**Question 1.** Write a bash script to get the current date, time, username, home directory and current working directory.

Anuj’s Approach : He saved the date, time, username ,home directory , and current directory into a variable then did the printing .

| curr\_date=$(date +"Year: %Y, Month: %m, Day: %d")   #Current Time curr\_time=$(date +"%T")  #Current User curr\_user=$(whoami)   #Home Directory home\_directory=$(echo $HOME)   #Working Directory current\_directory=$(pwd)  # Printing the fetched variables echo "Current date is: $curr\_date" echo "Current time is: $curr\_time" echo "Current User is: $curr\_user" echo "Home directory is located at: $home\_directory" echo "Current Working directory is located at: $current\_directory" |
| --- |

Raj Keshav’s Approach : Almost the same .

Shishir’s Approach : Almost the same

**Question 2.** Write a bash script (name Table.sh) to print the Table of a number by using a while loop. It should support the following requirements

● The script should accept the input from the command line.

● If you don’t input any data, then display an error message to execute the script correctly.

Anuj’s Approach : He is taking multiple numbers as input . So while loop runs till total number of

Numbers and then for loop running for each numbers table.

| #!/bin/bash  if [ $# -eq 0 ]; then  echo "Please enter arguments to generate tables!!!"  exit 1 fi  i=1; j=$#; while [ $i -le $j ]  do  n=$1   c=1  echo "Table of $n:"   while [ $c -le 10 ]   do  result=$(( $n \* $c ))  echo "$n x $c = $result"   c=$(( $c + 1 ))  done  shift 1;  i=$((i+1))  echo "" done  exit 0 |
| --- |

Raj Keshav’s Approach :

in the for loop he has used ‘lt : less than ‘ whereas I have used ‘le:less than or equal to”. So he has used while [$i,lt, 11] and i has used while [$i,le,10] both the loops will run 10 times only the syntax is different.

Shishir Approach:

run a loop from I=1 to I<=10, to print the table of n given as input by the user.

**Question 3.** Write a Function in bash script to check if the number is prime or not? It should support the following requirement

● The script should accept the input from the Use

Anuj’s Approach : He has used a separate prime function() to check whether a number is prime is not where i have done it using in the main function itself .

| function check\_prime() {  n=$1   if [ $n -lt 2 ]; then  echo "$n is not a prime number."  return  fi  for (( i=2; i<$(($n/2+1)); i++ ))  do  if [ $(($n%$i)) -eq 0 ]; then  echo "$n is not a prime number."  return  fi  done  echo "$n is a prime number." } |
| --- |

Rajkeshav’s Approach **:** He has used a while loop to iterate through the till number/2 starting from 2 whereas i have used a for loop.Another change is he is check if the number is divisible by i in the if statement directly whereas i have firstly save into ans variable the used it in if statement to look concise.He is using a flag variable to check if number is prime or not whereas i print in directly. Rest of the code is the same .

Shishir’s Approach : Run a loop from i=2 to I<=sqrt(n), and check if n%I == 0 for any i. If there exists any I, which divides n, then n Is non prime, else n is prime .

**Question 4.** Create a bash script that supports the following requirement.   
 ● Create a folder ‘Assignment’.   
 ● Create a file ‘File1.txt’ inside ‘Assignment’ Folder.   
 ● Copy all the content of Table.sh(2nd script) in ‘File1.txt’ without using ‘cp’ and ‘mv’ command.   
 ● Append the text Welcome to Sigmoid’ to the ‘File1.txt’ file.

● List all the directories and files present inside Desktop Folder.

Anuj’s Approach :

For copying the contents of the Table.sh file he has used >> whereas i have used > . Both the codes do the job here ; the basic difference is if there is >> appends whereas > overwrites. So if my file is already having some code ,it will overwrite it with Table.sh contents.

| mkdir ~/Desktop/Assignment echo "Assignment Folder created on the Desktop."  touch ~/Desktop/Assignment/File1.txt echo "File1.txt created in Assignment Folder."  cat ~/Desktop/Table.sh >> ~/Desktop/Assignment/File1.txt echo "Data in Table.sh copied to File1.txt."  echo "Welcome to Sigmoid" >> ~/Desktop/Assignment/File1.txt  echo "Folders in the Desktop:" ls -la ~/Desktop/ |
| --- |

Raj keshav Approach:

So for copying the contents of the Table.sh file he has not directly copied it. Firstly he created a variable then saved the address of the Table.sh then using ‘>’ he has copied.

Shishir’s Approach

For creating file he is using cd command to move into assignment folder and then creating the file using touch but we can do it directly in one step touch /Assignment/file.txt

* **Question 5.** You have given an array. Using Bash script, print its length, maximum element and minimum element.

Anuj’s Approach : So for calculating the min and max from the given array he is using two different for loop one for max and another for min, which is not needed . We can calculate it into the single for loop which has been implemented by me.

| arr=( 8 4 2 1 3 0 2) echo "Length of the array - ${#arr[@]} "  # Using a for loop to find the max and min element. echo "Max and Min of the array using for loop"  max=${arr[0]} #Assuming first element as max for n in "${arr[@]}" ; do   if [ $n -ge $max ]; then    max=$n   fi done echo "Maximum in the array - $max" min=${arr[0]} #Assuming first element as min  for n in "${arr[@]}" ; do   if [ $n -le $min ]; then    min=$n   fi  done echo "Minimum in the array - $min" |
| --- |

Raj Keshav Approach : Almost identical barring the variable names.

Shishir’s Approach : Just like Anuj , Shishir is also using two separate for loops for calculating maximum and minimum elements in the array which can be done in a single for loop.