Name: Priyanka Gupta **PRN: 22070122157**

Subject Title: Operating Systems Lab

EXPERIMENT No: 6

TITLE: Loop Statements

Aim: Implement shell script to demonstrate loop statements.

Learning Outcome: 1. To understand the loop statements
2. To demonstrate shell script to implement loop statements using Linux Commands

Hardware/Software:

This experiment is performed using Ubuntu OS in a virtualized environment within Windows 11. The software used is:

- VMware Workstation 17 Player (Free Virtualization Tool)
- Ubuntu 22.04.1 LTS ISO file
- VMware Tools installed on the virtual machine

Problem Statement:

1. Shell script to print pattern

```
    *
   ***
  *****
 *******
*********
 *******
  *****
   ***
    *
```

2. Shell script to using for loop to print the pattern:

```
1
22
333
4444
55555
4444
333
22
1
```

Name: Priyanka Gupta **PRN: 22070122157**

Theory:

The shell acts as an interface between the user and the OS services. It accepts human-readable commands from the user and converts them into a language that the kernel can understand. The shell can be accessed using the terminal.

Shell accepts the command as input from the user and executes it. A shell can also take inputs as a file and execute them in case of executing a bunch of commands routinely. This helps to avoid the repetitive work of typing each command in the terminal. This file containing a bunch of commands is known as a shell script and is saved with the extension of .sh.

Three types of looping statements can be used in shell scripting. These include:

1. **while statement**
   A while loop is a control flow structure that allows to repeatedly execute a block of code as long as the condition specified is evaluated to be true
   The syntax is:
   while [condition]
   do
   #Statement blocks to be executed
   done
   Here, '[condition]' is the test condition to be evaluated before each iteration of the loop. If the condition is evaluated to be true, the statements in the body of the loop are executed. The iteration of the loop continues until the condition becomes false at which the control is transferred to the 'done' keyword signifying the termination of the while loop.

2. **for statement**
   A for loop is a control flow structure that allows repeatedly executing a block of code a specified number of times or for each item in a list.
   The syntax is:
   for i in list
   do
   #Statement blocks to be executed
   done
   Here, 'i' is the test iterator variable that takes the value of each item in the 'list'. The statements in the body of the loop are executed for each value of 'i'. The iteration of the loop continues until all the items in the list have been processed after which the control is transferred to the 'done' keyword signifying the termination of the for loop.

3. **until statement**
   An until statement is a control flow structure that allows repeatedly executing a block of code until a specified condition is true.
   The syntax is:
   until [condition]
   do
   #Statement blocks to be executed
   done

Here, '[condition]' is the test condition that is evaluated before each iteration of the loop. If the condition is false, the statements inside the loop are executed. This process continues until the condition becomes true, at which point control is transferred to the statement following the done keyword.

The until loop is similar to the while loop, except that the test condition is inverted. In other words, a while loop will continue to execute as long as the condition is true, while an until loop will continue to execute as long as the condition is false.

Note:
By default, the .sh file may not have executable permission. Hence, using the command chmod +x filename, the file is given executable permission.

#!/bin/bash signifies that the script should be executed using the bash shell

Algorithm:
For performing looping statements using shell scripting:

1. Take input from the user for the number of rows and store it in some variables
2. Use nested for loop to execute over the range of numbers till n, for rows and columns
3. Inside the nested for loop, write code that will print a * character for each column in the current row.
4. After the nested for loop for each row, write code to print a newline character to move to the next row.
5. The required output is generated

Steps to Execute the Program:

1. Create a new file and make it executable using the chmod command
2. Edit the file using Text Editor and add the commands required for performing the desired looping statements.
3. Save and close the file. The file extension should be .sh
4. Run the script in the terminal using ./filename.sh
5. Based on the operations, provide the necessary inputs and the output is generated based on the evaluation/iteration of specified looping statements.

Name: Priyanka Gupta **PRN: 22070122157**

FOR LOOP:-

1. Shell script to print pattern star pattern

```
#!/bin/trash

echo "Enter a number: "
read maxnum

echo
for ((i=0;i<maxnum; i++))
do
  for ((j=i; j<maxnum-1; j++))
  do
    echo -n " "
  done

  for ((j=0; j<(2*i+1); j++))
  do
    echo -n "*"
  done
  echo
done

for ((i=maxnum-1; i>0; i--))
do
  for ((j=maxnum; j>i; j--))
  do
    echo -n " "
    done
    for ((j=0;j<(2*i-1); j++))
    do
      echo -n "*"
    done
    echo
  done
```

```
student@ubuntu:~/priyanka$ bash assign6.sh
ENter a number:
6

     *
    ***
   *****
  *******
 *********
***********
 *********
  *******
   *****
    ***
     *
student@ubuntu:~/priyanka$ bash assign6.sh
Enter a number:
7

      *
     ***
    *****
   *******
  *********
 ***********
*************
 ***********
  *********
   *******
    *****
     ***
      *
```

