Title: Shell scripting

Objective

- To learn about shell scripting
- To learn about different shell scripting commands
- To learn to program on the shell

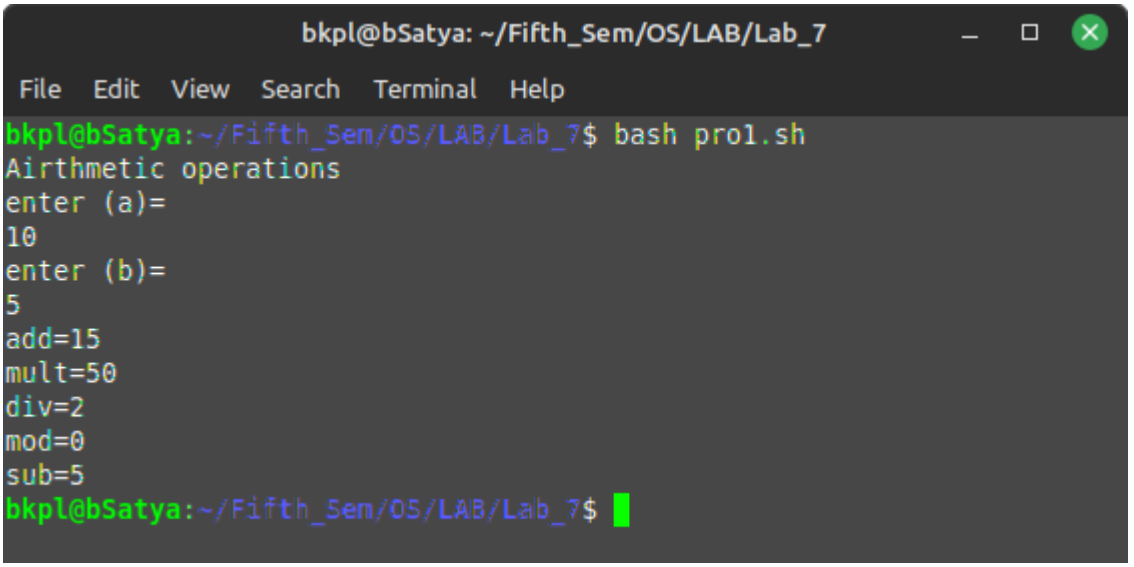
Background

Shell
The shell provides users with an interface to the UNIX system. It gathers input from users and executes programs based on that input. After successful executions of command, it displays the output. A shell is an environment in which we can run our commands, programs, and shell scripts. There are different types of shells, just as there are different operating systems. Each type of shell have its own set of recognized commands and functions.

Shell Scripting
Shell scripting is writing a series of commands for the shell to execute. It can combine lengthy and repetitive sequences of commands into a single and simple script, which can be stored and executed anytime. This reduces the effort required by the end user. The shell script is interpreted and not compiled. There are many types of shells. In this session, we are learning BASH Shell scripting.

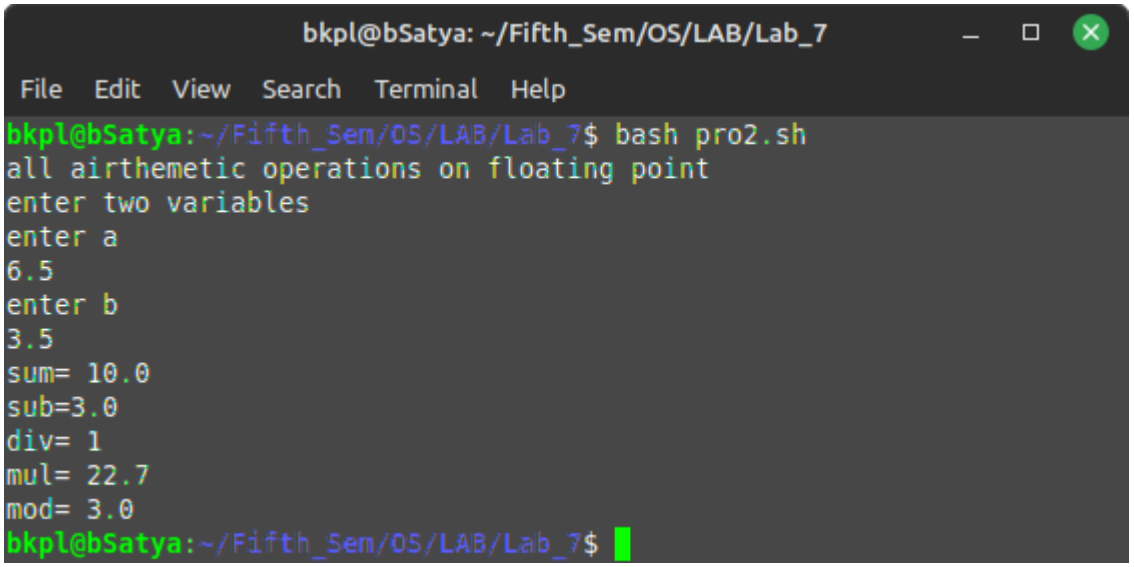
Lab Activities

1.
```
echo "Arithmetic operations"
echo "enter (a)="
read a
echo "enter (b)="
read b
echo "add=`expr $a + $b`"
echo "mult=`expr $a \* $b`"
echo "div=`expr $a / $b`"
echo "mod=`expr $a % $b`"
echo "sub=`expr $a
```



2.

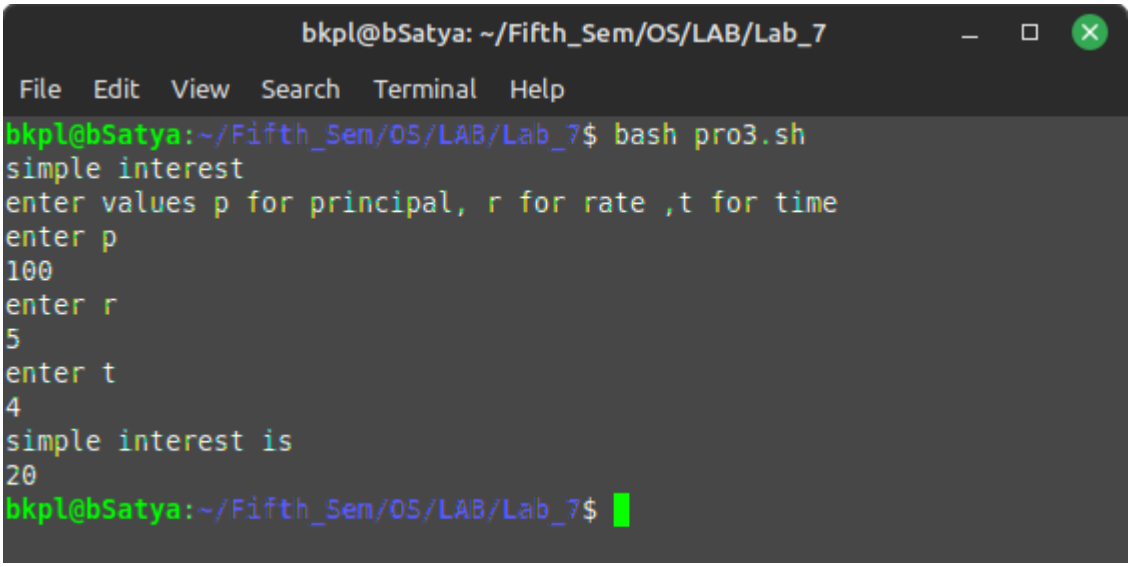
```
echo "all arithmetic operations on floating point"
echo "enter two variables"
echo "enter a"
read a
echo "enter b"
read b
sum=` echo $a + $b | bc `
sub=` echo $a - $b | bc `
div=` echo $a / $b | bc `
mul=` echo $a \* $b | bc `
mod=` echo $a % $b |bc `
```



3.

