**LINUX AND SHELL PROGRAMMING LAB**

**PROGRAM OBJECTIVE AND OUTCOME**

**PROGRAM** : B.Sc. Information Technology

**OBJECTIVE**

1. Apply the knowledge of Technology, Mathematics, Networks and computing in the core information technologies.
2. Identify, design, and analyze complex computer systems and implement and interpret the results from those systems
3. Analyze the local and global impact of computing on individuals, organizations, and society.
4. One of the most important benefits of taking computer courses is that the students will have more jobs available to them.
5. The types of new jobs that will be available depend on what kind of courses they take, but every group of courses will open up new opportunities.

**COURSE OBJECTIVE AND OUTCOME**

**COURSE OBJECTIVE**

1. In this Linux Shell Programming Lab the students will get clear depth knowledge in linux environment and basic commands.
2. The Basic concepts like File commands, Pipes, Filters, Loops and executing baisc C programs are clear in this course.

**COURSE OUTCOME**

1. To understand the basic concepts of single & multiuser Operating System, basic structure of LINUX kernel and its subsystems

2. To study process control subsystem, process scheduling paradigms and different types of scheduling employed in LINUX

3. Understand Process Control subsystem, types of scheduling and memory management policies

4. Execute various types of commands on the standard shell viz. basic commands, directory and file related, pipe and filter related, process related, user communication related and the system administration related commands

5. Implement shell scripts using this editor involving decision control, looping and control flow statements.

**LIST OF PROGRAMS**

1. Write a shell script to stimulate the file commands: rm, cp, cat, mv, cmp, wc, split, diff.
2. Write a shell script to show the following system configuration:
3. currently logged user and his log name
4. current shell , home directory , Operating System type , current Path setting , current working directory
5. show currently logged number of users, show all available shells
6. show CPU information like processor type , speed
7. show memory information
8. Write a Shell Script to implement the following: pipes, Redirection and tee commands.
9. Write a shell script for displaying current date, user name, file listing and directories by

getting user choice.

1. Write a shell script to implement the filter commands.
2. Write a shell script to remove the files which has file size as zero bytes.
3. Write a shell script to find the sum of the individual digits of a given number.
4. Write a shell script to find the greatest among the given set of numbers using command

line arguments.

1. Write a shell script for palindrome checking.
2. Write a shell script to print the multiplication table of the given argument using for

loop.

