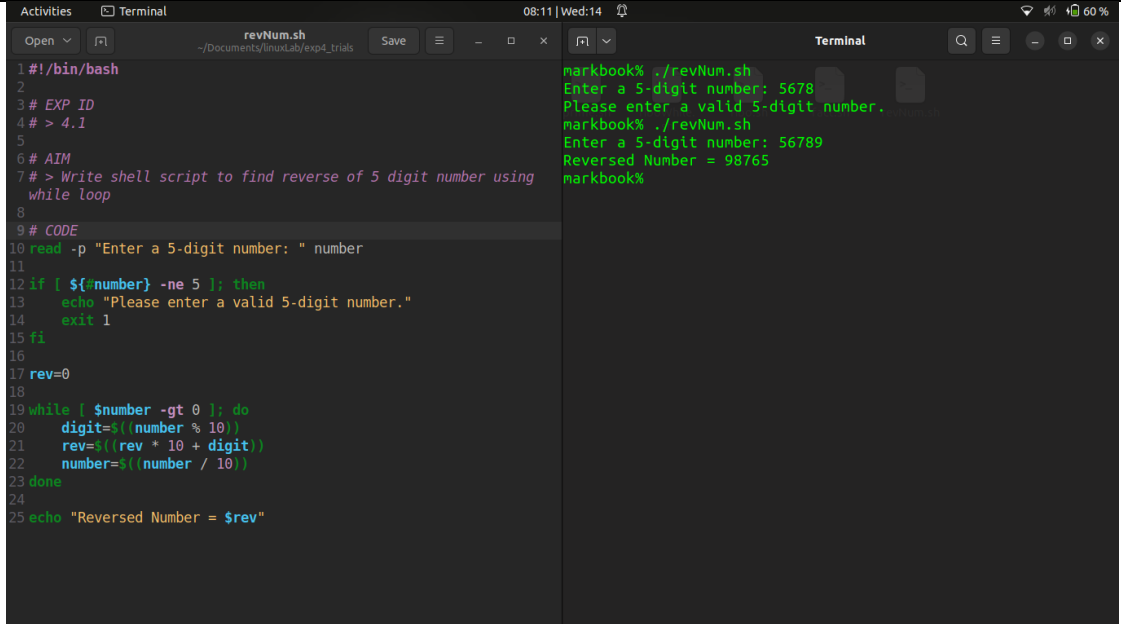
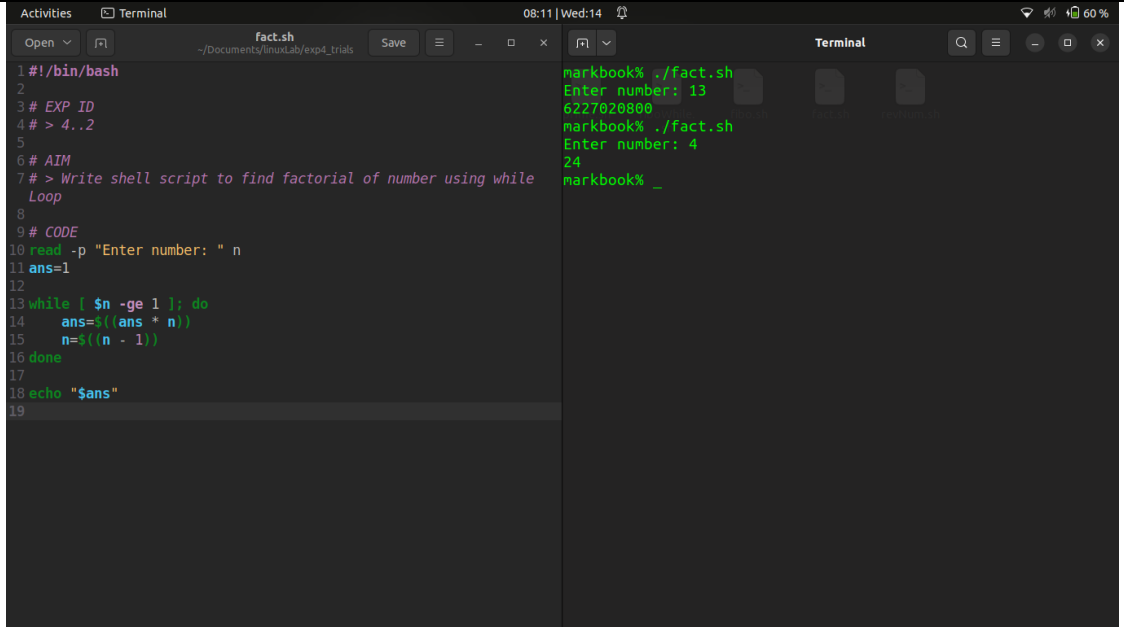


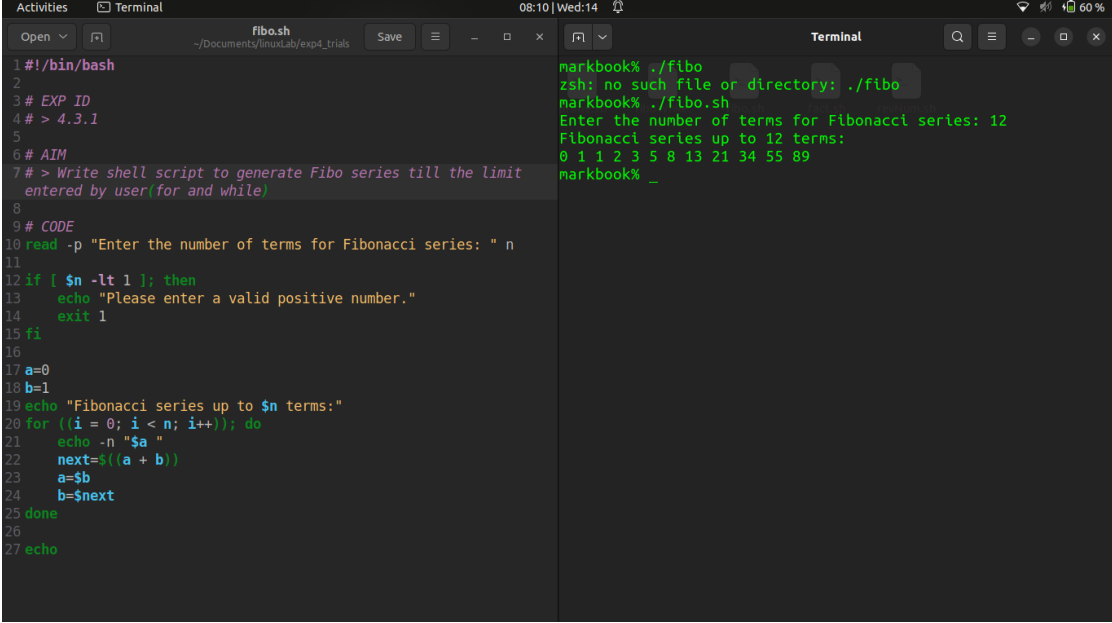
	Experiment No : 4	Date :
Title	Introduction to Shell Script – loops	
Aim	<p>Write a shell script for the following</p> <ol style="list-style-type: none"> Write shell script to find reverse of 5 digit number using while loop Write shell script to find factorial of number using while Loop Write shell script to generate Fibo series upto the limit entered by user(for and while) Write shell script to generate prime number between limit specified by user using for loop Write shell script to generate table of the number specified by the user using for in Loop Write shell script to generate even number between limit specified by user Write shell script to Generate sum of N numbers. Read N from user and until loop 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	<p>Looping Statements in Shell Scripting: There are total 3 looping statements which can be used in bash programming</p> <ol style="list-style-type: none"> while statement for statement until statement <p>To alter the flow of loop statements, two commands are used they are,</p> <ol style="list-style-type: none"> break continue <p>Their descriptions and syntax are as follows:</p> <p><u>while statement</u></p> <p>Here command is evaluated and based on the result loop will executed, if</p>	

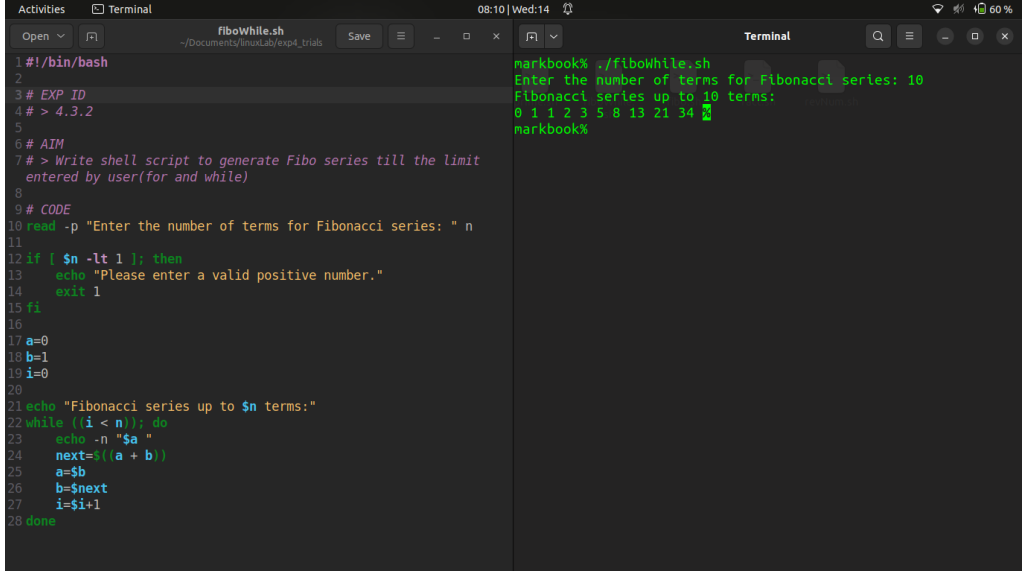
