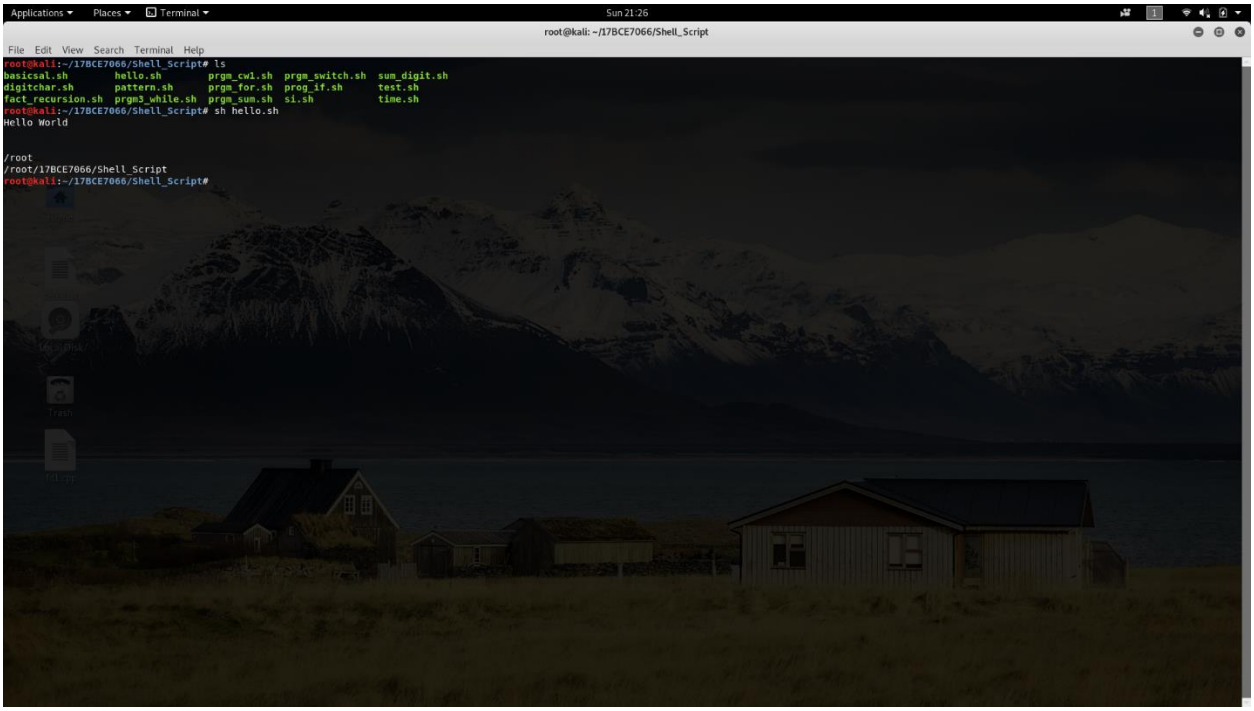
Shell-Scripting

Aditya Jain (17BCE7066)

Program 1: hello.sh

```
#! /bin/bash
#This is my first shell scripting program
echo "Hello World"
echo $BASH
echo $BASH_VERSION
echo $HOME
echo $PWD
```

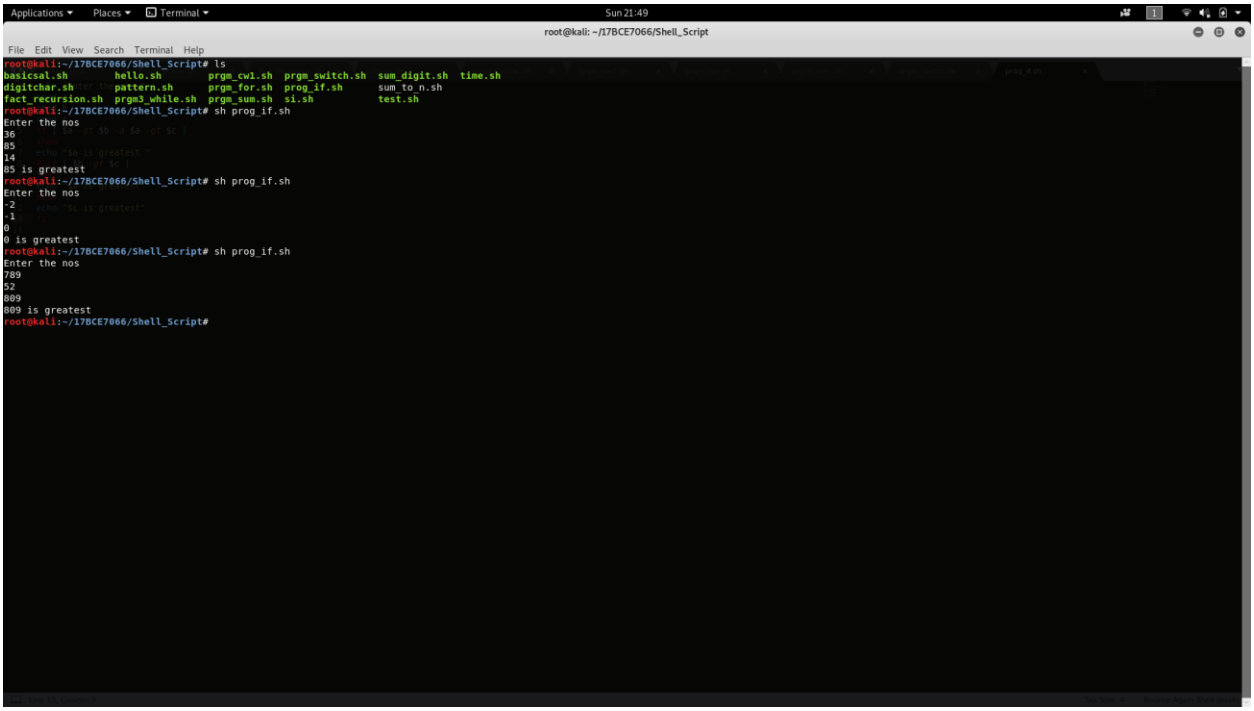
Output:



Program 2: To find the greatest of three numbers

```
echo "Enter the nos "
read a
read b
read c
if [ $a -gt $b -a $a -gt $c ]
then
echo "$a is greatest "
elif [ $b -gt $c ]
then
echo "$b is greatest "
else
echo "$c is greatest"
fi
```

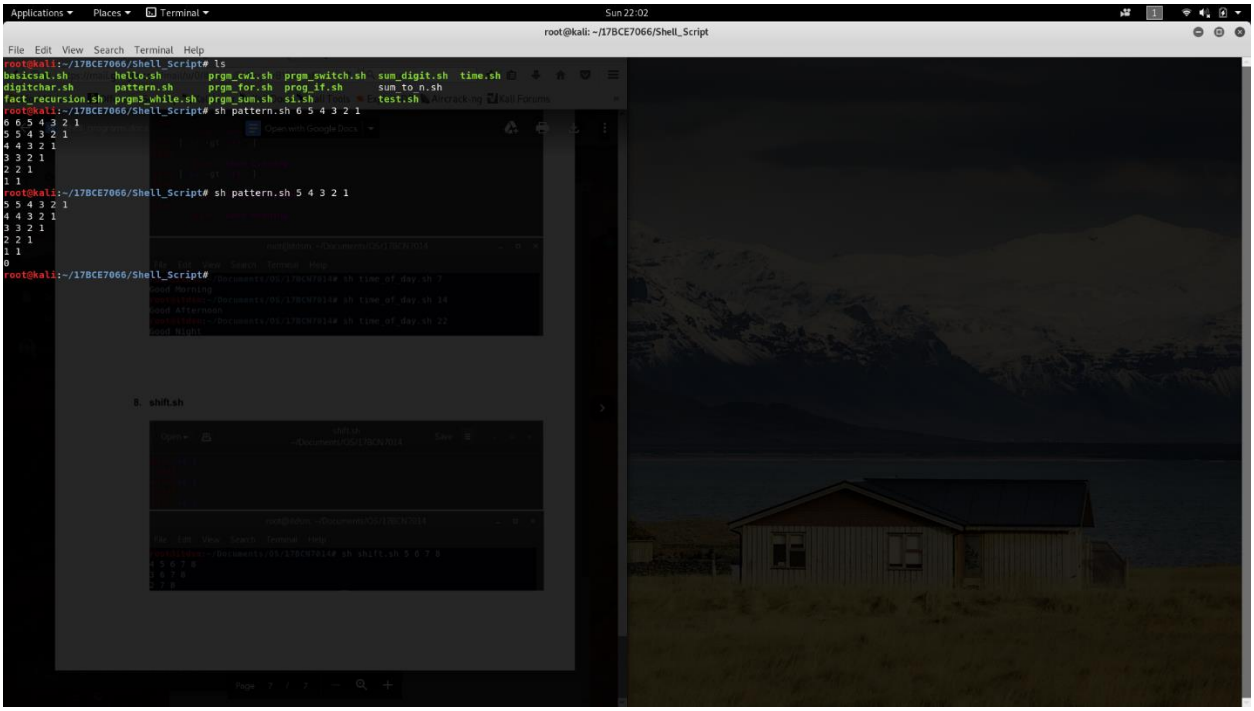
Output:



Program 3: To print a pattern using shift

```
echo $# $*
shift
echo $# $*
shift
echo $# $*
shift
echo $# $*
shift
echo $# $*
shift
echo $# $*
```

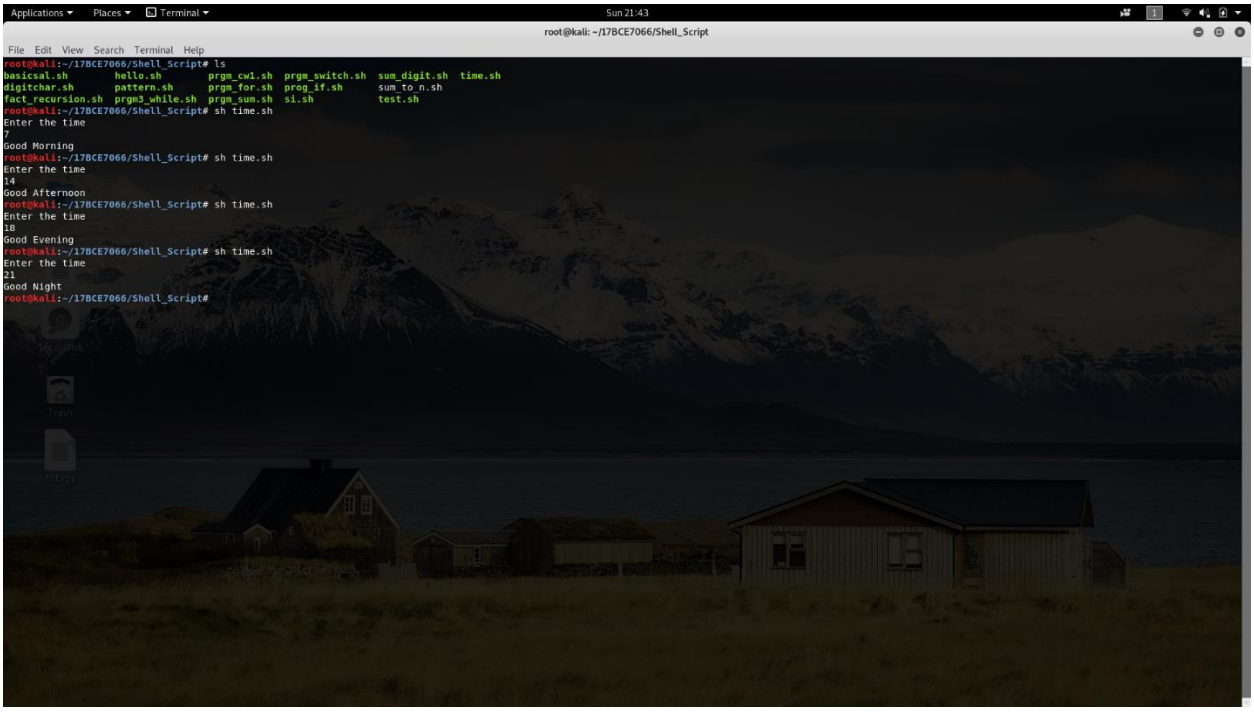
Output:



Program 4: To demonstrate else-if construct

```
echo "Enter the time"
read time
if [ $time -lt 10 ]
then
echo "Good Morning"
elif [ $time -lt 15 ]
then
echo "Good Afternoon"
elif [ $time -lt 19 ]
then
echo "Good Evening"
else
echo "Good Night"
fi
```

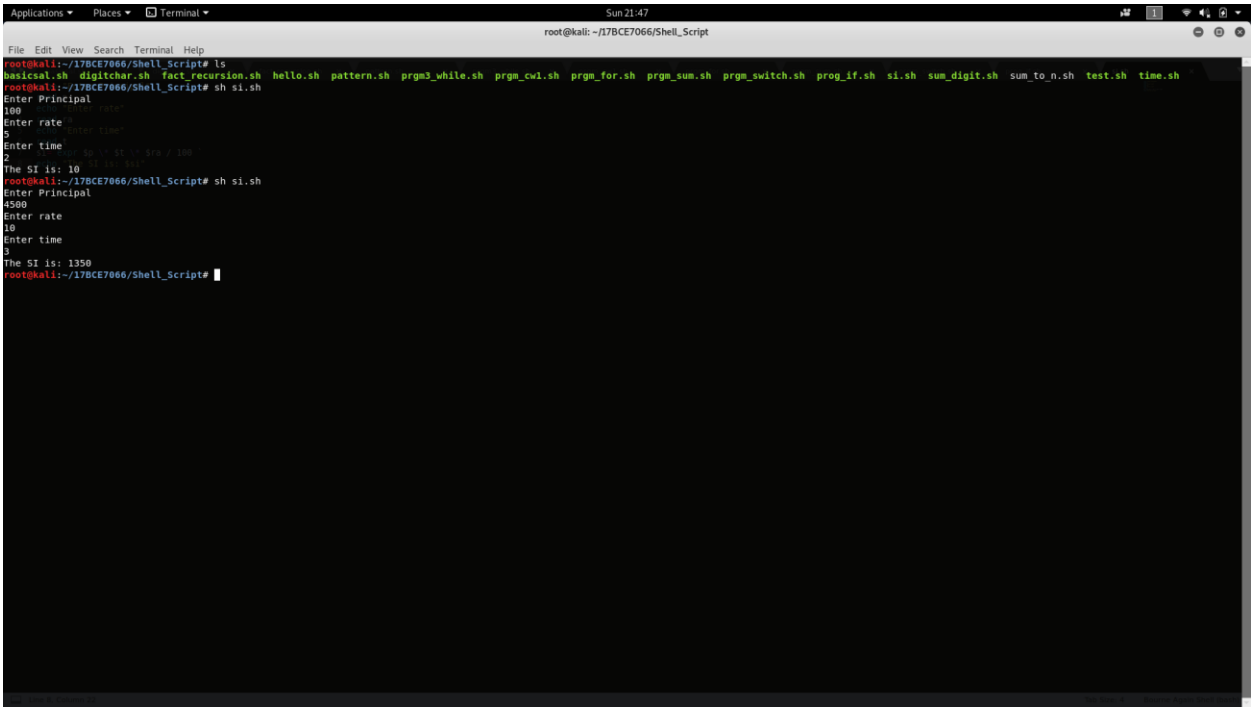
Output:



Program 5: To calculate simple interest

```
echo "Enter Principal"
read p
echo "Enter rate"
read ra
echo "Enter time"
read t
si=`expr $p \* $t \* $ra / 100 `
echo "The SI is: $si"
```

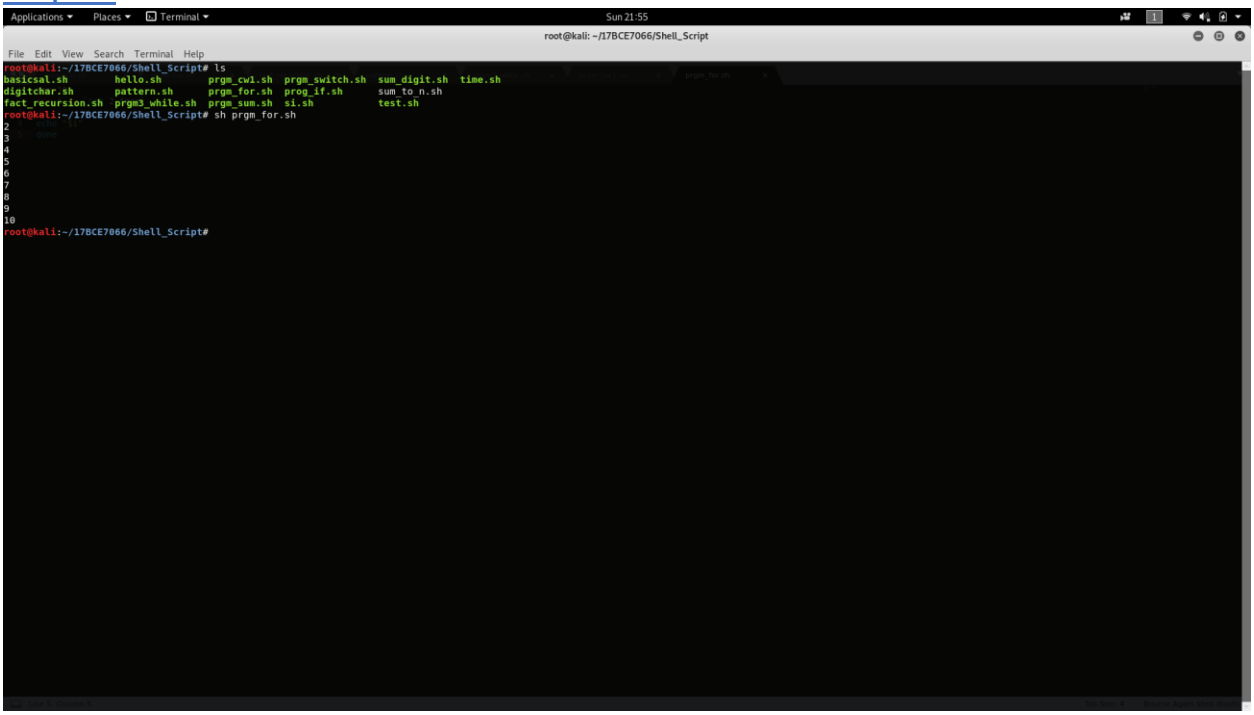
Output:



Program 6: To demonstrate for for-loop

```
max=10
for i in `seq 2 $max`
do
echo "$i"
done
```

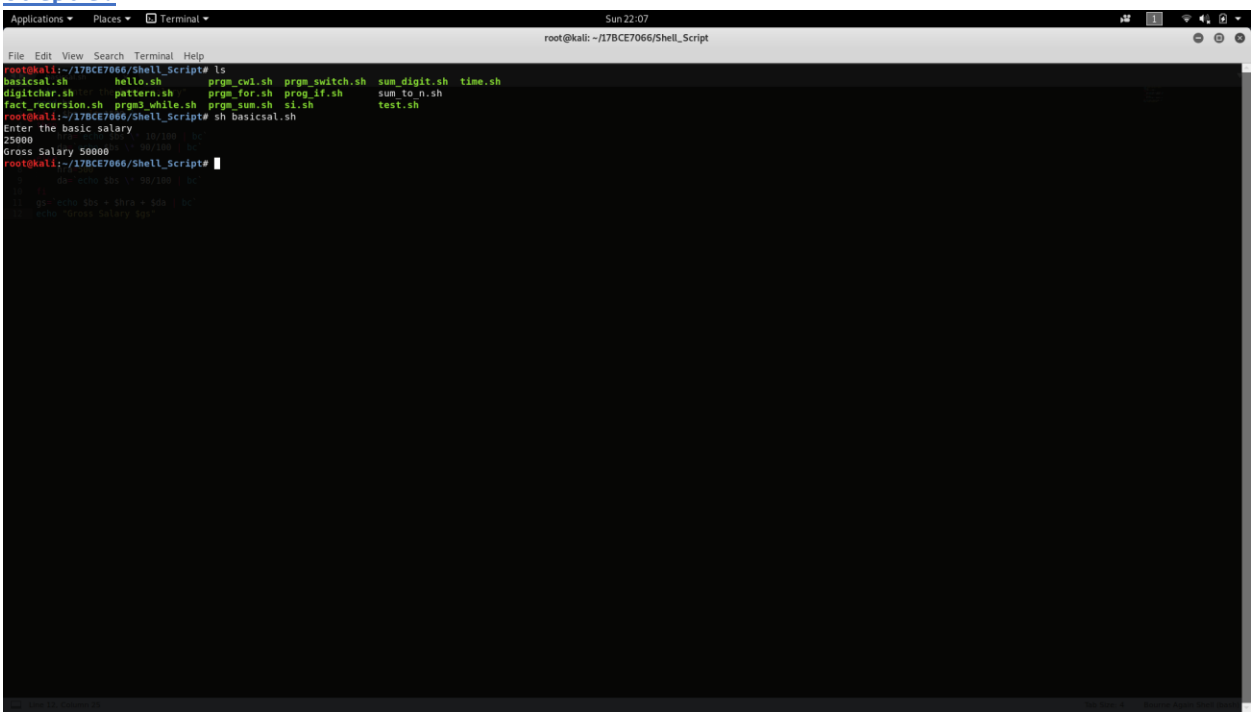
Output:



Program 7: To calculate gross salary

```
echo "Enter the basic salary"
read bs
if [ $bs -lt 1500 ]
then
        hra=`echo $bs \* 10/100 | bc`
        da=`echo $bs \* 90/100 | bc`
else
        hra=500
        da=`echo $bs \* 98/100 | bc`
fi
gs=`echo $bs + $hra + $da | bc`
echo "Gross Salary $gs"
```