**PREREQUISITES**

Software and Pre require knowledge

**Software Requirements:**

* + Linux Operating System
  + Telnet or Putty
  + Browsers

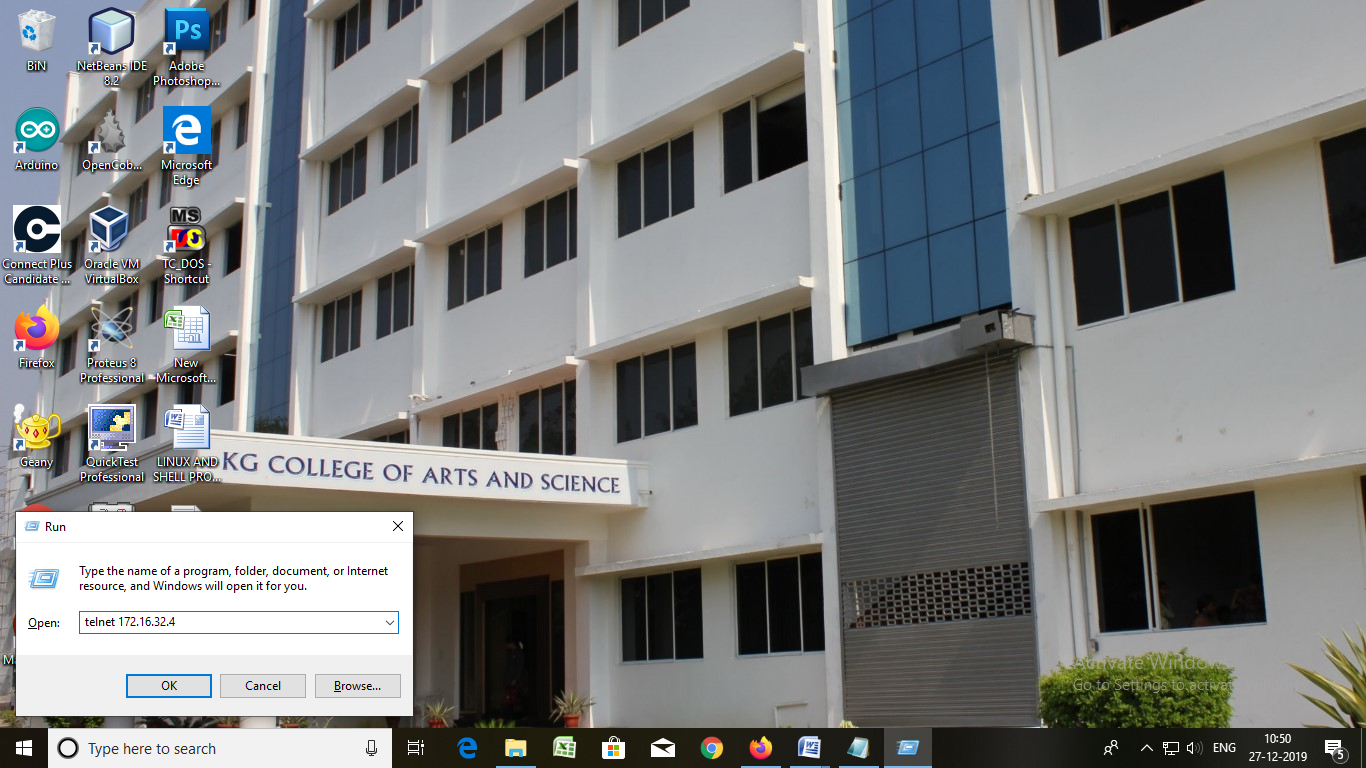
**Basic Skill Set Required:**

* + To open a file and command prompt.
  + Typing skills.
  + Basic Knowledge about Command Line Interface (CLI).
  + Fundamental concepts in C language.