	<p>command raise to false then loop will be terminated</p> <p>Syntax</p> <pre>while [condition] do command1 command2 done</pre> <p><u>for statement</u></p> <p>The for loop operate on lists of items. It repeats a set of commands for every item in a list.</p> <p>Here var is the name of a variable and word1 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.</p> <p>Syntax</p> <pre>for var in list do command 1 command 2 done</pre> <p><u>until statement</u></p> <p>The until loop is executed as many as times the condition/command evaluates to false. The loop terminates when the condition/command becomes true.</p> <p>Syntax</p> <pre>until [conditional statement] do command1 command2 done</pre>
Script Statement	Write shell script to find reverse of 5 digit number using while loop
Script Code	#!/bin/bash

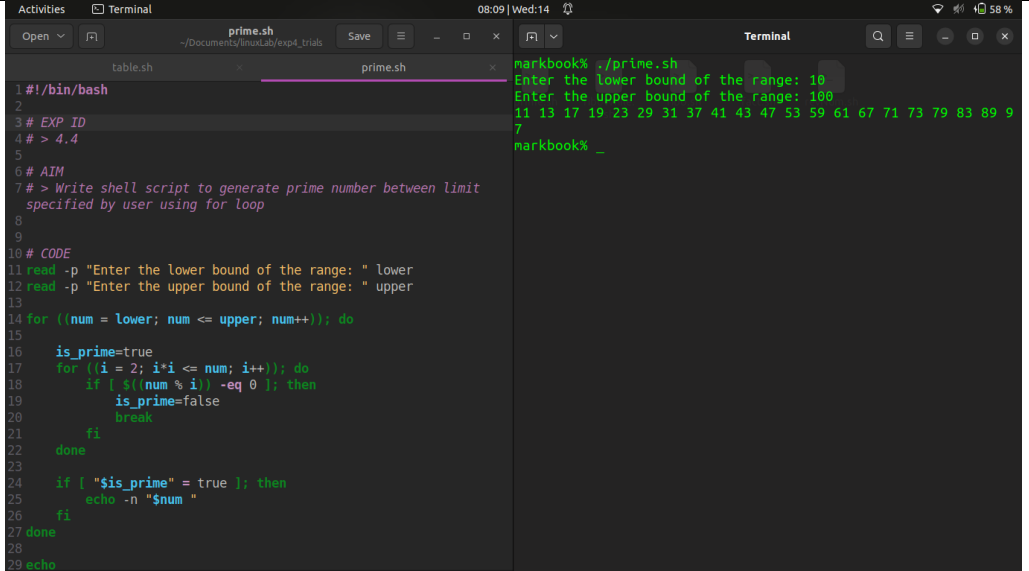
	<pre> # EXP ID # > 4.1 # AIM # > Write shell script to find reverse of 5 digit number using while loop # CODE read -p "Enter a 5-digit number: " number if [\${#number} -ne 5]; then echo "Please enter a valid 5-digit number." exit 1 fi rev=0 while [\$number -gt 0]; do digit=\$((number % 10)) rev=\$((rev * 10 + digit)) number=\$((number / 10)) done echo "Reversed Number = \$rev" </pre>
Output	 <pre> Activities Terminal 08:11 Wed:14 Open revNum.sh ~/Documents/tnuLab/exp4_trials Save 1 #!/bin/bash 2 3 # EXP ID 4 # > 4.1 5 6 # AIM 7 # > Write shell script to find reverse of 5 digit number using while loop 8 9 # CODE 10 read -p "Enter a 5-digit number: " number 11 12 if [\${#number} -ne 5]; then 13 echo "Please enter a valid 5-digit number." 