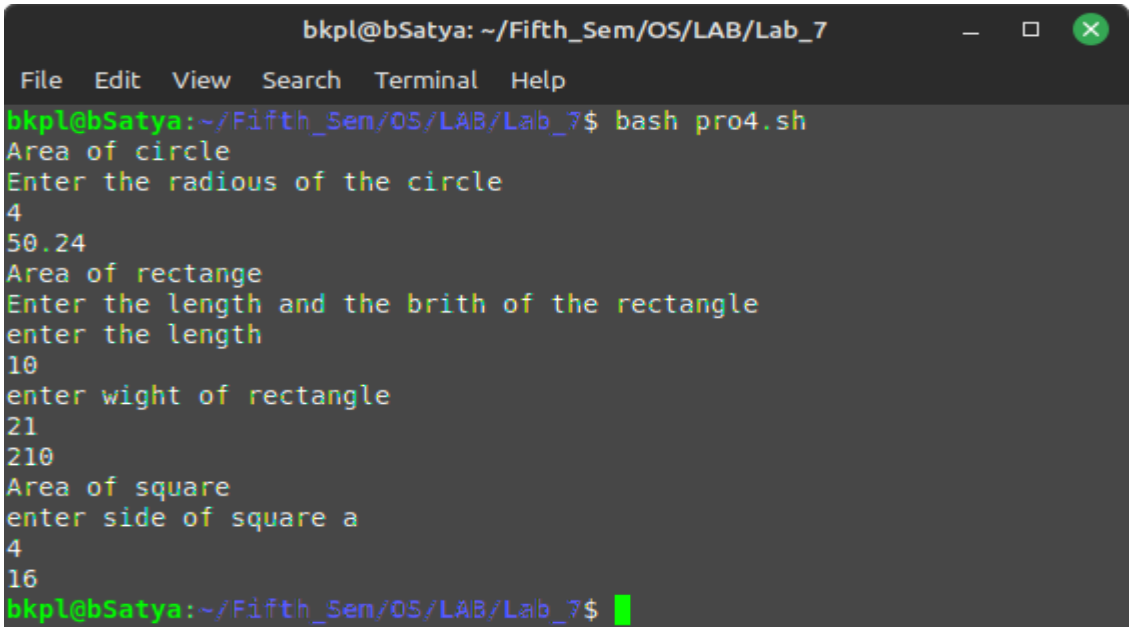
```
echo "simple interest"
echo "enter values p for principal , r for rate ,t for time"
echo "enter p"
read p
echo "enter r"
read r
echo "enter t"
read t
```

```
echo "simple interest is"
s=` expr $p \* $r \* $t `
si=`echo $s / 100 | bc`
echo "$si"
```



4.

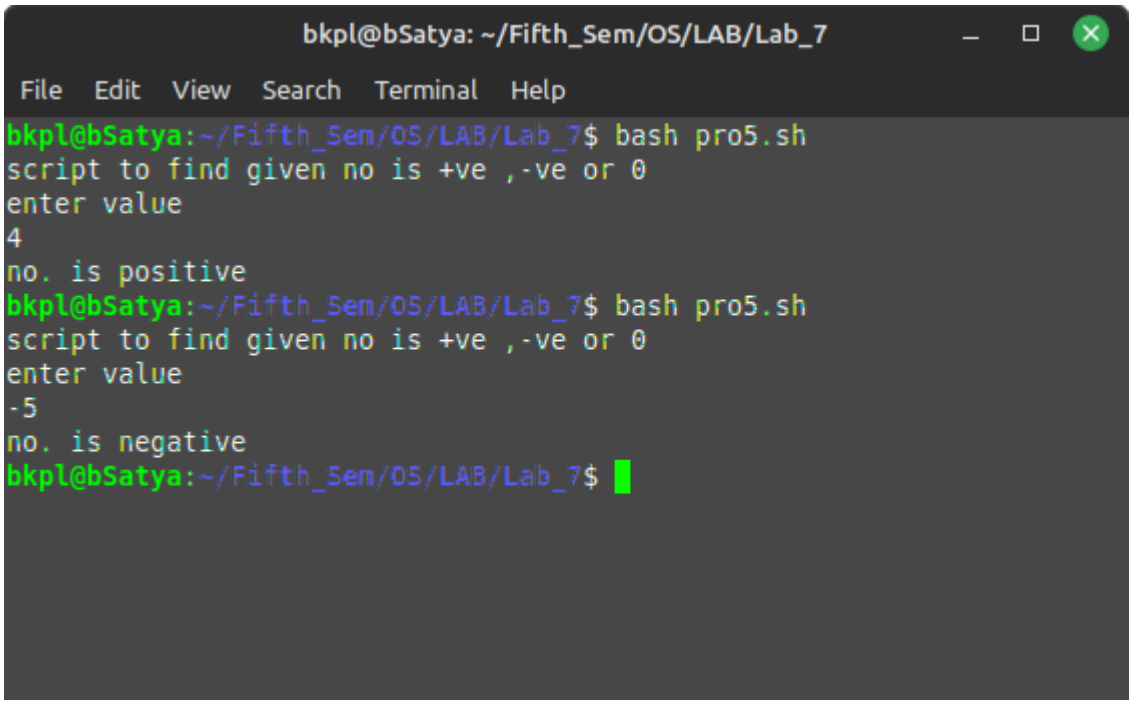
```
echo "Area of circle"
echo "Enter the radius of the circle"
read r
a1=` echo $r \* $r \* 3.14 | bc `
echo "$a1"
echo Area of rectangle
echo Enter the length and the breath of the rectangle
echo enter the length
read l
echo enter weight of rectangle
read b
a2=` echo $l \* $b | bc `
echo $a2
echo Area of square
echo enter side of square a
read a
a3=` echo $a \* $a | bc `
echo $a3
```



5.

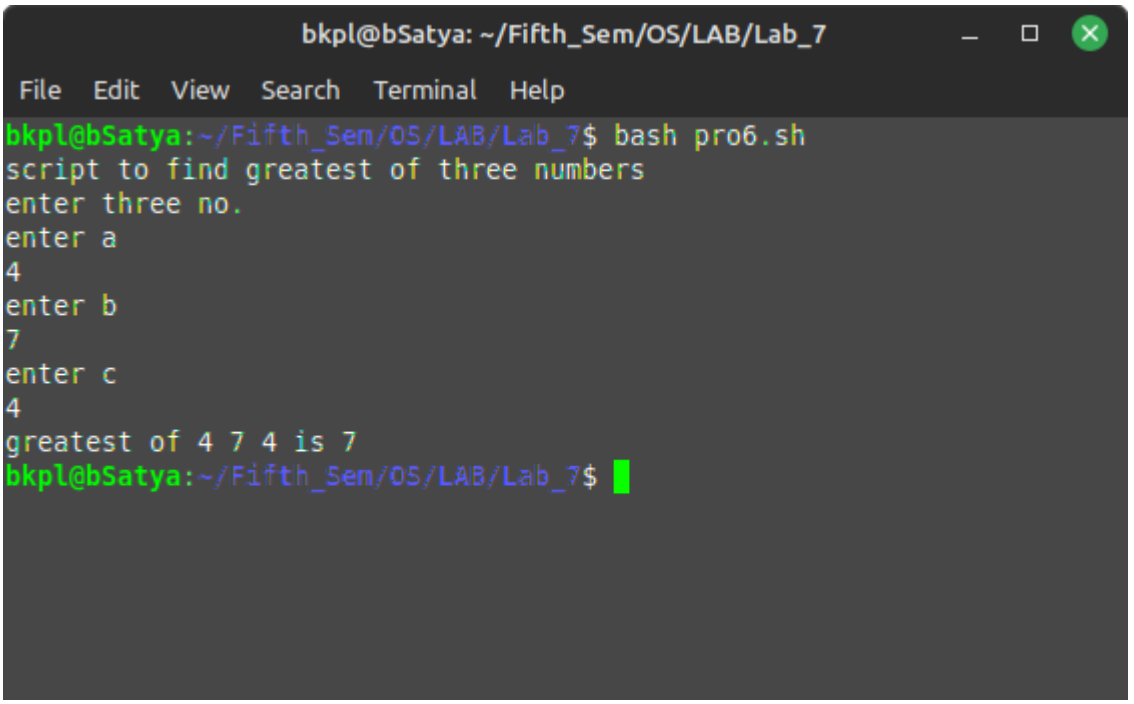
```
echo "script to find given no is +ve ,-ve or 0"
echo enter value
read a
if [ $a -gt 0 ]
then
  echo no. is positive

else
   if [ $a -lt 0 ]
   then
     echo no. is negative
   else
      echo no. is zero
    fi
fi
```