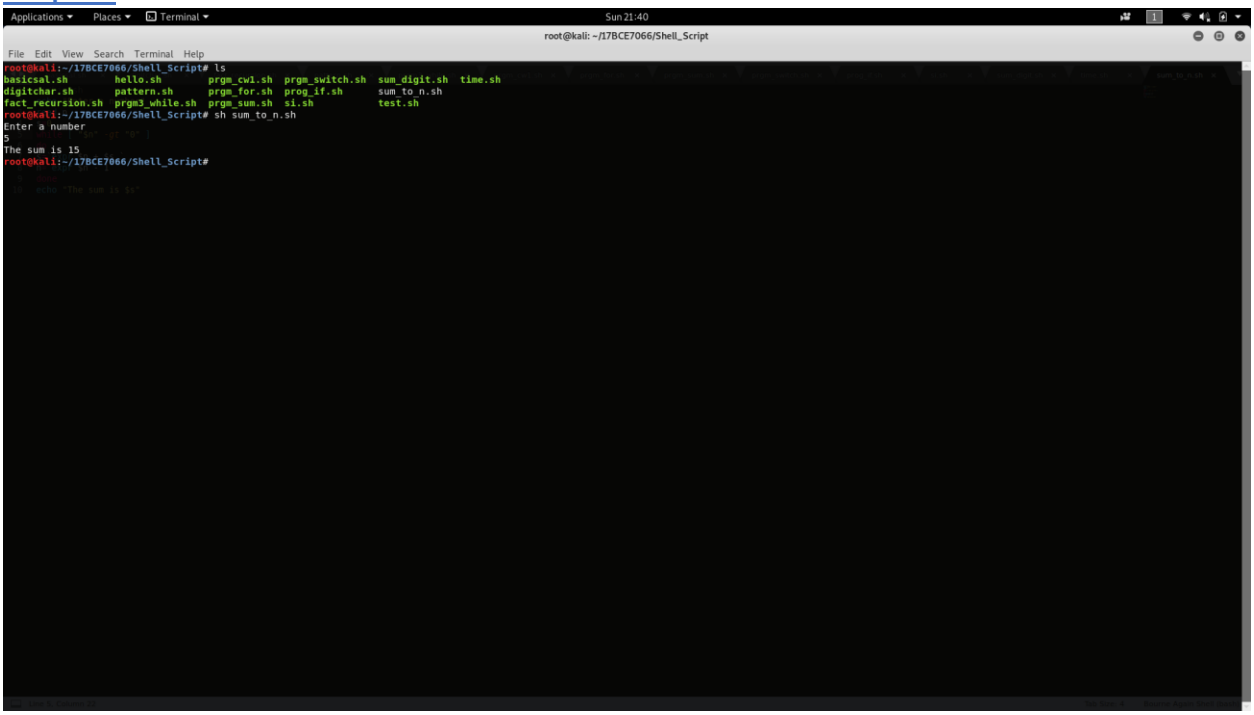
Output:



Program 8: To calculate the sum of natural numbers from 1 to n

```
#!/bin/sh
echo "Enter a number"
read n
s=0
while [ "$n" -gt "0" ]
do
s=`expr $n + $s `
n=`expr $n - 1 `
done
echo "The sum is $s"
```

Output:

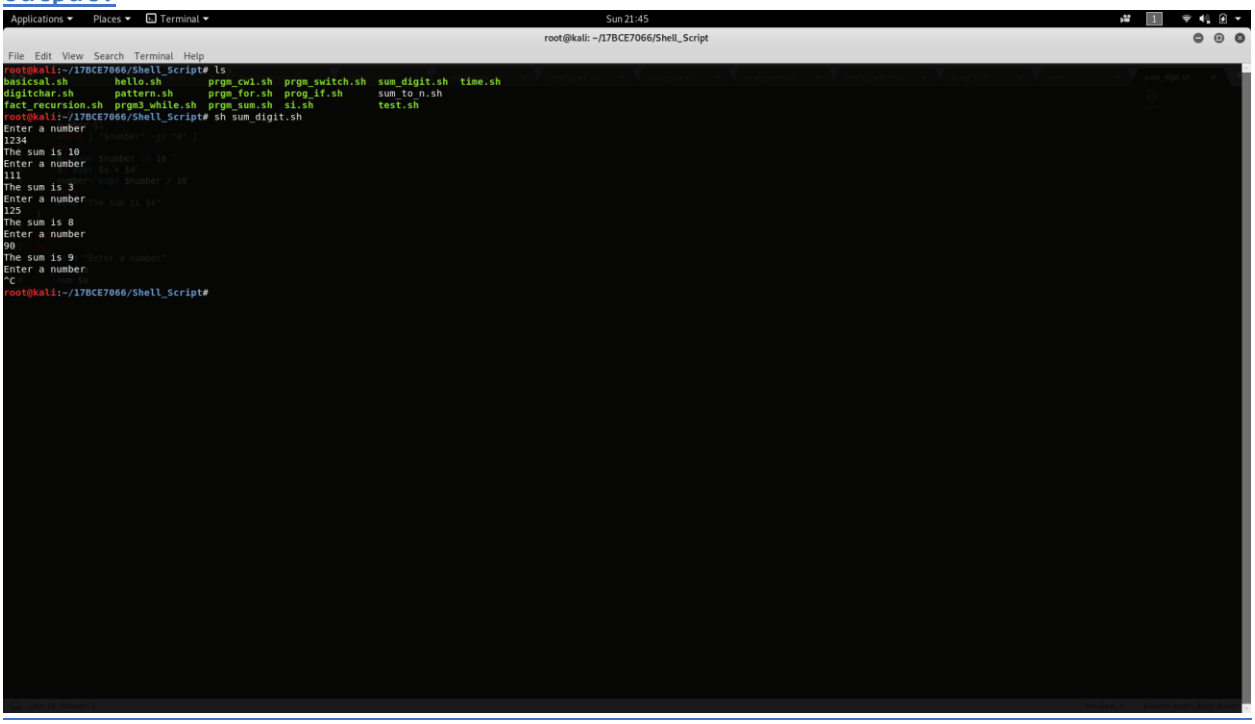


Program 9: To find the sum of digits of the number

```
num()
{
        s=0
        number=$1
        while [ "$number" -gt "0" ]
        do
        d=`expr $number \% 10 `
        s=`expr $s + $d`
        number=`expr $number / 10`
        done
        echo "The sum is $s"
}

while :
do
        echo "Enter a number"
        read a
        num $a
done
```