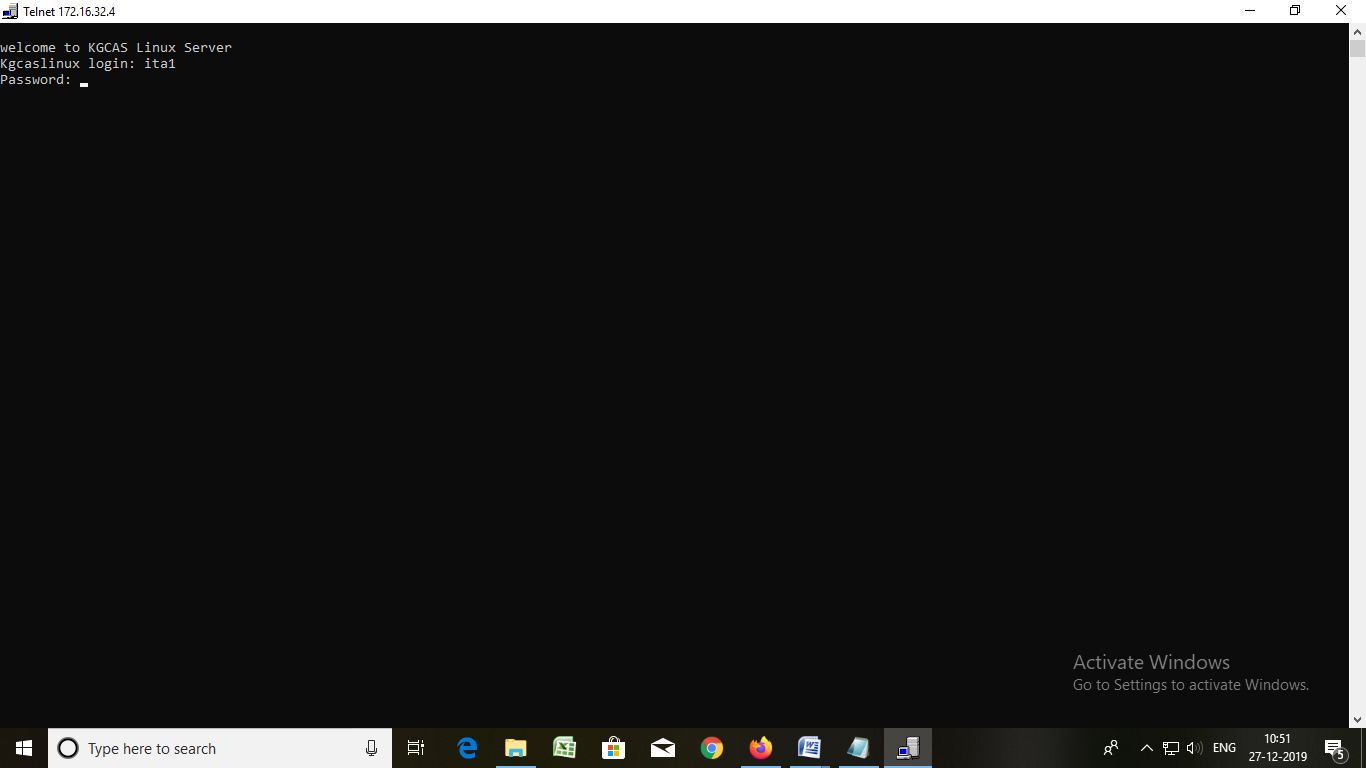
**GUIDE TO UTILIZE TOOL (MANUAL FOR TOOL)**

* **Steps for accessing the tool**

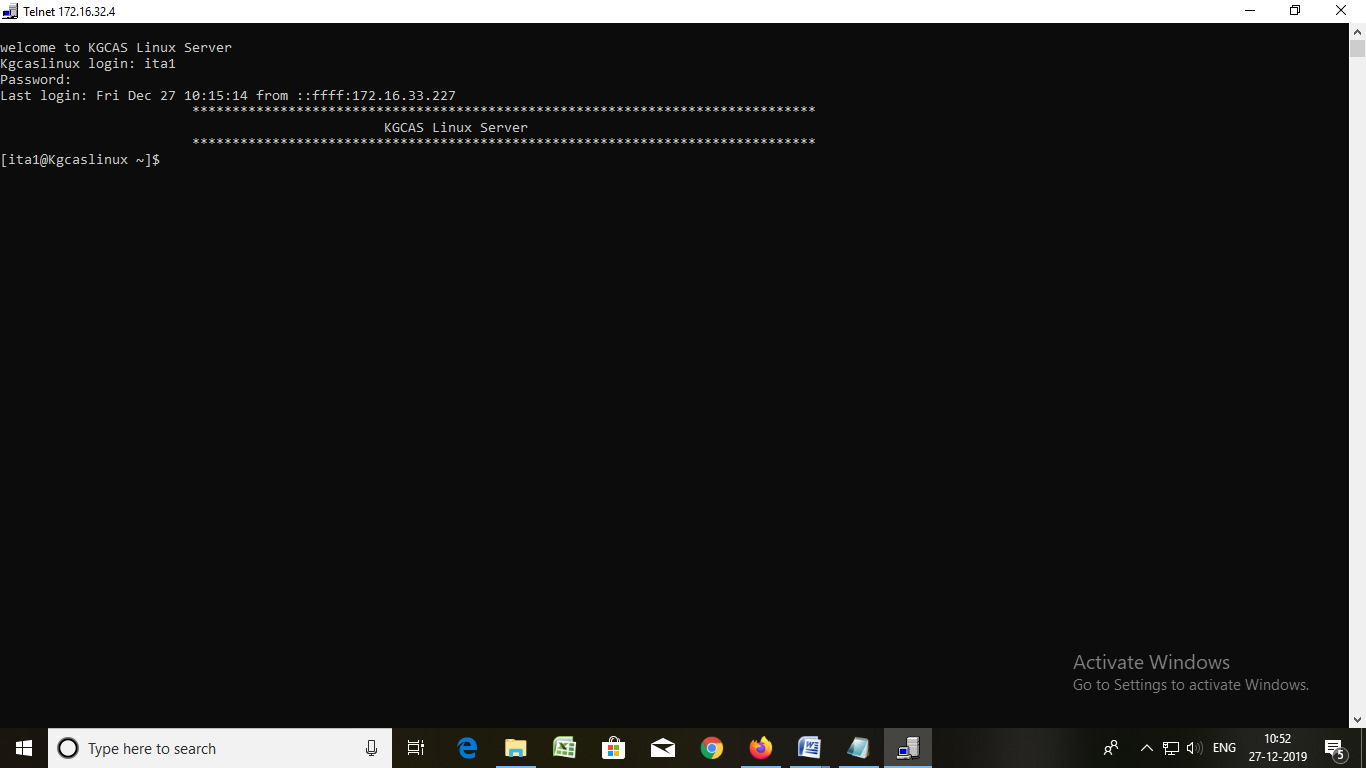
1. Going to Run Command and access telnet by typing Telnet 172.16.32.4



1. Type the Correct UserName and Password for logging to Linux



1. After Successful Logging it will connected Linux Server Machine.



**MANUAL FOR PROGRAMS**

**PROGRAM 1:**

Basic Linux commands: rm, cp, cat, mv, cmp, wc, split, diff.