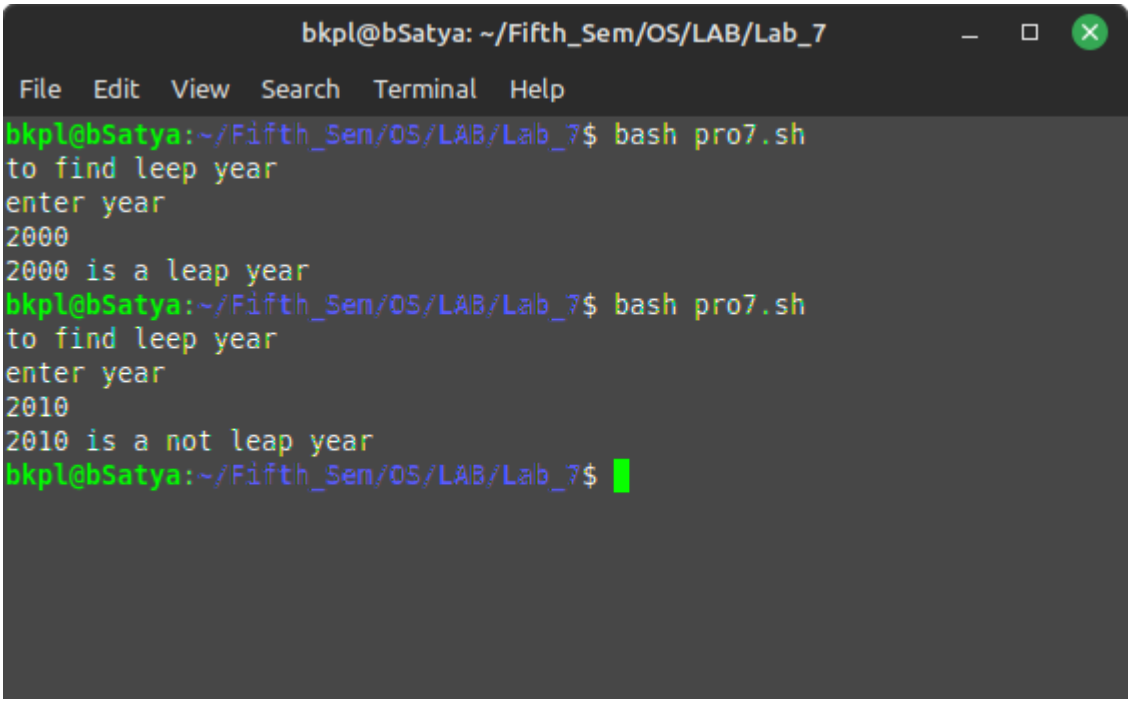
6.

```
echo "script to find greatest of three numbers"
echo enter three no.
echo enter a
read a
echo enter b
read b
echo enter c
read c
r=$a
if [ $b -gt $r ]
then
r=$b
fi
if [ $c -gt $r]
then
r=$c
fi
echo greatest of $a $b $c is $r
```



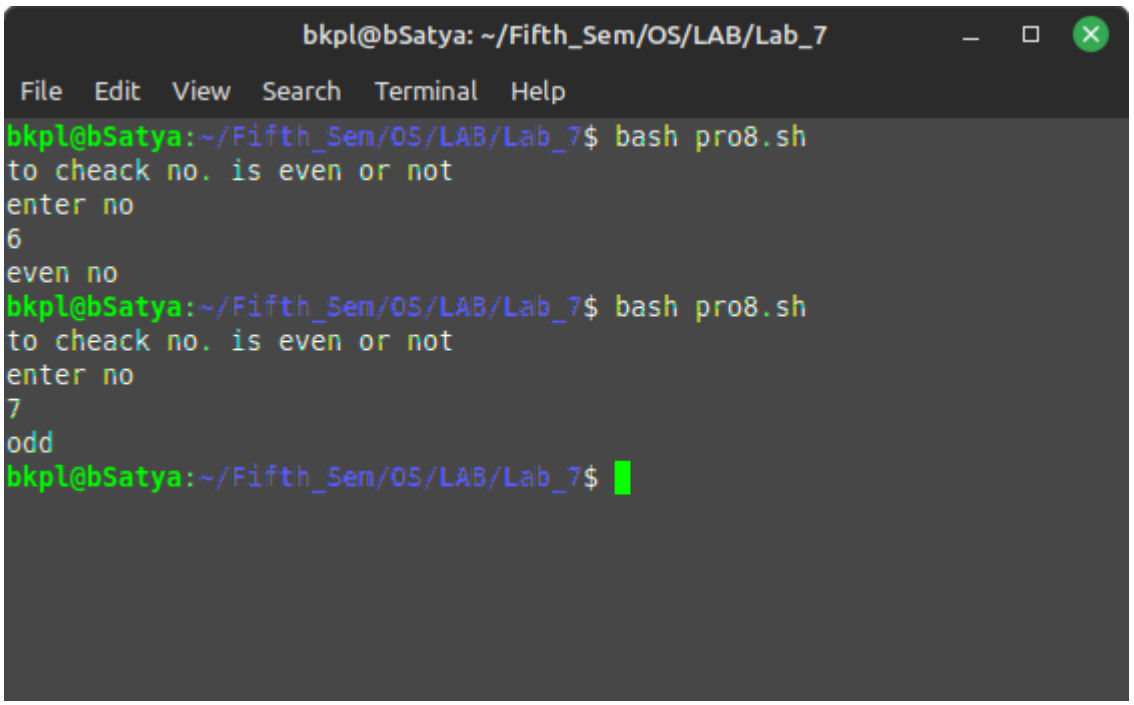
7.

```
echo "to find leap year"
echo enter year
read year
if [ ` expr $year % 100 ` -eq 0 -a ` expr $year % 400 ` -eq 0 ]
then
        echo "$year is a leap year"
elif [ ` expr $year % 100 ` -ne 0 -a ` expr $year % 4 ` -eq 0 ]
then
        echo "$year is a leap year"
else
        echo "$year is a not leap year"
fi
```



