|  |  |
| --- | --- |
|  | **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 word.   ***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** | echo "Enter a 5 digit number: ";  read num  reverse=0  while [ $num -gt 0 ]  do  remainder=$(( $num %10 ))  reverse=$(( $reverse \*10+$remainder ))  num=$(( $num / 10 ))  done  echo "Reversed number is:$reverse" |
|  |  |
| **Output** |  |
|  |  |
| **Script Statement** | Write shell script to find factorial of number using while Loop |
|  |  |
| **Script Code** | WHILE Loop-  echo "Enter a number"  read num  fact=1  while [ $num -gt 1 ]  do  fact=$((fact \* num)) #fact = fact \* num  num=$((num - 1)) #num = num - 1  done  echo $fact  FOR Loop-  echo "Enter a number"  read num  fact=1  for((i=2;i<=num;i++))  {  fact=$((fact \* i)) #fact = fact \* i  }  echo $fact |
|  |  |
| **Output** |  |
|  |  |
| **Script Statement** | Write shell script to generate Fibo series upto the limit entered by user(for and while) |
|  |  |
| **Script Code** | FOR Loop:  read -p "Enter the limit for Fibonacci series:" limit  a=0  b=1  echo -n "$a "  for((i=1;i<limit;i++))  do  echo -n "$b "  temp=$((a + b))  a=$b  b=$temp  done  echo “.”  WHILE Loop:  read -p "Enter the limit:" num  a=0  b=1  while [ $num != 0 ]  do  echo -n "$a "  temp=$((a + b))  a=$b  b=$temp  num=$((num-1))  done |
|  |  |
| **Output** |  |
| **Script Statement** | Write shell script to generate prime number between limit specified by user using for loop |
| **Script Code** | read -p "Enter the upper limit for prime numbers: " limit  echo "Prime numbers between 2 and $limit:"  for ((number = 2; number <= limit; number++)); do  is\_prime=true  for ((i = 2; i <= number / 2; i++)); do  if [ $((number % i)) -eq 0 ]; then  is\_prime=false  break  fi  done  if $is\_prime; then  echo -n "$number "  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** | read -p "Enter the number to generate table: " number  echo "Multiplication table for $number:"  for ((i = 1; i <= 10; i++));  do  result=$((number\*i))  echo "$number x $i=$result"  done |
|  |  |
| **Output** |  |
|  |  |
| **Script Statement** | Write shell script to generate even number between limit specified by user |
|  |  |
| **Script Code** | read -p "Enter the upper limit for even number: " limit  echo "Even numbers between 2 and $limit:"  for ((number = 2; number <= limit; number += 2));do  echo -n "$number "  done  echo |
| **Output** |  |
|  |  |
| **Script Statement** | Write shell script to Generate sum of N numbers. Read N from user and until loop |
|  |  |
| **Script Code** | read -p "Enter the limit: " n  sum=0  for ((i = 1; i < n; i++ ))  do  sum=$((sum+i))  done  echo "$sum" |
|  |  |
| **Output** |  |
|  |  |
| **Script Statement** | Write shell script to display 4x4 matrix and read data from user from keyboard. Use for Loop |
|  |  |
| **Script Code** | matrix=( )  rows=4  cols=4  echo "Enter data for the 4x4 matrix:"  for ((i=1; i<=$rows; i++))  do  for ((j=1; j<=$cols; j++))  do  index=$((i \* cols + j))  echo -n "Enter element at position [$i][$j]: "  read value  matrix[$index]=$value  done  done  echo -e "\nEntered 4x4 matrix:"  for ((i=1; i<=$rows; i++)); do  for ((j=1; j<=$cols; j++)); do  index=$((i \* cols + j))  echo -n "${matrix[$index]} "  done  echo ""  done |
|  |  |
| **Output** |  |
|  |  |
| **Conclusion** | In Conclusion, shell scripts help with numbers, like reversing them, finding factorials, making Fibonacci series, finding primes, and more, making math tasks simpler on the command line. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| **Signature** |  |
|  |  |
|  |  |
| **Grade** |  |
|  |  |
|  |  |
|  |  |
| **Date** |  |
|  |  |
|  |  |