14 exit 1 15 fi 16 17 rev=0 18 19 while [\$number -gt 0]; do 20 digit=\$((number % 10)) 21 rev=\$((rev * 10 + digit)) 22 number=\$((number / 10)) 23 done 24 25 echo "Reversed Number = \$rev" markbook% ./revNum.sh Enter a 5-digit number: 5678 Please enter a valid 5-digit number. markbook% ./revNum.sh Enter a 5-digit number: 56789 Reversed Number = 98765 markbook% </pre>

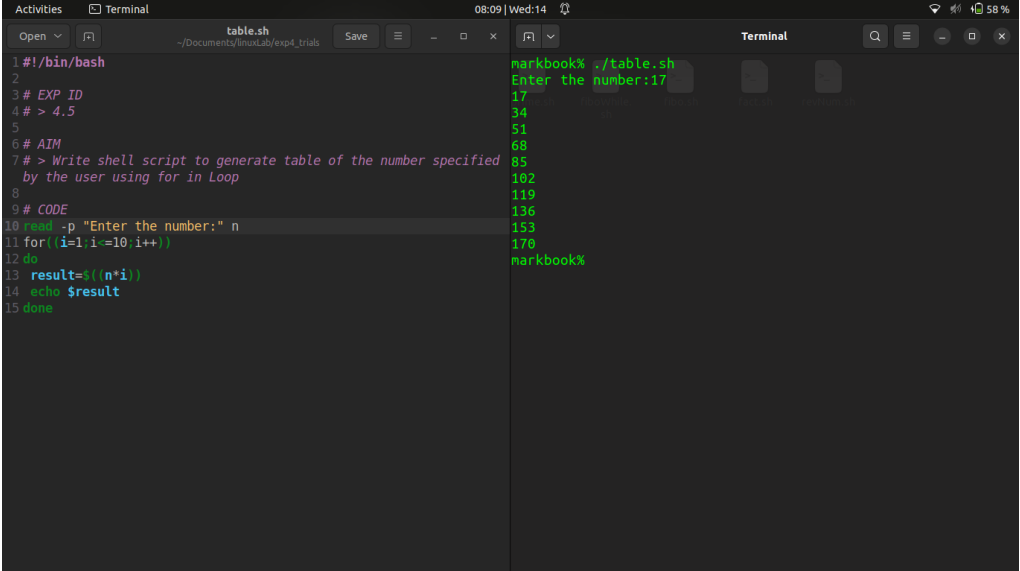
Script Statement	Write shell script to find factorial of number using while Loop
Script Code	<pre>#!/bin/bash # EXP ID # > 4..2 # AIM # > Write shell script to find factorial of number using while Loop # CODE read -p "Enter number: " n ans=1 while [\$n -ge 1]; do ans=\$((ans * n)) n=\$((n - 1)) done echo "\$ans"</pre>
Output	 <p>The screenshot shows a terminal window with the following content:</p> <pre>markbook% ./fact.sh Enter number: 13 6227020800 markbook% ./fact.sh Enter number: 4 24 markbook% _</pre>
Script Statement	Write shell script to generate Fibo series upto the limit entered by user(for)

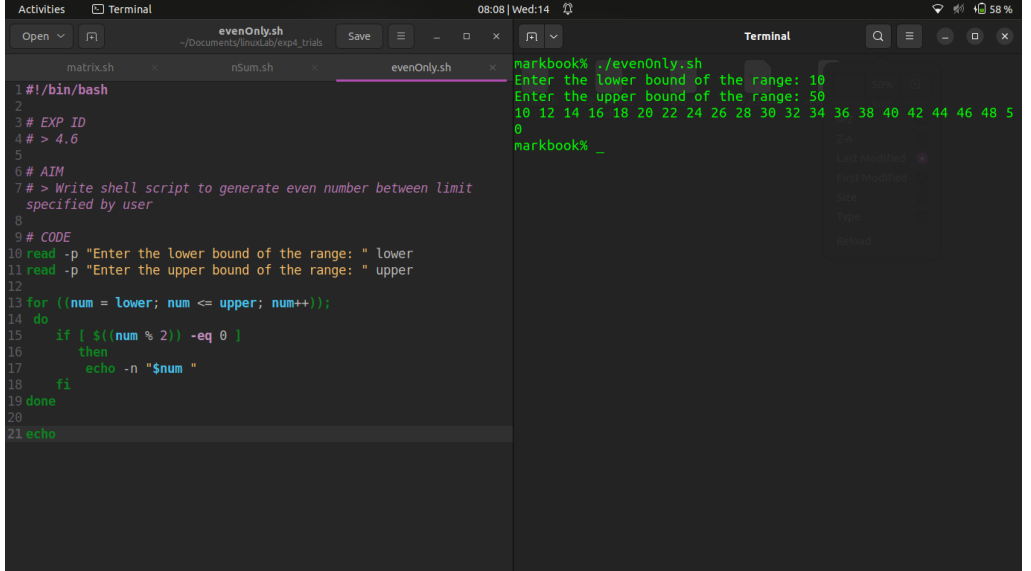
Script Code	<pre>#!/bin/bash # EXP ID # > 4.3.1 # AIM # > Write shell script to generate Fibo series till the limit entered by user(for and while) # CODE read -p "Enter the number of terms for Fibonacci series: " n if [\$n -lt 1]; then echo "Please enter a valid positive number." exit 1 fi a=0 b=1 echo "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</pre>