**PROCEDURE:**

**STEP 1:** Start the process

**STEP 2**: In the normal command prompt of Linux perform the operations for following commands.

**STEP 3:** Use Vi editor for creating files **Vi Filename** and use mkdir for creating directories **mkdir directoryname.**

**STEP 4:** Use cat command to create,append and display the file on the screen **cat >filename**  -to create **cat >>filename** -to append **cat filename** to display.

**STEP 5:** Use wc command to count the lines,character and bytes in the respective file like **wc filename.**

**STEP 6:** Use cp command to copy the old file into new file **cp filename (old) filename 2 (new).**

**STEP 7:** Use **cmp** command to compare **rm** command to remove mv command to move and wc command to count the word from the file.

**STEP 8 :** Finally use **split** command to split the one phase into another and **diff** command is to show the difference between one file with another.

**STEP 9:** Type all this command in shell script file by creating vi filename.sh and execute it like by sh filename.sh.

**STEP 10** : Stop the process.

**SHELL SCRIPTS:**

ls **// To list all the files in the directory**

cp file1 file2 **// To copy a file**

cat file2 **// To View the Contents in file2**

mv file3 file4 **// To move files**

cat file4 **// Contents in file4**

wc file1 **// To see the word count**

wc -l file1 **// To see the line count**

wc -w file1 **// To see the words count**

wc -c file1 **// To see the character count**

split -3 file1 **// To split a file**

head -3 file1 // **To read a Initial lines of file**

tail -3 file1 // **To read a Last lines of file**

cmp file1 file2 // **To compare two files**

diff file1 file2 // **To differ two files**

rm file6 // **To remove a file6 from the directory**

rm -r \* // **To remove all files from the directory**

**OUTPUT**

[ita1@Kgcaslinux ~]$ ls

file1 file2 sun

[ita1@Kgcaslinux ~]$ cp file1 file3

[ita1@Kgcaslinux ~]$ cat file3

a

b

c

d

e

f

g

[ita1@Kgcaslinux ~]$ mv file2 file4

[ita1@Kgcaslinux ~]$ cat file4

sun

mon

tue

wed

thu

fri

sat

sun

[ita1@Kgcaslinux ~]$ wc file4

7 7 28 file4

[ita1@Kgcaslinux ~]$ wc -l file4

7 file4

[ita1@Kgcaslinux ~]$ wc -w file4

7 file4

[ita1@Kgcaslinux ~]$ wc -c file4

28 file4

[ita1@Kgcaslinux ~]$ split -3 file4

[ita1@Kgcaslinux ~]$ ls

file1 file3 file4 sun xaa xab xac

[ita1@Kgcaslinux ~]$ head -3 file4

sun

mon

tue

[ita1@Kgcaslinux ~]$ tail -2 file4

fri

sat

[ita1@Kgcaslinux ~]$ cmp file1 file4

file1 file4 differ: byte 1, line 1

[ita1@Kgcaslinux ~]$ diff file1 file4

1,26c1,7

< a

< b

< c

< d

< e

< f

< g

---

> sun

> mon

> tue

> wed

> thu

> fri

> sat

[ita1@Kgcaslinux ~]$ ls

file1 file3 file4 sun xaa xab xac

[ita1@Kgcaslinux ~]$ rm xac

[ita1@Kgcaslinux ~]$ ls

file1 file3 file4 sun xaa xab

[ita1@Kgcaslinux ~]$ ls

file1 file3 file4 sun xaa xab

[ita1@Kgcaslinux ~]$ rm -r \*

[ita1@Kgcaslinux ~]$ ls

**PROGRAM 2:**

Write a shell script to implement the user and System information by commands.

