EXPERIMENT NO:2 **DATE –**

SHELL PROGRAMMING

Aim – To write shell script for the following

- **(i)** Find sum of second largest and second smallest digit of a number.
- **(ii)** Generate the series : 1 3 2 4 3 5 4 6 5
- **(iii)** Find the sum of even digits of the number .
- **(iv)** Calculate area and circumference of circle
- **(v)** Display 10 leap years from 2000

THOERY –

Shell scripting is a way to automate tasks in Unix/Linux using a series of commands written in a file. It supports variables, control structures, loops, and functions, making it powerful for system administration and automation.

1. If-Else Statement

Explanation: The if-else statement is used to make decisions in shell scripts based on conditions.

Syntax:

```
if [ condition ]
then
  commands if condition is true
else
  commands if condition is false
fi
```

Example:

Check if a number is positive or negative.

```
if [ number -gt 0 ]
```

```
then
  echo "Number is Positive"
else
  echo "Number is Negative"
fi
```

2. While Loop

Explanation: The while loop executes a block of commands repeatedly as long as the condition is true.

Syntax:

```
while [ condition ]
do
  commands
done
```

Example:

Print numbers from 1 to 5.

```
count=1
while [ $count -le 5 ]
do
  echo $count
  count=`expr $count + 1`
done
```

3. Case Conditional

Explanation: The case statement is used when we need to execute one block of code from multiple options. It is similar to the switch statement in other programming languages.

Syntax:

```
case variable in
  pattern1) commands ;;
```

```
  pattern2) commands ;;
  *) default commands ;;
esac
```

Example:

Check the day of the week.

```
case $day in
  Mon) echo "Today is Monday" ;;
  Tue) echo "Today is Tuesday" ;;
  *) echo "Other day" ;;
esac
```

4. For Loop

Explanation: The for loop is used to iterate over a list of values or a range.

Syntax:

```
for variable in list
do


  commands
done
```

Example:

Print numbers from 1 to 5.

```
for i in 1 2 3 4 5
do
  echo $i
done
```

PROGRAMS –

i]

Code –

```
#!/bin/bash

echo -n "Enter a number: "
read num
digits=($(echo "$num" | grep -o . | sort -n | uniq))
count=${#digits[@]}
if [ $count -lt 2 ]; then
  echo "Not enough unique digits to find second largest and second smallest."
  exit 1
fi
second_smallest=${digits[1]}
second_largest=${digits[$((count-2))]}
sum=$((second_smallest + second_largest))

echo "Second smallest digit: $second_smallest"
echo "Second largest digit: $second_largest"
echo "Sum: $sum"
```

OUTPUT –

```
root@Atharv:~/Atharv/Exp2$ bash 1.sh
Enter a number: 345
Second smallest digit: 4
Second largest digit: 4
Sum: 8
```

II]

CODE –

```
#!/bin/bash

echo -n "Enter number of terms: "
read n

num=1
echo -n "$num "

for ((i=1; i<=n; i++))
do
   if (( i % 2 == 0 )); then
      num=$((num - 1))
   else
      num=$((num + 2))
   fi
   echo -n "$num "
done

echo
```

OUTPUT –

```
root@Atharv:~/Atharv/Exp2$ bash 2.sh
Enter number of terms: 8
1 3 2 4 3 5 4 6 5
```

Iii]

CODE –

```
#!/bin/bash

echo -n "Enter a number: "
read num

sum=0
temp=$num

while [ $temp -gt 0 ]
do
   digit=$((temp % 10))
   if (( digit % 2 == 0 )); then
      sum=$((sum + digit))
   fi
   temp=$((temp / 10))
done

echo "Sum of even digits of $num = $sum"
```

OUTPUT –

```
root@Atharv:~/Atharv/Exp2$ bash 3.sh
Enter a number: 45689
Sum of even digits of 45689 = 18
```

Iv]

CODE –

```
#!/bin/bash

echo -n "Enter the radius of the circle: "
read r

pi=3.14

area=$(echo "$pi * $r * $r" | bc -l)
circumference=$(echo "2 * $pi * $r" | bc -l)

echo "Area of circle = $area"
echo "Circumference of circle = $circumference"
```

OUTPUT –

```
root@Atharv:~/Atharv/Exp2$ bash 4.sh
Enter the radius of the circle: 12
Area of circle = 452.16
Circumference of circle = 75.36
```

V]

CODE –

```
#!/bin/bash

count=0
year=2000
```

```
echo "First 10 leap years from 2000 are:"

while [ $count -lt 10 ]
do
   if [ $((year % 400)) -eq 0 ] || { [ $((year % 4)) -eq 0 ] && [ $((year % 100)) -ne 0 ]; }
   then
      echo $year
      count=$((count + 1))
   fi
   year=$((year + 1))
done
```

OUTPUT –

```
root@Atharv:~/Atharv/Exp2$ bash 5.sh
First 10 leap years from 2000 are:
2000
2004
2008
2012
2016
2020
2024
2028
2032
2036
```

CONCLUSION -The Shell programs were written and executed successfully.