--------------------	---

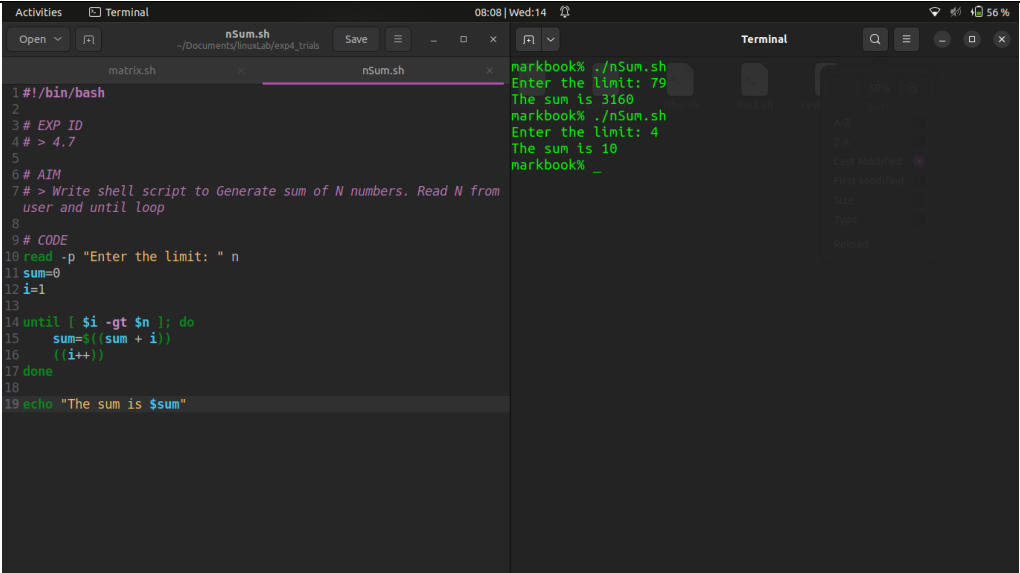
Output	 <pre> 1 #!/bin/bash 2 3 # EXP ID 4 # > 4.3.1 5 6 # AIM 7 # > Write shell script to generate Fibo series till the limit entered by user(for and while) 8 9 # CODE 10 read -p "Enter the number of terms for Fibonacci series: " n 11 12 if [\$n -lt 1]; then 13 echo "Please enter a valid positive number." 14 exit 1 15 fi 16 17 a=0 18 b=1 19 echo "Fibonacci series up to \$n terms:" 20 for ((i = 0; i < n; i++)); do 21 echo -n "\$a " 22 next=\$((a + b)) 23 a=\$b 24 b=\$next 25 done 26 27 echo </pre> <pre> markbook% ./fibo zsh: no such file or directory: ./fibo markbook% ./fibo.sh Enter the number of terms for Fibonacci series: 12 Fibonacci series up to 12 terms: 0 1 1 2 3 5 8 13 21 34 55 89 markbook% _ </pre>
Script Statement	<p>Write shell script to generate Fibo series upto the limit entered by user(while)</p>
Script Code	<pre> #!/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 </pre>

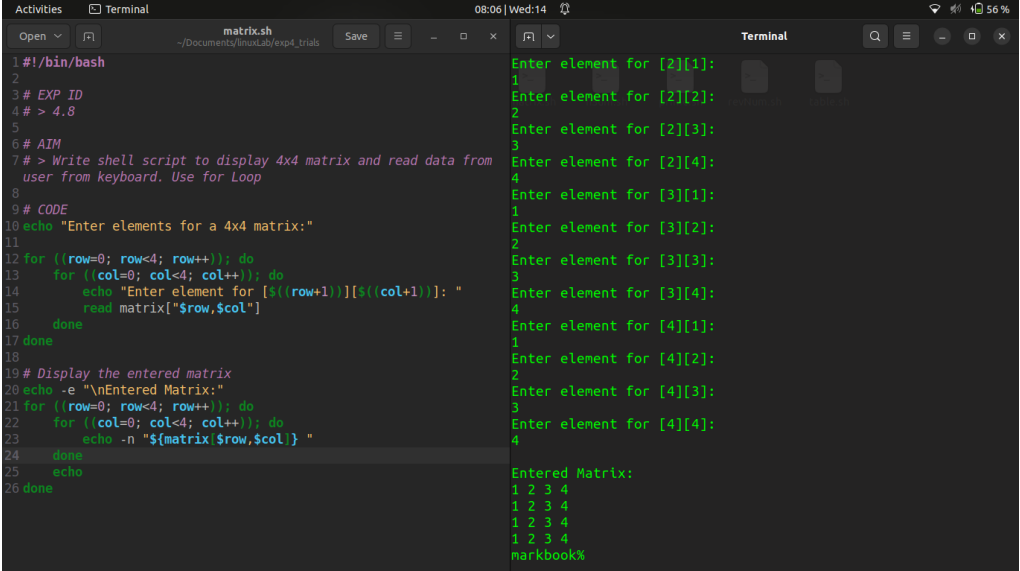
	<pre> echo "Fibonacci series up to \$n terms:" while ((i < n)); do echo -n "\$a " next=\$((a + b)) a=\$b b=\$next i=\$((i + 1)) done </pre>
Output	 <pre> #!/bin/bash 2 3 # EXP ID 4 # > 4.3.2 5 6 # AIM 7 # > Write shell script to generate Fibo series till the limit entered by user(for and while) 8 9 # CODE 10 read -p "Enter the number of terms for Fibonacci series: " n 11 12 if [\$n -lt 1]; then 13 echo "Please enter a valid positive number." 