**PROCEDURE :**

**STEP 1**: start the process.

**STEP 2:** In the shell prompt window perform the following commands use **logname** to check log name of user, shell to check the following shell information.

**STEP 3:** type **ostype** command to check the ostype of Linux os.

**STEP 4:** type path to check the path for particular directories.

**STEP 5:** type **pwd** command to view the present working directory.

**STEP 6:** **is CPU** command to check the CPU information.

**STEP 7:** stop the process

**SHELL SCRIPTS :**

#!/bin/bash  
echo -e "username:”$USER  
echo -e "loginname:”$LOGNAME  
echo -e "currentshell:”$SHELL  
echo -e "homedirectory:$HOME  
echo -e "our pc os is:$OSTYPE  
echo -e "current directory:"pwd  
echo -e "system configuration (or) pc configuration:"lscpu  
echo -e "free memory space:"free -m  
**OUTPUT**

Tsa

Tsa

Cshshell

Command ! name from package ‘home(multiverse)

Command ! name from package ‘home(universe)

**PROGRAM 3 :**

Write a shell script to implement the following of pipes, redirection and the commands.

**PROCEDURE:**

**STEP 1 :** Start the process.

**STEP 2 :** use **( | )** pipe command to link one command with another (or) one operation.

**STEP 3 :** use **(>>)** command to transfer the file from one part into another.

**STEP 4 :** use **More** command to check the information on the screen.

**STEP 5 :** display the result on the screen.

**STEP 6 :**Save the process.

**STEP 7:**End the process

**SHELL SCRIPTS :**

(i) **Pipes**:

[tsa03@localhost ~]**$ ls -la |wc**

**Output:**

64 569 3333

(ii) \* **Append :**

**[syntax : $cat > >filename]**

[tsa@telnet ~]$ cat >> sample1

Always best learning institute

**After Appending:**

[tsa@telnet ~]$ cat sample1

welcome to tsa

Always best learning institute

**More:**

[tsa03@localhost ~]$ **ls -la | more**

total 504

drwx------ 4 tsa03 tsa03 4096 Feb 13 08:40 .

drwxr-xr-x 53 root root 4096 Feb 18 2002 ..

-rw-rw-r-- 1 tsa03 tsa03 24 Jan 23 01:33 aa

-rw-rw-r-- 1 tsa03 tsa03 8 Feb 5 23:35 ais

-rw-rw-r-- 1 tsa03 tsa03 16 Feb 5 23:35 aismin

-rw-rw-r-- 1 tsa03 tsa03 13 Feb 6 01:28 aisminaa

**PROGRAM 4**:

Write a shell script for displaying **current date user name, file listing and directories** using **switch case** statement.

**PROCEDURE :**

**STEP 1 :** Start the process.

**STEP 2 :** use case statement for performed an different action into an single prompt.

**STEP 3 :** declare **case** variables and case command for the program.

**STEP 4 :** if the choice is **1** then the current date will be displayed on the screen.

**STEP 5 :** if the choice is **2** then username will be shown .if the choice is three file can be listed and finally if the choice is four directories are displayed on the screen.

**STEP 6** : it none of the choice is met finally default case get executed on the screen.

**STEP 7 :** stop the process

**SHELL SCRIPTS :**

echo "1- who am i?"

echo "2- who is logged on?"

echo "3- date"

echo "4- calendar"

echo "5- Current directory"

echo "6- file listing"

echo "enter your choice:"

read n

case $n in

1) whoami;;

2) who;;

3) date;;

4) cal;;

5) pwd;;

6) ls;;

esac

**OUTPUT**

echo "1- who am i?"

echo "2- who is logged on?"

echo "3- date"

echo "4- calendar"

echo "5- Current directory"

echo "6- file listing"

echo "enter your choice:"

2

**tsa**

**PROGRAM 5:**

Shell script to implement filter commands.