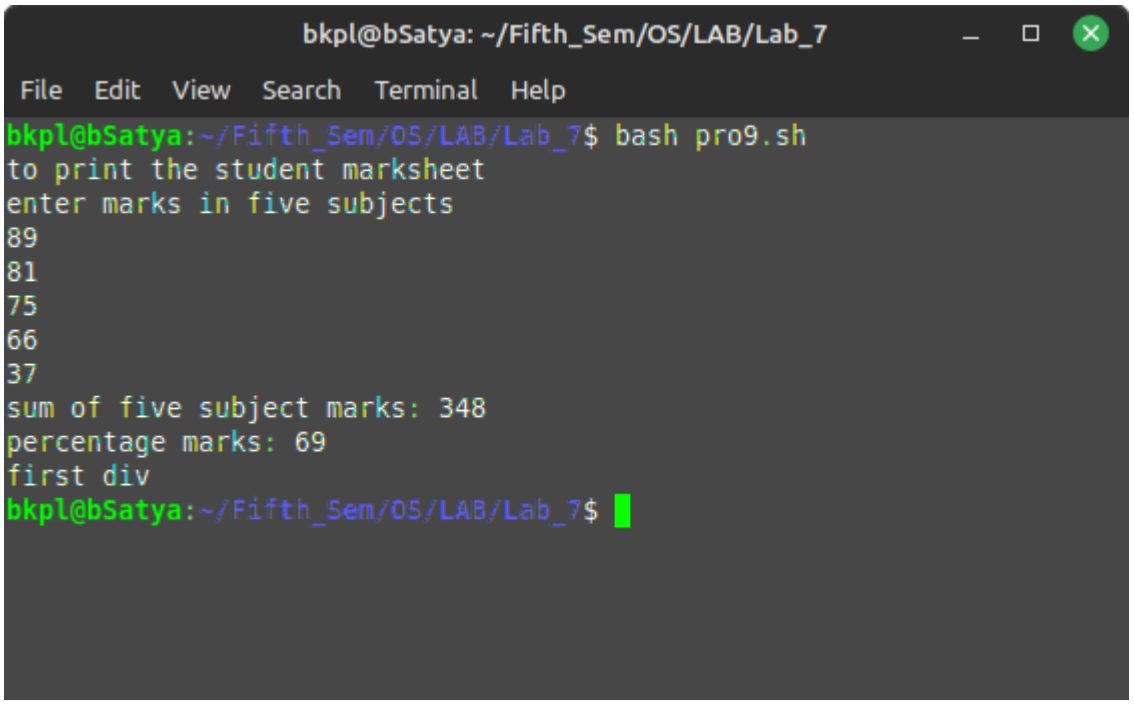
8.

```
echo to check no. is even or not
echo enter no
read a
e=` expr $a % 2 `
if [ $e -eq 0 ]
  then
        echo even no
else
        echo odd
fi
```



9.

```
echo to print the student marksheet
echo "enter marks in five subjects"
read s1
read s2
read s3
read s4
read s5
sum=` echo $s1 + $s2 + $s3 + $s4 +$s5 | bc `
echo "sum of five subject marks: $sum "
per=` echo $sum / 5 | bc `
echo "percentage marks: $per "
if [ $per -ge 60 ]
then
   echo first div
elif [ $per -ge 50 ]
then
   echo second div
elif [ $per -ge 40 ]
then
   echo third div
else
   echo fail
fi
```



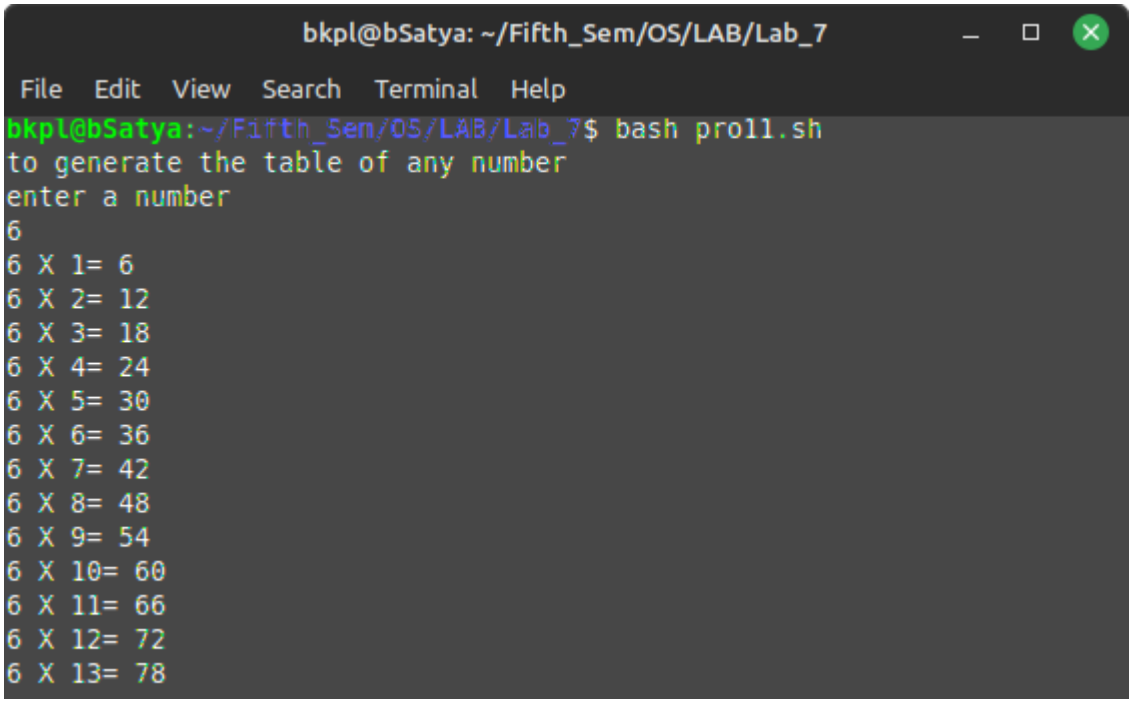
10.

```
echo "to find employee payroll"
read basic
if [ $basic -lt 1500 ]
then
hra=` echo $basic \* 10 / 100 | bc `
da=` expr $basic \* 90 / 100 100 | bc `
else
hra=500
da=` echo $basic \* 98 / 100 | bc `
```

```
echo dearness allowance is =rs $da
fi
gs=` echo $basic + $hra +$da | bc `
echo Gross salary=rs.$gs
```

