Lab # 04

SHELL PROGRAMMING AND TEXT EDITOR

1. WHY WE NEED SHELL SCRIPTING?

- Combine lengthy and repetitive sequences of commands into a single, simple command.
- Generalize a sequence of operations on one set of data, into a procedure that can be applied to any similar set of data.
- Create new commands using combinations of utilities in ways the original authors never thought of.
- And much more.

2. BASIC ARITHMETIC OPERATION USIG SHELL PROGRAMMIG

AIM: To write a shell program to solve arithmetic operation.

ALGORITHM :

Step 1 : Include the necessary header files.
Step 2 : get the input
Step 3 : perform the arithmetic calculation.
Step 4 : print the result.
Step 5 :stop the exeution.

PROGRAM CODING:

Example 1:

```
#!/bin/bash
echo "enter the a vale"
read a
echo "enter b value"
read b
c=`expr $a + $b`
echo "sum:"$c
c=`expr $a - $b`
echo "sub:"$c
c=`expr $a \* $b`
echo "mul:"$c
c=`expr $a / $b`
echo "div:"$c
```

3. FOR LOOP IN SHELL PROGRAMMING

i. Number checking using shell programing

AIM: To write a shell program to check whether the number is odd or even.

PROGRAM CODING:

Example 2:

```
#!/bin/bash
num="1 2 3 4 5 6 7 8"
for n in $num
do
q=`expr $n % 2`
if [ $q -eq 0 ]
then
echo "even no"
continue
fi
echo "odd no"
done
```

ii. Multiplication table using shell programing

PROGRAM CODING:

Example 3:

```
#!/bin/bash
echo " which table you want"
read n
for i in 1 2 3 4 5 6 7 8 9 10
do
echo $i "*" $n "=" `expr $i \* $n`
done
```

iii. Listing output of the command using for loop

PROGRAM CODING:

Example 4:

```
#!/bin/bash
echo " which table you want"
read n
for i in $(ls)
do
gedit i
done
```

4. WHILE LOOP IN SHELL PROGRAMMING

PROGRAM CODING:

Example 5:

```
#!/bin/bash
a=1
while [ $a -lt 11 ]     //-ge -gt -lt -le -eq -ne
  //[ $a -ne 11  -a $a -ne 12    ]
  //[ $a -ne 11 -o  $a -ne 12     ]
do
echo "$a"
a=`expr $a + 1`
done
```

5. USING IF STATEMENT IN SHELL PROGRAMING

PROGRAM CODING:

Example 7:

```
#!/bin/bash
for var1 in 1 2 3
do
for var2 in 0 5
do
if [ $var1 -eq 2 -a $var2 -eq 0 ]
then
continue
else
echo "$var1 $var2"
fi
done
```

6. USING ElSE IF STATEMENT IN SHELL PROGRAMING

PROGRAM CODING:

Example 8:

```
#!/bin/bash
for var1 in 1 2 3
do
for var2 in 0 5
do
if [ $var1 -eq 2 -a $var2 -eq 0 ]
then
continue
else if [ $var1 -eq 4 -a $var2 -eq 1 ]
then
echo "$var1"
else
echo "$var1 $var2"
fi
fi
done
done
```

7. SIMPLE FUNCTION IN SHELL PROGRAMING

AIM: to write a shell program to add a two number using function.

PROGRAM CODING:

Example 9:

```
#!/bin/bash
add()
{
c=`expr $1 + $2`
echo "addition = $c"
}
add 5 10
```

8. SWITCH STATEMENT IN SHELL PROGRAMING

Example 6:

```
#!/bin/bash
ch='y'
while [ $ch = 'y ' ]
do
echo "enter your choice"
echo "1 no of user loged on"
echo "2 print calender"
echo "3 print date"
read d
case $d in
1) who;;
2) cal 20;;
3) date;;
*) break;;
esac
echo "do you wish to continue (y/n)"
read ch
done
```

9. TEXT EDITOR

A text editor is a software where you can enter text in its native format and save it to _le. A word processor is a software where you can take text and process its appearance, format, spell-check, paragraph settings, etc. Linux has a number of text editors (both graphical and command-line based). The one which you will use most commonly is the Nano text editor. Nano is an advanced text editor provided by GNU. Simply typing Nano on the shell will give you the editor

```
nano
or
nano <file name>
```

Near the end of your screen you will see a list of shortcuts. The ones which you should get yourself familiar with are as such:
CTRL+X for exit CTRL+O for saving
CTRL+W for searching CTRL+K for cutting
CTRL+U for pasting CTRL+C for displaying cursor position
Other commands are listed at the bottom of the text-editor window.

Exercise 1:

Write a shell script to create a file with extension .c. Copy contents(c code) from another file to this file. Now use if statement to ask user if user enter 1 just compile it. If user enters 2 compile it and run it. If user enters 3 just print the contents of the original file. Otherwise print the contents of the current directory. Perform the same task using switch inside a function.