**PROCEDURE:**

**STEP 1:** Start the process.

**STEP 2:** Create a file using vi editor.

**STEP 3:** Copy the files **/etc/passwd** to passwd file.

**STEP 4:** To display the lines containing the wordhome use **grep –n “root” passwd**

**STEP 5:** To display the no.of.lines containing the word root use **grep –c “root” passwd**

**STEP 6:** To display all the lines, words, characters in passwd file use **wc passwd**

**STEP 7:** To display all the lines that don’t match with the line root use **grep –v “root” passwd.**

**STEP 8:** To replace “:” with “\*” in the file passwd use **tr “:” “\*” <passwd>** statement.

**STEP 9:** To display the first column of the file passwd use cut **–d ‘:’ –f1 passwd**

**STEP 10:** The output will be displayed on the screen.

**STEP 11:** Stop the process.

**SHELL SCRIPTS :**

echo “filter cmd”

cp /etc/passwd passwd **// create the file passwd by copying from /etc/passwd**

echo “display no of lines, words and char in passwd file”

wc passwd **// display no of lines, words and char in passwd file**

echo “ display the lines containing the word root”

grep –n “root” passwd | more **//Search & display the line which contain the word “root” in the passwd file**

echo “display the count of lines that containing the word root”

grep –c “root” passwd **//Search & display the line which contain the word “root” in the passwd file**

echo “display all the lines that don’t match with the line root”

grep –v “root” passwd | more **//Search & display the line which does not contain the word “root” in the passwd file**

echo “replace “:” with “\*” in the passwd”

tr ‘:’ ‘\*’ < passwd | more **// Translate each occurrence of “: “ with “ \* “**

echo “display first column of the file passwd”

cut –d‘:’ –f1 passwd // **Display first column of the file passwd**

**OUTPUT:**

[cs2ax@kgcaslinux ~] $ sh program5

FILTER COMMANDS

Display the lines containing the word root:

1:root:x:0:0:root:/root:/bin/bash

10:operator:x:11:0:operator:/root:/sbin/nologin

Display the count of lines that containing the word root:

2

Display all the lines that donot match with the line root:

bin:x:1:1:bin:/bin:/sbin/nologin

daemon:x:2:2:daemon:/sbin:/sbin/nologin

adm:x:3:4:adm:/var/adm:/sbin/nologin

Display no of lines, words and characters in passwd file:

1404 1451 65016 passwd

Replace : with \* in the file passwd:

root\*x\*0\*0\*root\*/root\*/bin/bash

bin\*1\*1\*bin\*/bin\*/sbin/nologin

daemon\*x\*2\*2\*daemon\*/sbin\*/sbin/nologin

Display first column of the file passwd:

root

bin

daemon

adm

**PROGRAM 6:**

SHELL SCRIPT TO REMOVE FILE WHICH HAS SIZE AS 0 BYTE

**PROCEDURE:**

**STEP 1:** Start the process.

**STEP 2:** Create a file using vi editor.

**STEP 3:** Get the file name as input from the user.

**STEP 4:** Using the if..else statement check the condition.

**STEP 5:** If the condition of file size as zero means remove the file.

**STEP 6:** By using the run command to remove the file.

**STEP 7:** Stop the process.

**SHELL SCRIPTS :**

clear

echo “enter any file name : ”

read filenm **// Read the file name from the user**

if [ -e $filenm ] **//Checking existence of file filenm**

then

echo $filenm “file exist”

if [ -s $filenm ]

then

echo $filenm “file has size > 0” else

rm $filenm  **//Remove the file filenm when it’s file size is zero**

echo $filenm “file is deleted which has size = 0”

fi

else

echo “file not exist”

fi

**OUTPUT:**

[cs2ax@kgcaslinux ~] $ sh program6

Enter any file name:

One

One file exist

One file has size > 0

One file is deleted which has size = 0

Enter any file name:

Fruit

Fruit file exist

Enter any file name:

Second

Second file not exist

**PROGRAM 7:**

SHELL SCRIPT TO FIND SUM OF INDIVIDUAL DIGITS

**PROCEDURE:**

**STEP 1:** Start the process.

**STEP 2:** Create a file using vi editor.

**STEP 3:** Using echo print the statement.

**STEP 4:** Get the file name as input from the user. Declare sd=0 and sum=0.

**STEP 5:** By using the while condition print the sum of the individual digits of the given numbers.

**STEP 6:** The output will be displayed on the screen.

**STEP 7:** Stop the process.

**SHELL SCRIPTS :**

echo –n “enter number : ”

read n

sd=0 **//store single digit**

sum=0 **//store number of digit**

while [ $n –gt 0 ] **// use while loop to calculate sum of all digits**

do

sd=$(( $n % 10 )) **//get Remainder**

n=$(( $n / 10 )) **//get Next digit**

sum=$(( $sum + $sd ))/**/calculate sum of digit**

done

echo “sum of all digits is $sum”

**OUTPUT:**

[cs2ax@kgcaslinux ~] $ sh program7

Enter the number

12

Sum of all digits is 3

**PROGRAM 8:**

Shell Script to find greatest among the given number using command line argument.