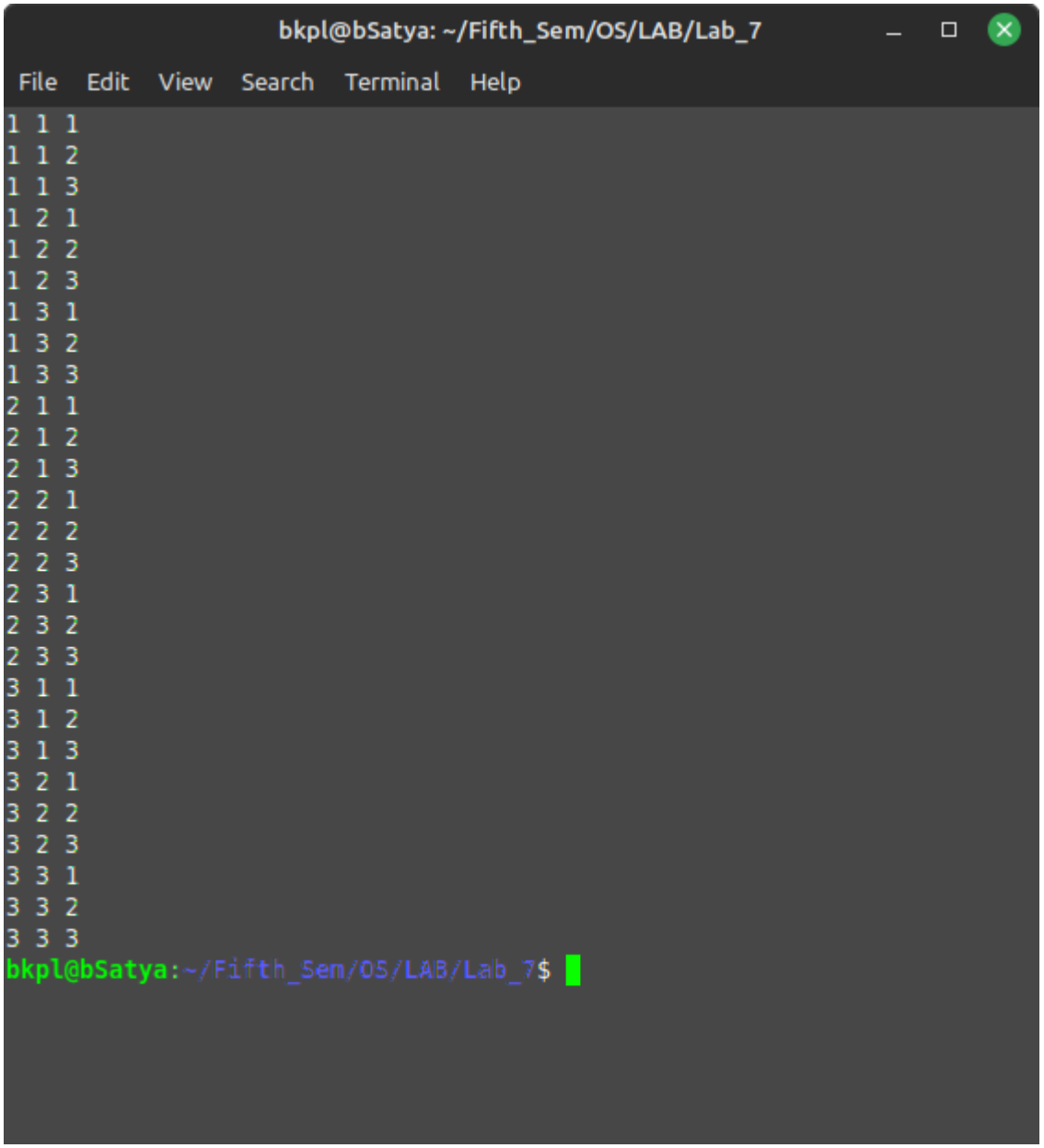
11.

```
echo to generate the table of any number
echo enter a number
read num
j=1
while [ $j -le 10 ]
do
res=` expr $j \* $num `
echo "$num X $j= $res "
j=` expr $j + 1 `
done
```



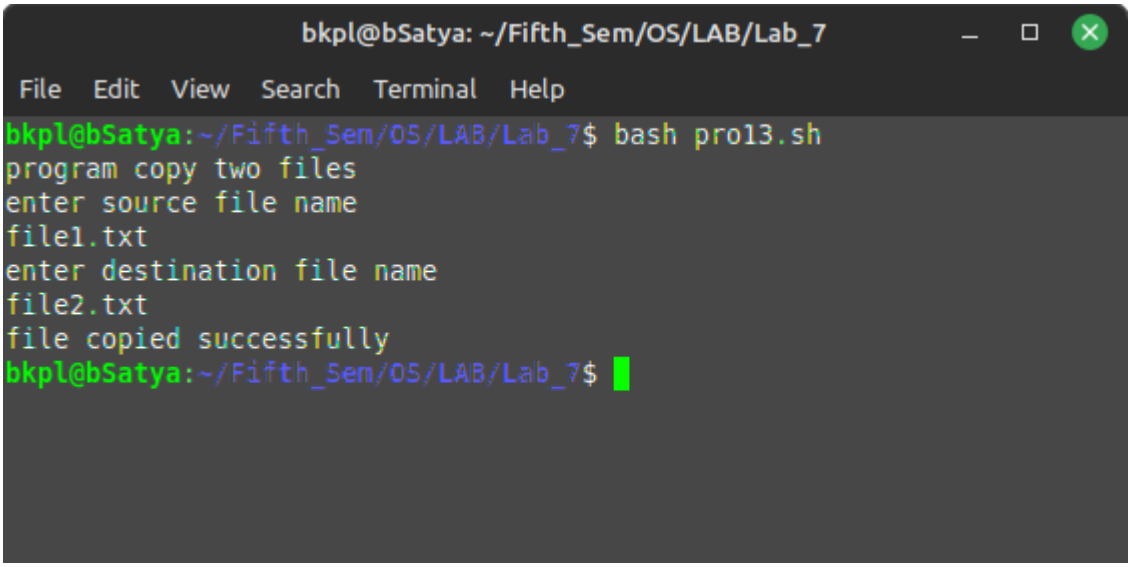
12.

```
echo program to generate all possible combinations of 1 2 3
clear
for i in 1 2 3
do
for j in 1 2 3
do
for k in 1 2 3
do
echo $i $j $k
done
done
done
```



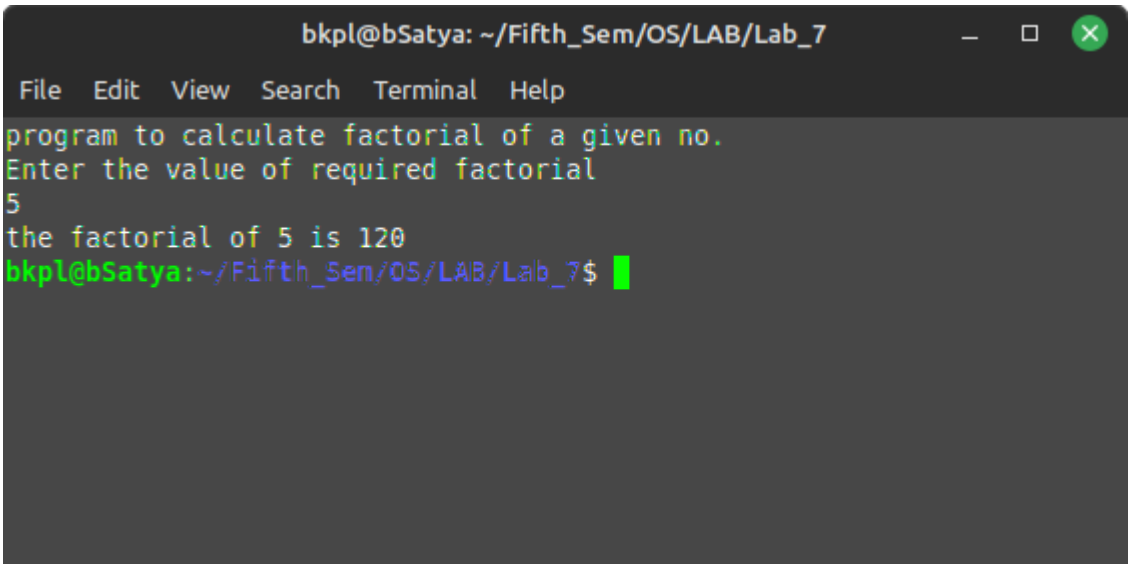
13.

```
echo program copy two files
echo "enter source file name
read file1
echo "enter destination file name"
read file2
if cp $file1 $file2 2>rav
then
        echo "file copied successfully"
else
        echo "unable to copy"
fi
```



14.