Name: Priyanka Gupta **PRN: 22070122157**

2. Shell script to using for loop to print the pattern:

Open assign6.sh ~/priyanka

```
#!/bin/trash

echo "Enter max number of stars: "
read num
echo
for ((i=1; i<num; i++))
do
  for ((j=1; j<i+1; j++))
  do
    echo -n "$i"
  done
  echo
done
for ((i=num; i>=0; i--))
do
  for ((j=i; j>0; j--))
  do
    echo -n "$i"
  done
  echo
done
```

```
student@ubuntu:~/priyanka$ bash assign6.sh
Enter max number of stars:
5

1
22
333
4444
55555
4444
333
22
1

student@ubuntu:~/priyanka$ bash assign6.sh
Enter max number of stars:
8

1
22
333
4444
55555
666666
7777777
88888888
7777777
666666
55555
4444
333
22
1
```

Name: Priyanka Gupta **PRN: 22070122157**

WHILE LOOP:

1. Shell script to print pattern star pattern

starWhile.sh

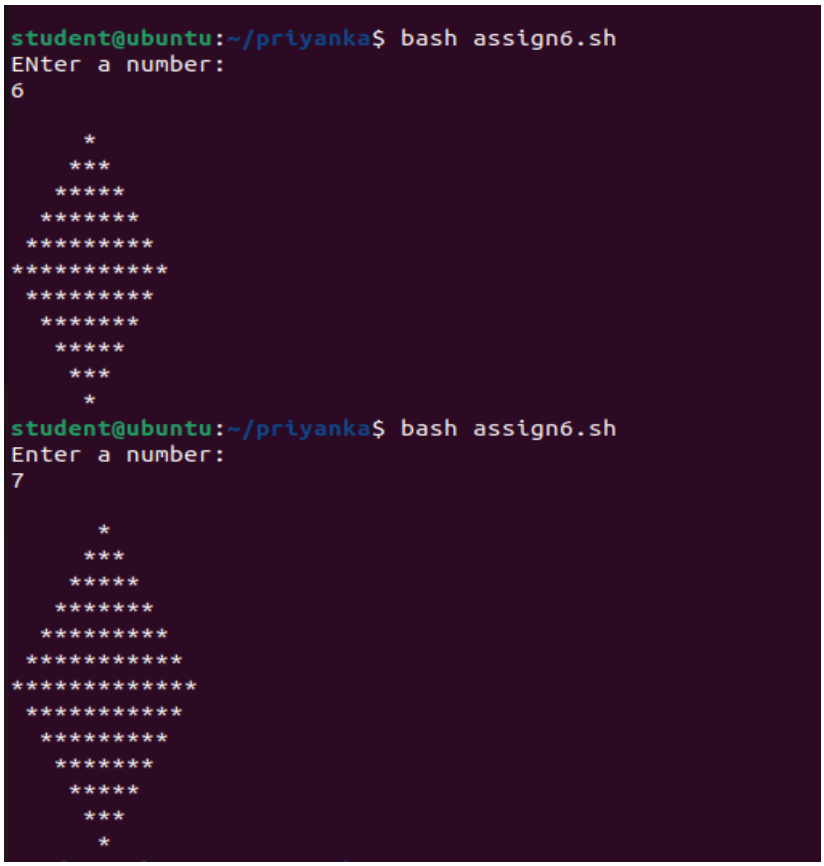
```
echo "Enter the number of rows"
read rows

i=1
while [ $i -le $rows ]
do
    j=1
    while [ $j -le `expr $rows - $i` ]
    do
        echo -n " "
        j=`expr $j + 1`
    done
    k=1
    while [ $k -le `expr 2 \* $i - 1` ]
    do
        echo -n "*"
        k=`expr $k + 1`
    done
    echo
    i=`expr $i + 1`
done
i=`expr $rows - 1`
while [ $i -ge 1 ]
do
    j=1
    while [ $j -le `expr $rows - $i` ]
    do
        echo -n " "
        j=`expr $j + 1`
    done
    k=1
    while [ $k -le `expr 2 \* $i - 1` ]
    do
        echo -n "*"
        k=`expr $k + 1`
    done
    echo
    i=`expr $i - 1`
done

```

Name: Priyanka Gupta **PRN: 22070122157**

output:



2. Shell script to using for loop to print the pattern:

Name: Priyanka Gupta **PRN: 22070122157**

numberWhile.sh

```
echo "Enter the number of rows"
read rows

i=1
while [ $i -le $rows ]
do
    j=1
    while [ $j -le $i ]
    do
        echo -n "$i"
        j=`expr $j + 1`
    done
    echo
    i=`expr $i + 1`
done

i=`expr $rows - 1`
while [ $i -ge 1 ]
do
    j=1
    while [ $j -le $i ]
    do
        echo -n "$i"
        j=`expr $j + 1`
    done
    echo
    i=`expr $i - 1`
done

```

Name: Priyanka Gupta **PRN: 22070122157**

output:

```
student@ubuntu:~/priyanka$ bash assign6.sh
Enter max number of stars:
5

1
22
333
4444
55555
4444
333
22
1

student@ubuntu:~/priyanka$ bash assign6.sh
Enter max number of stars:
8

1
22
333
4444
55555
666666
7777777
88888888
7777777
666666
55555
4444
333
22
1
```

Conclusion:

Hence, we learned how to use while and for loops to form different patterns such as star and numeric patterns.