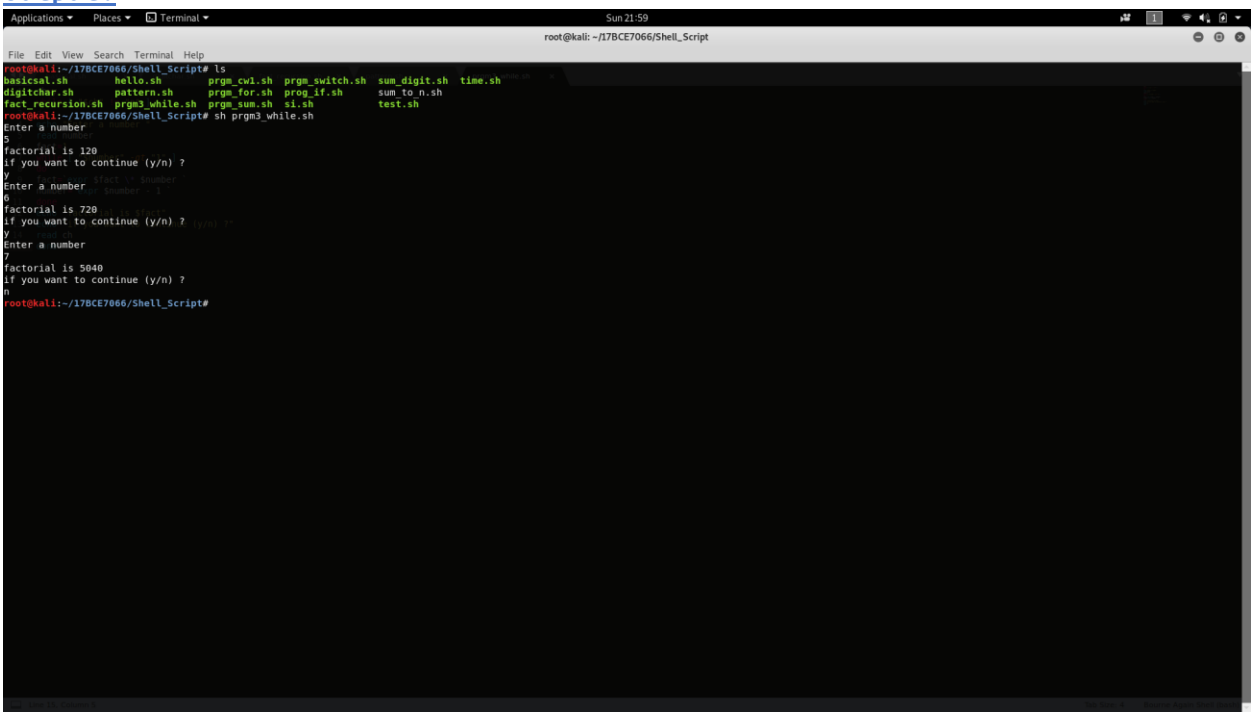
Output:



Program 10: To demonstrate while-loop

```
ch=y
while [ "$ch" != "n" ]
do
echo "Enter a number "
read number
fact=1
while [ "$number" -gt "1" ]
do
fact=`expr $fact \* $number `
number=`expr $number - 1 `
done
echo "factorial is $fact"
echo "if you want to continue (y/n) ?"
read ch
done
```

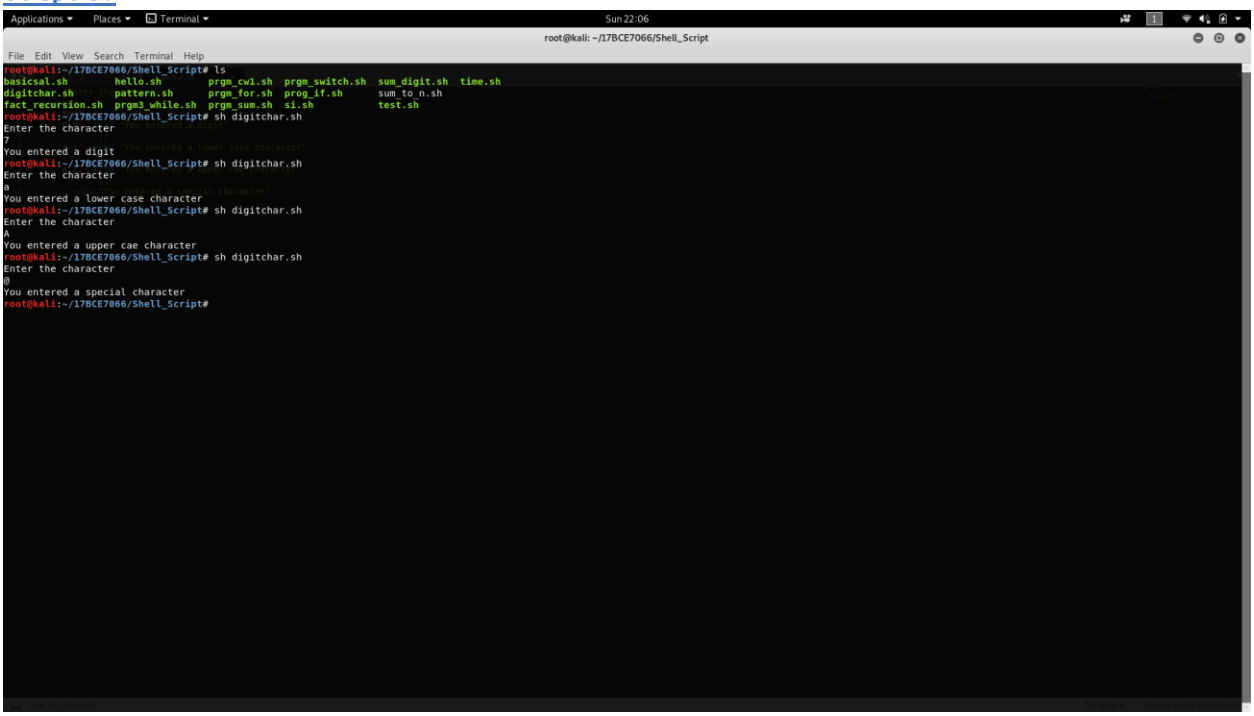
Output:



Program 11: To demonstrate switch-case

```
echo "Enter the character"
read num
case $num in
        [0-9]) echo "You entered a digit"
                        ;;
        [a-z]) echo "You entered a lower case character"
                        ;;
        [A-Z]) echo "You entered a upper cae character"
                        ;;
        *) echo "You entered a special character"
                        ;;
esac
```

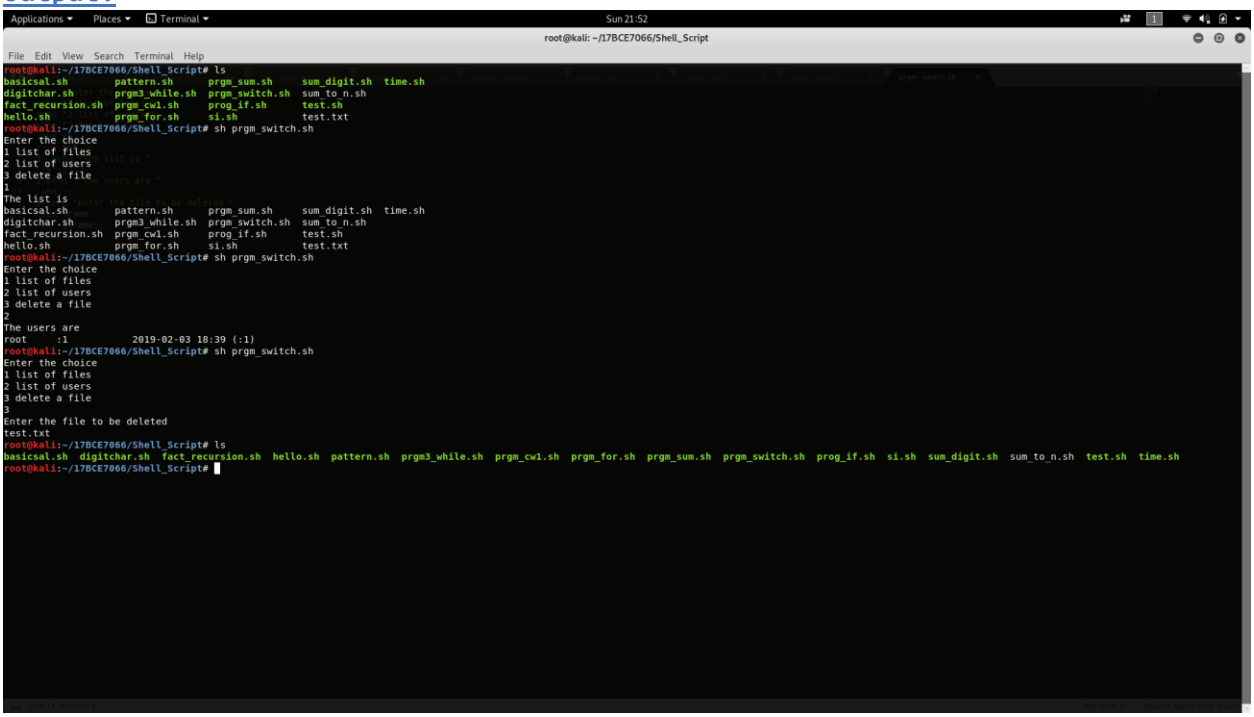
Output:



Program 12: To use switch case to make a program which list files, users and deletes the file asked by the user

```
echo "Enter the choice "
echo "1 list of files "
echo "2 list of users "
echo "3 delete a file "
read ch
case $ch in
1) echo "The list is "
'ls';;
2)echo  "The users are "
'who';;
3)echo "Enter the file to be deleted "
read frame
'rm' $frame;;
esac
```

Output:



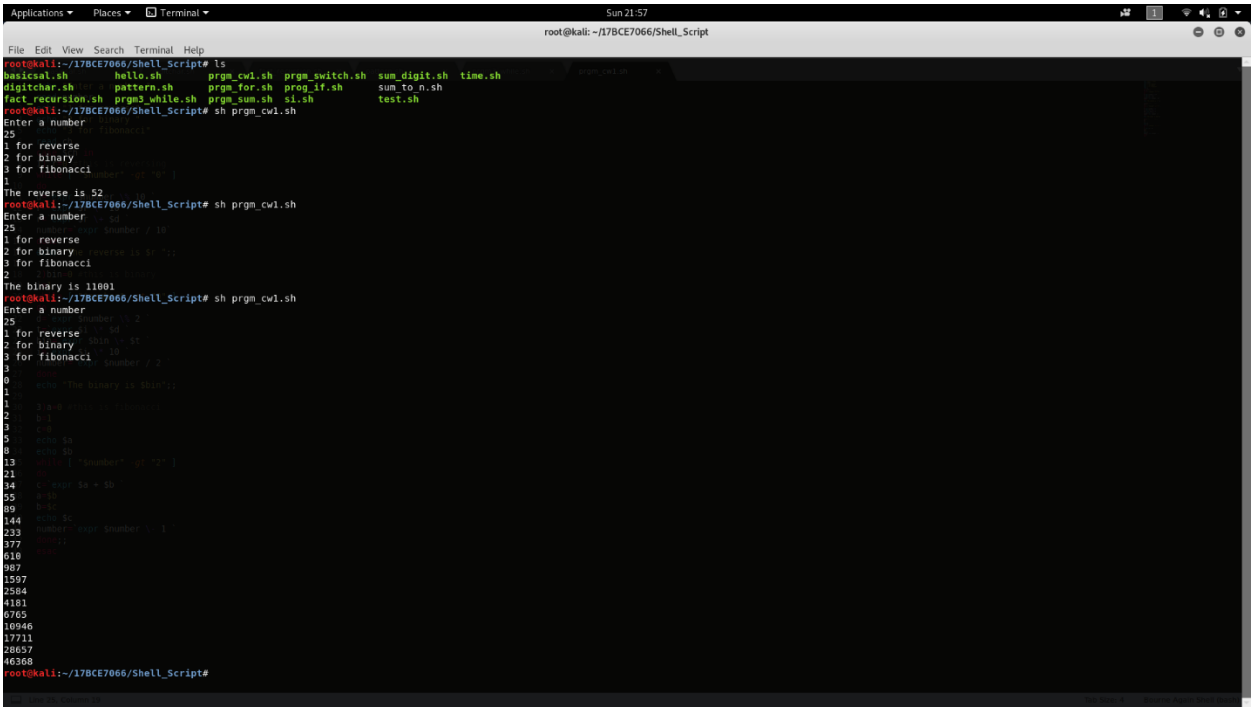
Program 13: To make a program using switch case to reverse a number or convert it to binary or print Fibonacci series to n terms

```
echo "Enter a number "
read number
echo "1 for reverse"
echo "2 for binary "
echo "3 for fibonacci"
read ch
case $ch in
1) r=0 #this is reversing
while [ "$number" -gt "0" ]
do
d=`expr $number \% 10 `
r=`expr $r \* 10 `
r=`expr $r \+ $d `
number=`expr $number / 10`
done
echo "The reverse is $r ";;

2)bin=0 #this is binary
i=1
while [ "$number" -gt "0" ]
do
d=`expr $number \% 2 `
t=`expr $i \* $d `
bin=`expr $bin \+ $t `
i=`expr $i \* 10 `
number=`expr $number / 2 `
done
echo "The binary is $bin";;

3)a=0 #this is fibonacci
b=1
c=0
echo $a
echo $b
while [ "$number" -gt "2" ]
do
c=`expr $a + $b `
a=$b
b=$c
echo $c
number=`expr $number \- 1 `
done;;
esac
```