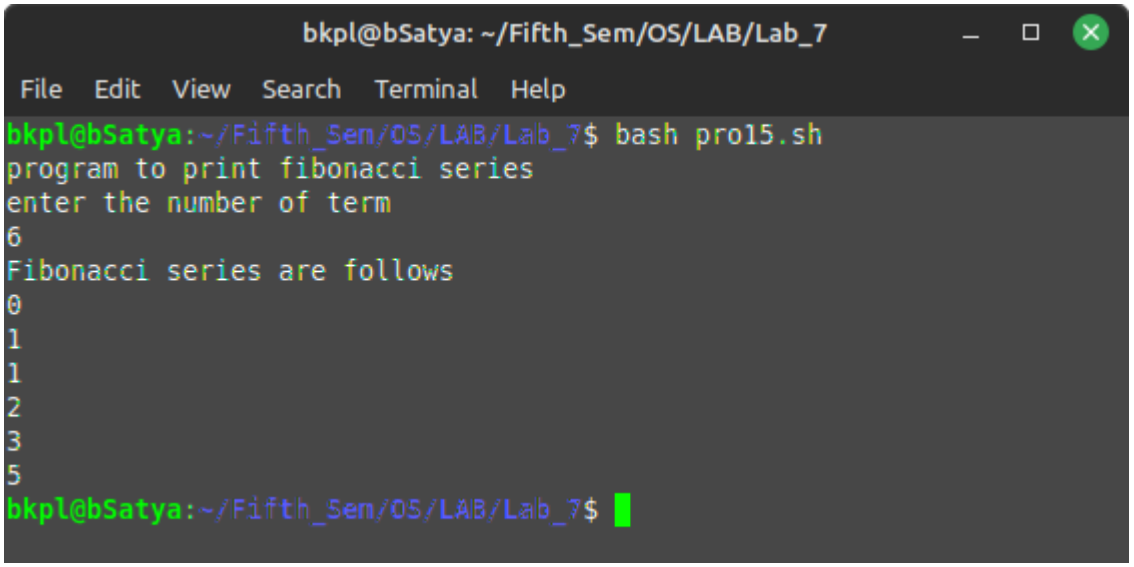
```
clear
echo program to calculate factorial of a given no.
j=1
k=1
echo Enter the value of required factorial
read num
while [ $j -le $num ]
do
k=` expr $k \* $j `
j=`expr $j + 1 `
done
echo the factorial of $num is $k
```



15.

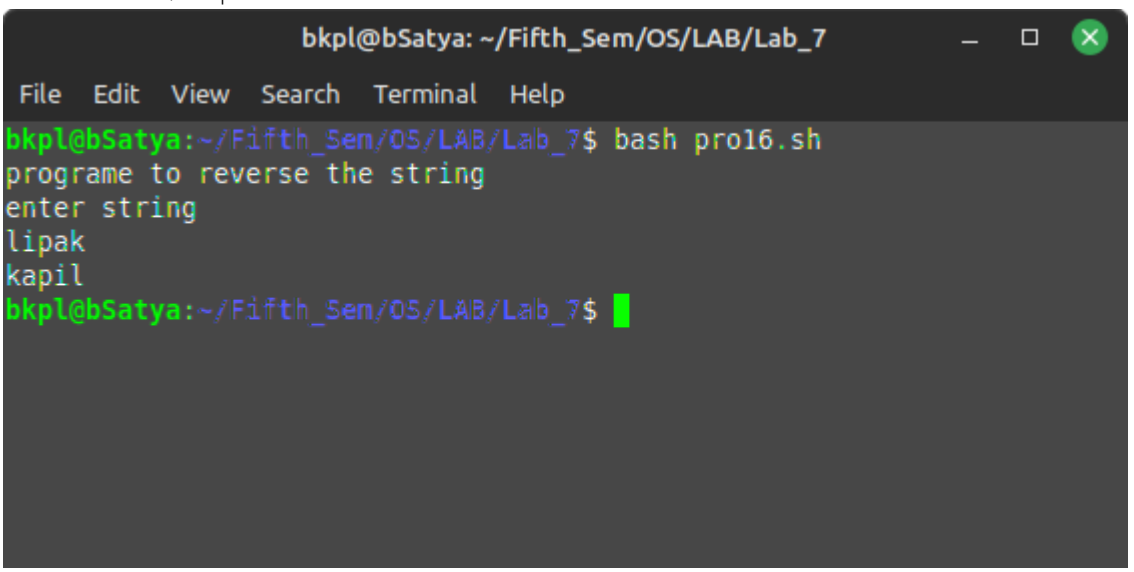
```
echo program to print fibonacci series
echo enter the number of term
read num
num=` expr $num - 2 `
a=0
b=1
echo "Fibonacci series are follows"
echo $a
```

```
echo $b
count=1
while [ $count -le $num ]
do
c=` expr $a + $b `
echo $c
a=$b
b=$c
count=` expr $count + 1 `
done
```



16.

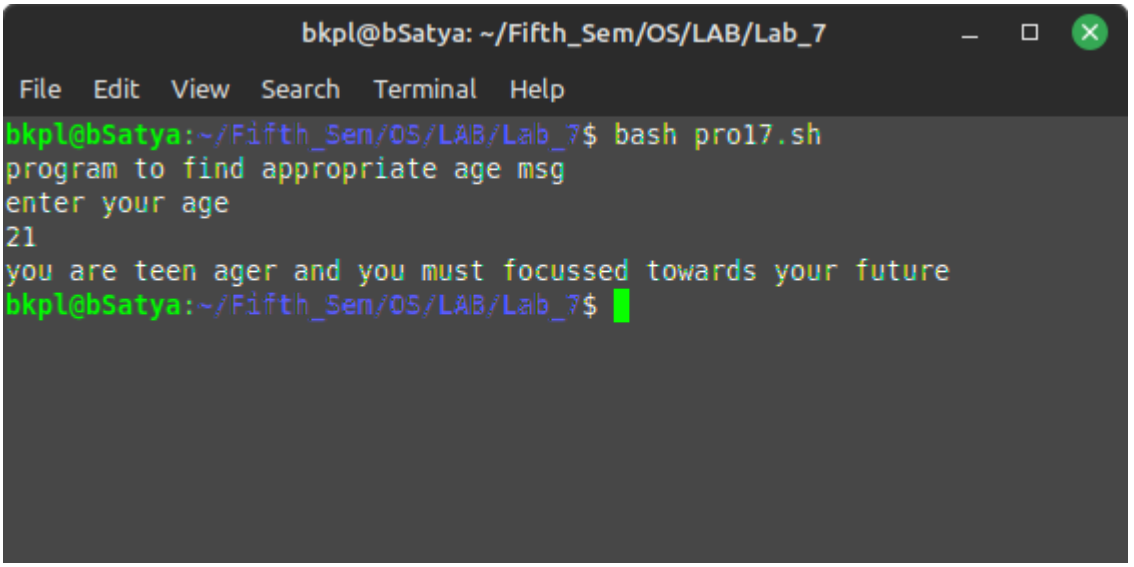
```
echo programe to reverse the string
echo enter string
read str
echo $str | rev
```



17.

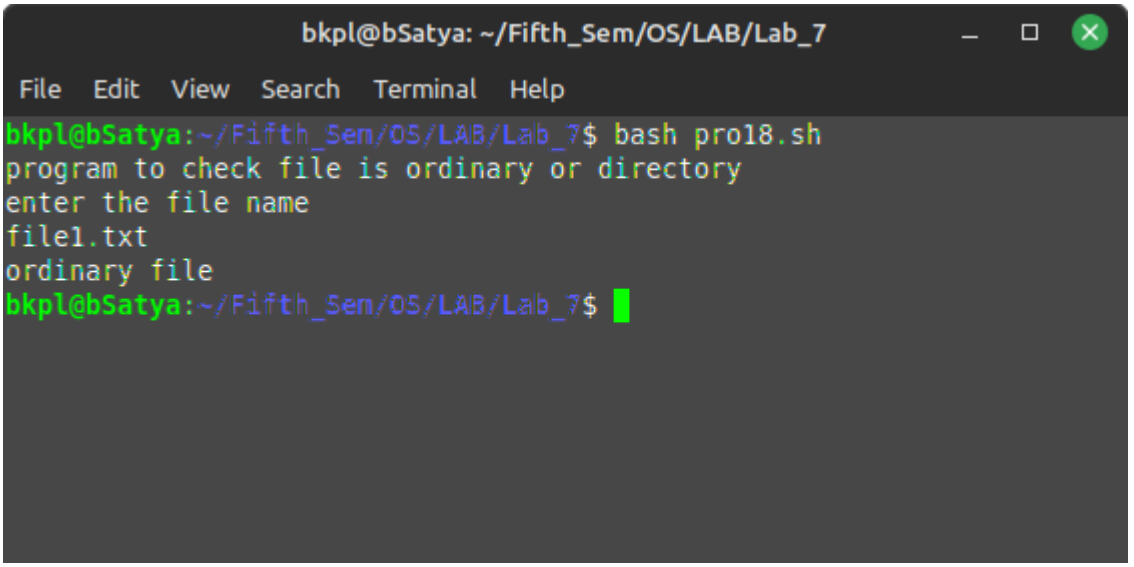
```
echo program to find appropriate age msg
echo enter your age
read age
if [ $age -ge 18 -o $age -lt 30 ]
 then
```

```
    echo you are teen ager and you must focussed towards your future
elif [ $age -ge 31 -o $age -lt 60]
 then
    echo take care your family
elif [ $age -ge 61 -o $age -ge 100]
 then
    echo ecpend your time happily
fi
```



18.

