second largest and second smallest digit of a number.

ii. Calculate Roots of Quadratic Equation

iii. Generate the series: 1 3 2 4 3 5 4 6 5

iv. To calculate factorial of a number

v. Calculate sum and average of given n numbers

THEORY:

A shell is a command-line interpreter and typical operations performed by shell scripts include file manipulation, program execution, and printing text. To Run Shell Script program, do the following:

1. Save the file with "filename.sh" having the shell script code by creating a Directory and saving this file under that Directory Name say XYZ.

2. Go to cmd/ Linux shell and say "cd XYZ" and hit Enter.

3. Now say ./filename.sh to Run the file.

IF-ELSE Statement:

- The if...else...fi statement is the next form of control statement that allows Shell to execute statements in a controlled way and make the right choice.

- If the resulting value is true, given statement(s) are executed.

- If the expression is false, then no statement will be executed.

- Syntax:

```
if [ expression ] then

        Statement(s) to be executed if expression is true else
        Statement(s) to be executed if expression is not true

  fi
```

WHILE Loop:

- The while loop enables you to execute a set of commands repeatedly until some condition occurs.
- It is usually used when you need to manipulate the value of a variable repeatedly.
- If the resulting value is true, given statement(s) are executed.
- If command is false then no statement will be executed and the program will jump to the next line after the done statement.
- Syntax:

```
while command do
        Statement(s) to be executed if command is true
Done
```

CASE Condition:

- The basic syntax of the case...esac statement is to give an expression to evaluate and to execute several different statements based on the value of the expression.
- Here the string word is compared against every pattern until a match is found.
- The statement(s) following the matching pattern executes.
- If nothing matches, a default condition will be used.
- When statement(s) part executes, the command ;; indicates that the program flow should jump to the end of the entire case statement. This is similar to break in the C programming language.
- Syntax:

```
case word in
        pattern1)
```

Statement(s) to be executed if pattern1 matches

;;

pattern2)

Statement(s) to be executed if pattern2 matches

;;

*)

Default condition to be executed

;;

Esac

FOR Loop:

- The for loop operate on lists of items.
- It repeats a set of commands for every item in a list.
- Here var is the name of a variable and word 1 to word N are sequences of characters separated by spaces (words).
- Syntax:

for var in word 1 word 2 ...word n

do

Statement to be executed

done

CODE:

I) To find the difference of second largest and second smallest digit of a number.

```
echo 'Enter a number'
read num
m=$num
largest=0
smallest=9

while [ $m -ne 0 ]
do
   d=$(($m%10))
   if [ $largest -lt $d ]
   then
      largest=$d
   fi
   if [ $smallest -gt $d ]
   then
      smallest=$d
   fi
   m=$(($m/10))

done

echo "laregest,smallest : $largest,$smallest"

m=$num
second_largest=0
second_smallest=$largest
while [ $m -ne 0 ]
do
   d=$(($m%10))
   if [ $second_largest -lt $d ]
   then
      if [ $d -ne $largest ]
      then
         second_largest=$d
      fi
   fi
   if [ $second_smallest -gt $d ]
   then
      if [ $d -ne $smallest ]
      then
         second_smallest=$d
```

```
      fi
   fi
   m=$((m/10))
done

echo "2 nd laregest , smallest : $second_largest,$second_smallest"

diff=$(($second_largest - $second_smallest))
echo "difference = $diff"
```

O/P :

```
smitesh@LAPTOP-867GCRA6:/mnt/c/Users/smite/Desktop/CP/SE_LAB/OS$ bash exp2_1.sh
Enter a number
2356
laregest,smallest : 6,2
2 nd laregest , smallest : 5,3
difference = 2
smitesh@LAPTOP-867GCRA6:/mnt/c/Users/smite/Desktop/CP/SE_LAB/OS$
```

vi. Calculate Roots of Quadratic Equation

```
echo "enter a b c for  eqn 1 : ax2 + bx + c"
read a
read b
read c

d=$(($b*$b - 4*$a*$c))

if [ $d -gt 0 ]
then
  root1=$( echo "scale=2; (-$b + sqrt($d)) / (2*$a)" | bc )
  root2=$( echo "scale=2; (-$b - sqrt($d)) / (2*$a)" | bc )
  echo "Root 1 : $root1"
  echo "Root 2 : $root2"
else [ $d -eq 0 ]
 root=$( echo "scale=2; -$b / (2*$a)" | bc )
 echo "roots : $root"
fi
```

```
smitesh@LAPTOP-867GCRA6:/mnt/c/Users/smite/Desktop/CP/SE_LAB/OS$ bash exp2_2.sh
enter a b c for  eqn 1 : ax2 + bx + c
2
4
2
roots : -1.00
smitesh@LAPTOP-867GCRA6:/mnt/c/Users/smite/Desktop/CP/SE_LAB/OS$
```

vii. Generate the series: 1 3 2 4 3 5 4 6 5

```
echo "Enter n "
read n
flag=2
i=1
j=0
while [ $j -ne $n ]
do
  echo "$i"
  if [ $flag -eq 2 ]
  then
      i=$(($i + $flag))
      flag=-1
  else
     i=$(($i + $flag))
     flag=2
  fi
  j=$(($j+1))
done
```

```
smitesh@LAPTOP-867GCRA6:/mnt/c/Users/smite/Desktop/CP/SE_LAB/OS$ bash exp2_3.sh
Enter n
9
1
3
2
4
3
5
4
6
5
smitesh@LAPTOP-867GCRA6:/mnt/c/Users/smite/Desktop/CP/SE_LAB/OS$
```

viii. To calculate factorial of a number

```
echo "Enter number"
read n
fact=1
while [ $n -gt 0 ]
do
                fact=$(($fact * $n))
                n=$(($n - 1))
done
echo "Factorial is $fact"
```

```
smitesh@LAPTOP-867GCRA6:/mnt/c/Users/smite/Desktop/CP/SE_LAB/OS$ bash Factorial.sh
Enter number
4
Factorial is 24
smitesh@LAPTOP-867GCRA6:/mnt/c/Users/smite/Desktop/CP/SE_LAB/OS$
```

ix. Calculate sum and average of given n numbers

```
echo "Enter number of terms"
read n
sum=0
num=$n
while [ $num -gt 0 ]
do
        echo "Enter number"
        read a
        sum=$(($sum + $a))
        num=$(($num - 1))
done
avg=$(($sum / $n))
echo "Sum is $sum"
echo "Average is $avg"
```

```
smitesh@LAPTOP-867GCRA6:/mnt/c/Users/smite/Desktop/CP/SE_LAB/OS$ bash Sum.sh
Enter number of terms
4
Enter number
1
Enter number
2
Enter number
3
Enter number
4
Sum is 10
Average is 2
smitesh@LAPTOP-867GCRA6:/mnt/c/Users/smite/Desktop/CP/SE_LAB/OS$
```