Output:

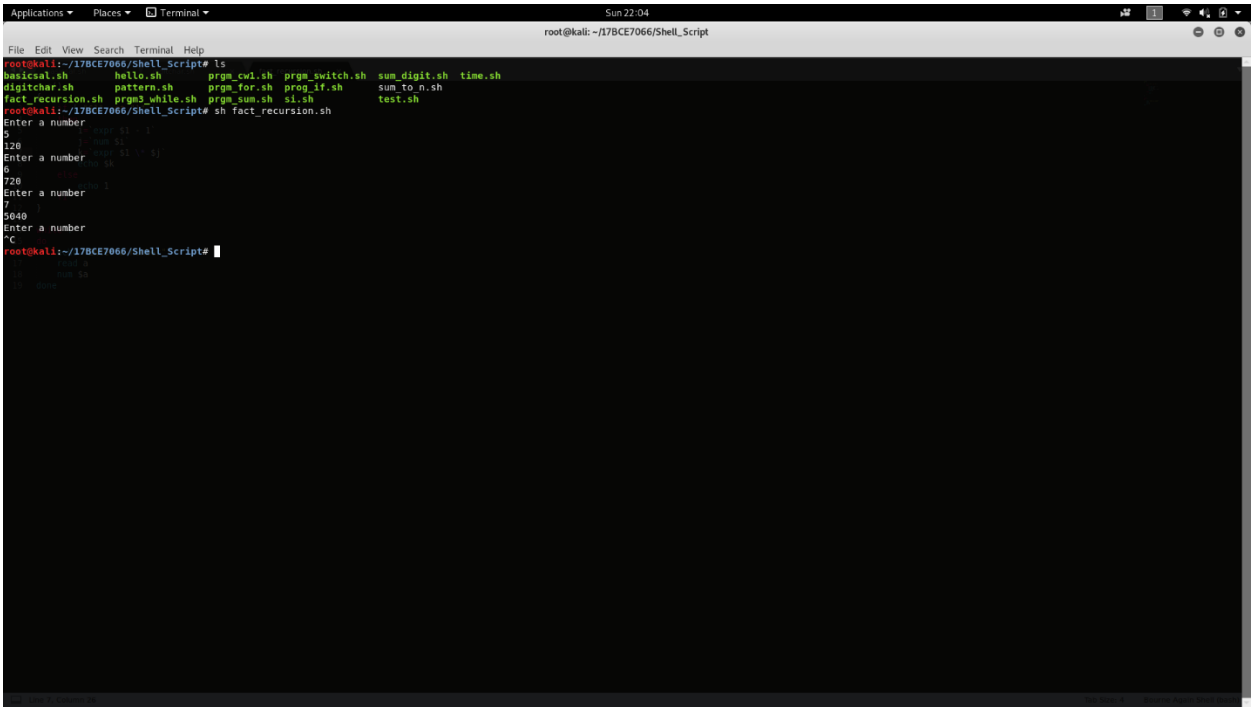


Program 14: To demonstrate recursion and find factorial using recursion

```
num()
{
        if [ "$1" -gt "1" ]
        then
                i=`expr $1 - 1`
                j=`num $i`
                k=`expr $1 \* $j`
                echo $k
        else
                echo 1
        fi
}

while :
do
        echo "Enter a number"
        read a
        num $a
done
```

Output:


Applications
Places
Terminal
Sun 22:04
root@kali: ~/17BCE7066/Shell_Script
File Edit View Search Terminal Help
root@kali:~/17BCE7066/Shell_Script# ls
basicsal.sh hello.sh prgm_cw1.sh prgm_switch.sh sum_digit.sh time.sh
digitchar.sh pattern.sh prgm_for.sh prog_if.sh sum_to_n.sh
fact_recursion.sh prgm3_while.sh prgm_sum.sh si.sh test.sh
root@kali:~/17BCE7066/Shell_Script# sh fact_recursion.sh
Enter a number
5
120
Enter a number
0
720
Enter a number
7
5040
Enter a number
^C
root@kali:~/17BCE7066/Shell_Script#