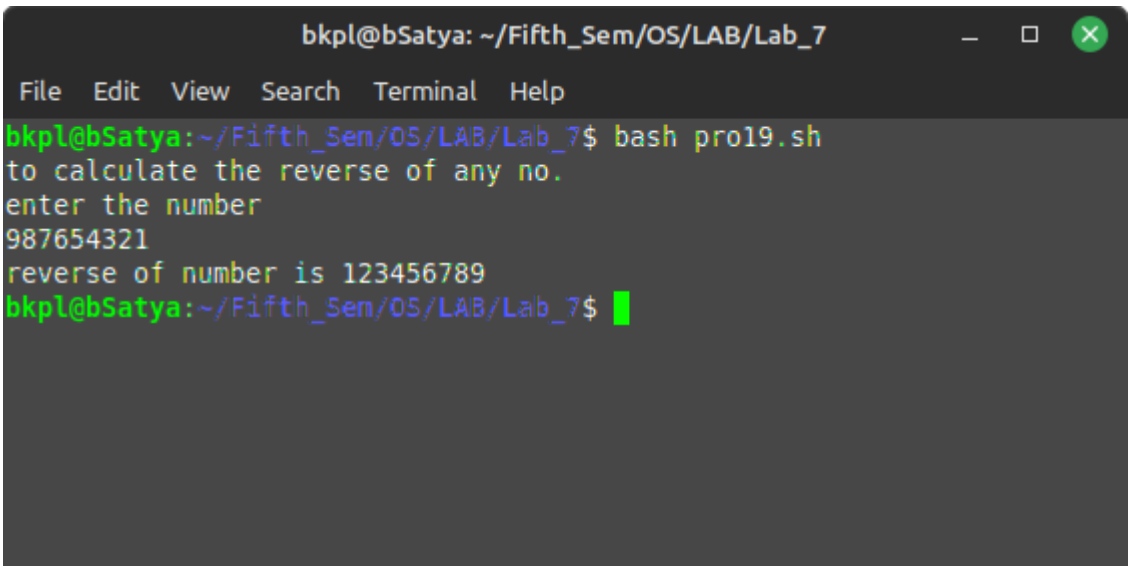
```
echo "program to check file is ordinary or directory"
echo "enter the file name"
read f1
if [ -d $f1 ]
then
echo "directory file"
else
echo "ordinary file"
fi
```



19.

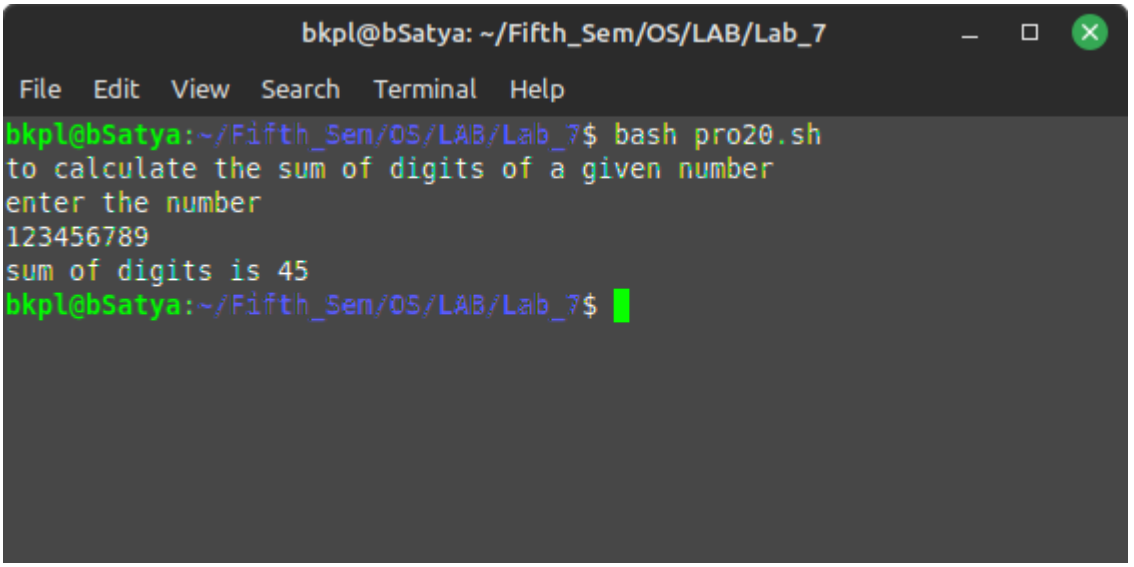
echo to calculate the reverse of any no.

```
echo enter the number
read no
rev=0
while [ $no -gt 0 ]
do
d=`expr $no % 10 `
rev=`expr $rev \* 10 `
rev=`expr $rev + $d `
no=`expr $no / 10 `
done
echo reverse of number is $rev
```



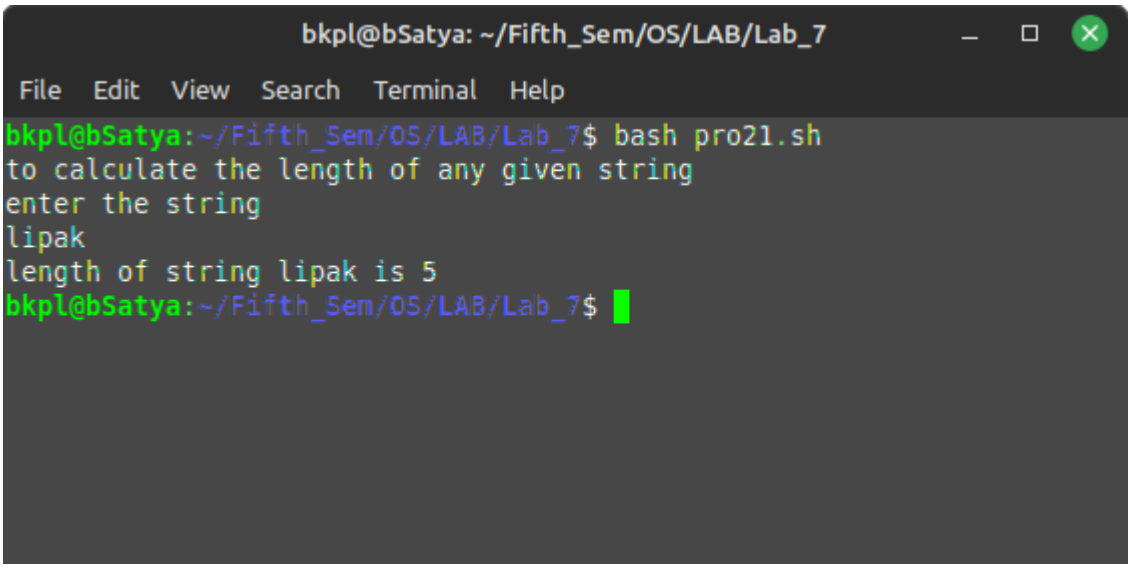
20.

```
echo "to calculate the sum of digits of a given number"
echo enter the number
read no
sum=0
while [ $no -gt 0 ]
do
d=`expr $no % 10 `
sum=`expr $sum + $d `
no=`expr $no / 10 `
done
echo "sum of digits is $sum"
```