**PROCEDURE:**

**STEP 1:** Start the process.

**STEP 2:** Create a file using vi editor.

**STEP 3:** Using echo print the statement.

**STEP 4:** Get the file name as input from the user.

**STEP 5:** Using the for loop check the condition and print the numbers stored in the array.

**STEP 6:** Using if print the statement as greater and smaller numbers.

**STEP 7:** Print the smallest numbers and largest numbers.

**STEP 8:** The output will be displayed on the screen.

**STEP 9:** Stop the process.

**SHELL SCRIPTS :**

clear

for((i=0;i<$1;i++))

do

echo "enter $((i+1)) number” **//Enter each integer element separated by white spaces.**

read nos[$i] **//Read the numbers**

done

echo "number entered are"

for((i=0;i<$1;i++))

do

echo ${nos[$i]}

done

small=${nos[0]}

greater=${nos[0]}

for((i=0;i<$1;i++))// **To display smaller and greatest number**

do

if [ ${nos[$i]} -lt $small ]; then

small=${nos[$i]}

elif [ ${nos[$i]} -gt $greater ]; then

greater=${nos[$i]}

fi

done

echo "Smallest number in an array is $small"

echo "Greatest number in an array is $greater"

**OUTPUT :**

[ctb11@Kgcaslinux ~]$ sh command.sh 3

Enter 1 number:

3

Enter 2 number:

98

Enter 3 number:

56

Numbers entered are:

3

98

56

Smallest number in a array 3

Greatest number in a array 98

**PROGRAM 9:**

Shell Script for palindrome checking

**PROCEDURE:**

**STEP 1**: Create a file using vi editor.

**STEP 2**:Read the string or number n from user .

**STEP 3:** Initialize variables sd to 0 and rev to null.

**STEP 4 :** Find n%10 and store it in sd.

**STEP 5** : Find n/10 and store it in n.

**STEP 6** : Use rev command to reverse the value in sd.

**STEP 7:** Repeat the steps from 3 to 5 till n>0 using while loop.

**STEP 8:** Compare the reversed string with original string.Both are equal print that it is a Palindrome otherwise it is not a palindrome.

**STEP 9 :**Stop the Process.

**SHELL SCRIPT:**

clear

echo -n "Enter Number: "

read n **//Reads the entered number**

sd=0 **//storing the remainder**

rev=" " **// store number in reverse order**

on=$n **//store orginal number in another variable**

while [ $n -gt 0 ]

do

sd=$(($n%10 )) /**/Get Remainder**

n=$(($n/10 )) **//Get Next digit**

rev=$(echo ${rev}${sd} ) **//Store previous number and current digit in reverse**

done

if [ $on -eq $rev ]; then **//Compare two numbers**

echo "Number is palindrome"

else

echo "Number is not palindrome"

fi

**OUTPUT :**

[ctb11@Kgcaslinux ~]$sh palindrome.sh

Enter number

121

Number is palindrome

**PROGRAM 10:**

Shell Script to print multiplication table

**PROCEDURE:**

**STEP** 1: Create a file using vi editor.

**STEP** 2: Read the table number n from the user using read command.

**STEP** 3: Read the range from the user using read command.

**STEP** 4: Repeat the statement echo " $i X $n = `expr $n \\* $i`" until i less than range using for loop.

**STEP** 5: Stop the process.

**SHELL SCRIPT :**

echo "enter the table number"

read n

echo "enter the range"

read range

echo "multiplication table for $n upto the range $range"

for((i=1;i<=range;i++))

{

echo " $i X $n = `expr $n \\* $i`" **//The loop will be executed until i less than range**

}

**OUTPUT :**

[csb1@Kgcaslinux ~]$ sh multiplication.sh

Enter the table number

5

enter the range

10

multiplication table for 5 upto the range 10

1 \* 5 = 5

2 \* 5 = 10

3 \* 5 = 15

4 \* 5 = 20

5 \* 5 = 25

6 \* 5 = 30

7 \* 5 = 35

8 \* 5 = 40

9 \* 5 = 45

10 \* 5 = 50