|  |  |
| --- | --- |
|  | **Experiment No : 4 Date :** |
|  |  |
| **Title** | **Introduction to Shell Script – loops** |
|  |  |
| **Aim** | Write a shell script for the following   1. Write shell script to find reverse of 5 digit number using while loop 2. Write shell script to find factorial of number using while Loop 3. Write shell script to generate Fibo series upto the limit entered by user(for and while) 4. Write shell script to generate prime number between limit specified by user using for loop 5. Write shell script to generate table of the number specified by the user using for in Loop 6. Write shell script to generate even number between limit specified by user 7. Write shell script to Generate sum of N numbers. Read N from user and until loop 8. Write shell script to display 4x4 matrix and read data from user from keyboard. Use for Loop |
|  |  |
| **Hardware**  **Requirement** | Personal Computer |
|  |  |
| **Software**  **Requirement** | Linux Operating System(Ubuntu 16.04) , Shell-Interpreter  Nano or Vi or Vim or gedit text editor |
|  |  |
| **Theory** | **Looping Statements in Shell Scripting:** There are total 3 looping statements which can be used in bash programming   1. while statement 2. for statement 3. until statement   To **alter the flow of loop statements**, two commands are used they are,   1. break 2. continue   Their descriptions and syntax are as follows:   **while statement**  Here command is evaluated and based on the result loop will executed, if command raise to false then loop will be terminated   ***Syntax***    while [ condition ]  do  command1  command2  done    **for statement**  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 word1 to wordN are sequences of characters separated by spaces (words). Each time the for loop executes, the value of the variable var is set to the next word in the list of words, word1 to wordN.   ***Syntax***    for var in list  do  command 1  command 2  done    **until statement**  The until loop is executed as many as times the condition/command evaluates to false. The loop terminates when the condition/command becomes true.   ***Syntax***    until [ conditional statement ]  do  command1  command2  done |
|  |  |
| **Script Statement** | Write shell script to find reverse of 5 digit number using while loop |
|  |  |
| **Script Code** | #!/bin/bash  # EXP ID  # > 4.1  # AIM  # > Write shell script to find reverse of 5 digit number using while loop  # CODE  read -p "Enter a 5-digit number: " number  if [ ${#number} -ne 5 ]; then  echo "Please enter a valid 5-digit number."  exit 1  fi  rev=0  while [ $number -gt 0 ]; do  digit=$((number % 10))  rev=$((rev \* 10 + digit))  number=$((number / 10))  done  echo "Reversed Number = $rev" |
|  |  |
| **Output** |  |
|  |  |
| **Script Statement** | Write shell script to find factorial of number using while Loop |
|  |  |
| **Script Code** | #!/bin/bash  # EXP ID  # > 4..2  # AIM  # > Write shell script to find factorial of number using while Loop  # CODE  read -p "Enter number: " n  ans=1  while [ $n -ge 1 ]; do  ans=$((ans \* n))  n=$((n - 1))  done  echo "$ans" |
|  |  |
| **Output** |  |
| **Script Statement** | Write shell script to generate Fibo series upto the limit entered by user(for) |
| **Script Code** | #!/bin/bash  # EXP ID  # > 4.3.1  # AIM  # > Write shell script to generate Fibo series till the limit entered by user(for and while)  # CODE  read -p "Enter the number of terms for Fibonacci series: " n  if [ $n -lt 1 ]; then  echo "Please enter a valid positive number."  exit 1  fi  a=0  b=1  echo "Fibonacci series up to $n terms:"  for ((i = 0; i < n; i++)); do  echo -n "$a "  next=$((a + b))  a=$b  b=$next  done  echo |
| **Output** |  |
| **Script Statement** | Write shell script to generate Fibo series upto the limit entered by user(while) |
| **Script Code** | #!/bin/bash  # EXP ID  # > 4.3.2  # AIM  # > Write shell script to generate Fibo series till the limit entered by user(for and while)  # CODE  read -p "Enter the number of terms for Fibonacci series: " n  if [ $n -lt 1 ]; then  echo "Please enter a valid positive number."  exit 1  fi  a=0  b=1  i=0  echo "Fibonacci series up to $n terms:"  while ((i < n)); do  echo -n "$a "  next=$((a + b))  a=$b  b=$next  i=$i+1  done |
| **Output** |  |
| **Script Statement** | Write shell script to generate prime number between limit specified by user using for loop |
| **Script Code** | #!/bin/bash  # EXP ID  # > 4.4  # AIM  # > Write shell script to generate prime number between limit specified by user using for loop  # CODE  read -p "Enter the lower bound of the range: " lower  read -p "Enter the upper bound of the range: " upper  for ((num = lower; num <= upper; num++)); do  is\_prime=true  for ((i = 2; i\*i <= num; i++)); do  if [ $((num % i)) -eq 0 ]; then  is\_prime=false  break  fi  done  if [ "$is\_prime" = true ]; then  echo -n "$num "  fi  done  echo |
| **Output** |  |
| **Script Statement** | Write shell script to generate table of the number specified by the user using for in Loop |
| **Script Code** | #!/bin/bash  # EXP ID  # > 4.5  # AIM  # > Write shell script to generate table of the number specified by the user using for in Loop  # CODE  read -p "Enter the number:" n  for((i=1;i<=10;i++))  do  result=$((n\*i))  echo $result  done |
| **Output** |  |
| **Script Statement** | Write shell script to generate even number between limit specified by user |
| **Script Code** | #!/bin/bash  # EXP ID  # > 4.6  # AIM  # > Write shell script to generate even number between limit specified by user  # CODE  read -p "Enter the lower bound of the range: " lower  read -p "Enter the upper bound of the range: " upper  for ((num = lower; num <= upper; num++));  do  if [ $((num % 2)) -eq 0 ]  then  echo -n "$num "  fi  done  echo |
| **Output** |  |
| **Script Statement** | Write shell script to Generate sum of N numbers. Read N from user and until loop |
| **Script Code** | #!/bin/bash  # EXP ID  # > 4.7  # AIM  # > Write shell script to Generate sum of N numbers. Read N from user and until loop  # CODE  read -p "Enter the limit: " n  sum=0  i=1  until [ $i -gt $n ]; do  sum=$((sum + i))  ((i++))  done  echo "The sum is $sum" |
| **Output** |  |
| **Script Statement** | Write shell script to display 4x4 matrix and read data from user from keyboard. Use for Loop |
| **Script Code** | #!/bin/bash  # EXP ID  # > 4.8  # AIM  # > Write shell script to display 4x4 matrix and read data from user from keyboard. Use for Loop  # CODE  echo "Enter elements for a 4x4 matrix:"  for ((row=0; row<4; row++)); do  for ((col=0; col<4; col++)); do  echo "Enter element for [$((row+1))][$((col+1))]: "  read matrix["$row,$col"]  done  done  # Display the entered matrix  echo -e "\nEntered Matrix:"  for ((row=0; row<4; row++)); do  for ((col=0; col<4; col++)); do  echo -n "${matrix[$row,$col]} "  done  echo  done |
| **Output** |  |
|  |  |
| **Conclusion** | Practiced loop statements in bash scripting. |
|  |  |
|  |  |
| **Signature** |  |
|  |  |
|  |  |
| **Grade** |  |
|  |  |
|  |  |
|  |  |
| **Date** |  |
|  |  |
|  |  |