14 exit 1 15 fi 16 17 a=0 18 b=1 19 i=0 20 21 echo "Fibonacci series up to \$n terms:" 22 while ((i < n)); do 23 echo -n "\$a " 24 next=\$((a + b)) 25 a=\$b 26 b=\$next 27 i=\$((i + 1)) 28 done </pre> <pre> markbook% ./fiboWhile.sh Enter the number of terms for Fibonacci series: 10 Fibonacci series up to 10 terms: 0 1 1 2 3 5 8 13 21 34 markbook% </pre>
Script Statement	Write shell script to generate prime number between limit specified by user using for loop
Script Code	<pre> #!/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 </pre>

	<pre> 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 </pre>
Output	 <pre> Activities Terminal 08:09 Wed:14 prime.sh Save ~/Documents/linuxLab/exp4_trials table.sh prime.sh 1 #!/bin/bash 2 3 # EXP ID 4 # > 4.4 5 6 # AIM 7 # > Write shell script to generate prime number between limit specified by user using for loop 8 9 10 # CODE 11 read -p "Enter the lower bound of the range: " lower 12 read -p "Enter the upper bound of the range: " upper 13 14 for ((num = lower; num <= upper; num++)); do 15 16 is_prime=true 17 for ((i = 2; i*i <= num; i++)); do 18 if [\$((num % i)) -eq 0]; then 19 is_prime=false 20 break 21 fi 22 done 23 24 if ["\$is_prime" = true]; then 25 echo -n "\$num " 26 fi 27 done 28 29 echo markbook% ./prime.sh Enter the lower bound of the range: 10 Enter the upper bound of the range: 100 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 9 7 markbook% _ </pre>
Script Statement	Write shell script to generate table of the number specified by the user using for in Loop
Script Code	<pre> #!/bin/bash # EXP ID # > 4.5 # AIM # > Write shell script to generate table of the number specified by the user using for in Loop </pre>

	<pre># CODE read -p "Enter the number:" n for((i=1;i<=10;i++)) do result=\$((n*i)) echo \$result done</pre>
Output	 <p>The screenshot shows a terminal window with the following content:</p> <pre>#!/bin/bash 2 3 # EXP ID 4 # > 4.5 5 6 # AIM 7 # > Write shell script to generate table of the number specified by the user using for in Loop 8 9 # CODE 10 read -p "Enter the number:" n 11 for((i=1;i<=10;i++)) 12 do 13 result=\$((n*i)) 14 echo \$result 15 done</pre> <p>The user input '17' is shown, followed by the output of the script:</p> <pre>markbook% ./table.sh Enter the number:17 17 34 51 68 85 102 119 136 153 170 markbook%</pre>
Script Statement	Write shell script to generate even number between limit specified by user
Script Code	<pre>#!/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</pre>

	<pre> if [\$((num % 2)) -eq 0] then echo -n "\$num " fi done echo </pre>
Output	 <pre> markbook% ./evenOnly.sh Enter the lower bound of the range: 10 Enter the upper bound of the range: 50 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50 markbook% _ </pre>
Script Statement	Write shell script to Generate sum of N numbers. Read N from user and until loop
Script Code	<pre> #!/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)) </pre>

	<pre>((i+ +)) done echo "The sum is \$sum"</pre>
Output	
Script Statement	Write shell script to display 4x4 matrix and read data from user from keyboard. Use for Loop
Script Code	<pre>#!/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</pre>

	<pre>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</pre>
Output	 <pre>Activities Terminal 08:06 Wed:14 matrix.sh ~/Documents/linuxLab/exp4_trials Save 1 #!/bin/bash 2 3 # EXP ID 4 # > 4.8 5 6 # AIM 7 # > Write shell script to display 4x4 matrix and read data from user from keyboard. Use for Loop 8 9 # CODE 10 echo "Enter elements for a 4x4 matrix:" 11 12 for ((row=0; row<4; row++)); do 13 for ((col=0; col<4; col++)); do 14 echo "Enter element for [\${row+1}][\${col+1}]: " 15 read matrix["\$row,\$col"] 16 done 17 done 18 19 # Display the entered matrix 20 echo -e "\nEntered Matrix:" 21 for ((row=0; row<4; row++)); do 22 for ((col=0; col<4; col++)); do 23 echo -n "\${matrix[\$row,\$col]} " 24 done 25 echo 26 done Enter element for [2][1]: 1 Enter element for [2][2]: 2 Enter element for [2][3]: 3 Enter element for [2][4]: 4 Enter element for [3][1]: 1 Enter element for [3][2]: 2 Enter element for [3][3]: 3 Enter element for [3][4]: 4 Enter element for [4][1]: 1 Enter element for [4][2]: 2 Enter element for [4][3]: 3 Enter element for [4][4]: 4 Entered Matrix: 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 markbook%</pre>
Conclusion	Practiced loop statements in bash scripting.
Signature	
Grade	
Date	