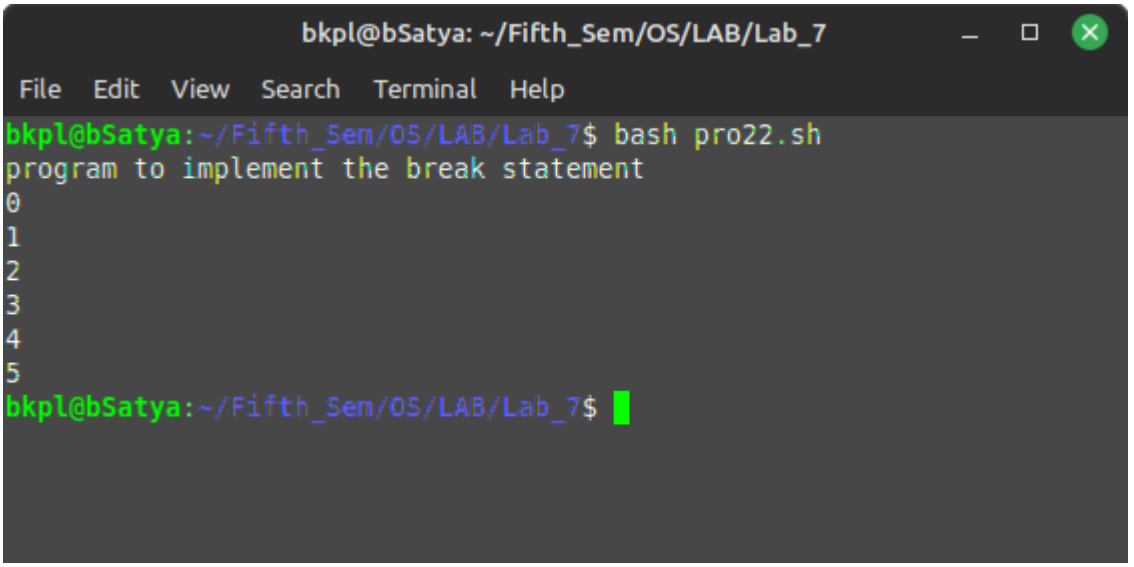
21.

```
echo to calculate the length of any given string
echo enter the string
read s
len=` expr $s | wc -c `
len=` expr $len - 1 `
echo length of string $s is $len
```



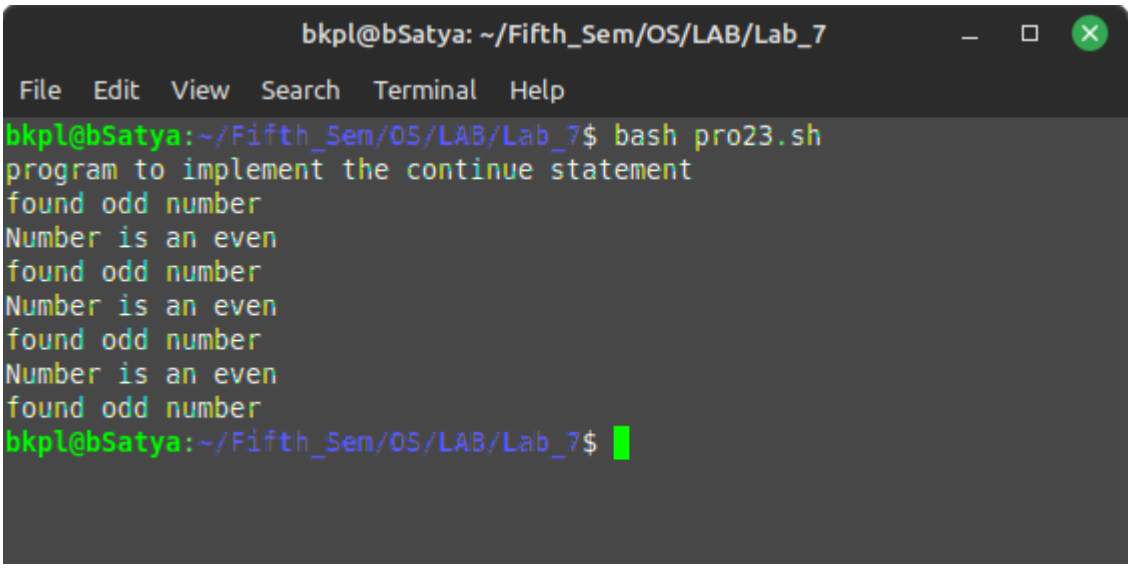
22.

```
echo "program to implement the break statement"
a=0
while [ $a -lt 10 ]
do
echo $a
if [ $a -eq 5 ]
then
break
fi
a=` expr $a + 1 `
done
```



23.

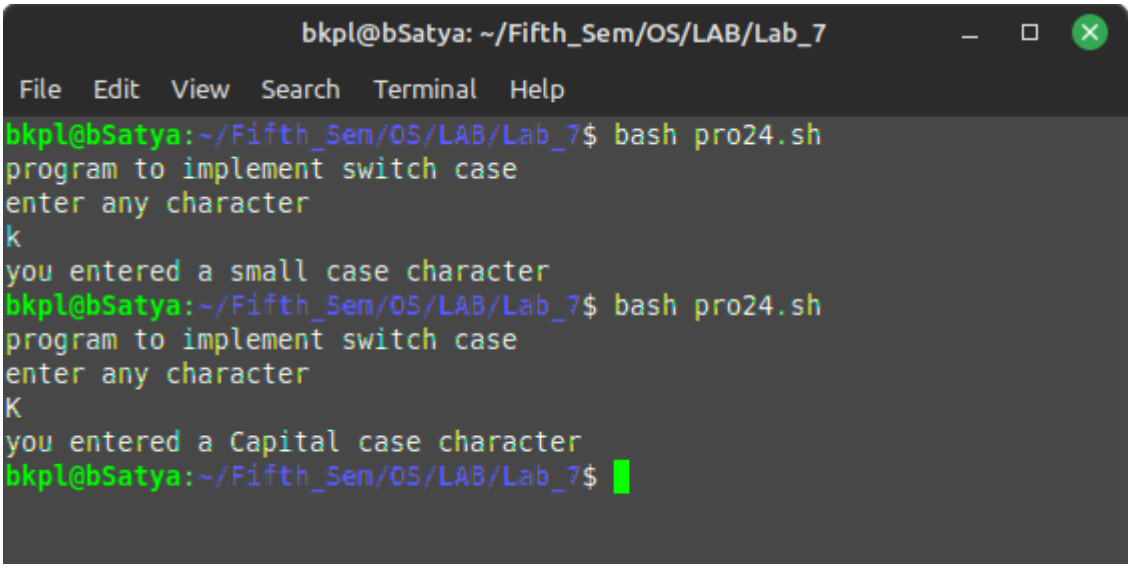
```
echo "program to implement the continue statement"
num="1 2 3 4 5 6 7"
for nu in $num
do
p=` expr $nu % 2 `
if [ $p -eq 0 ]
then
echo "Number is an even"
continue
fi
echo found odd number
done
```



24.

```
echo program to implement switch case
echo enter any character
read char
case $char in [A-Z])
echo you entered a Capital case character;;
```

```
[a-z])
echo you entered a small case character;;
[0-9])
echo you entered a digit;;
?)
 echo you entered special character;;
*)
 echo you entered more than one character;;
esac
```

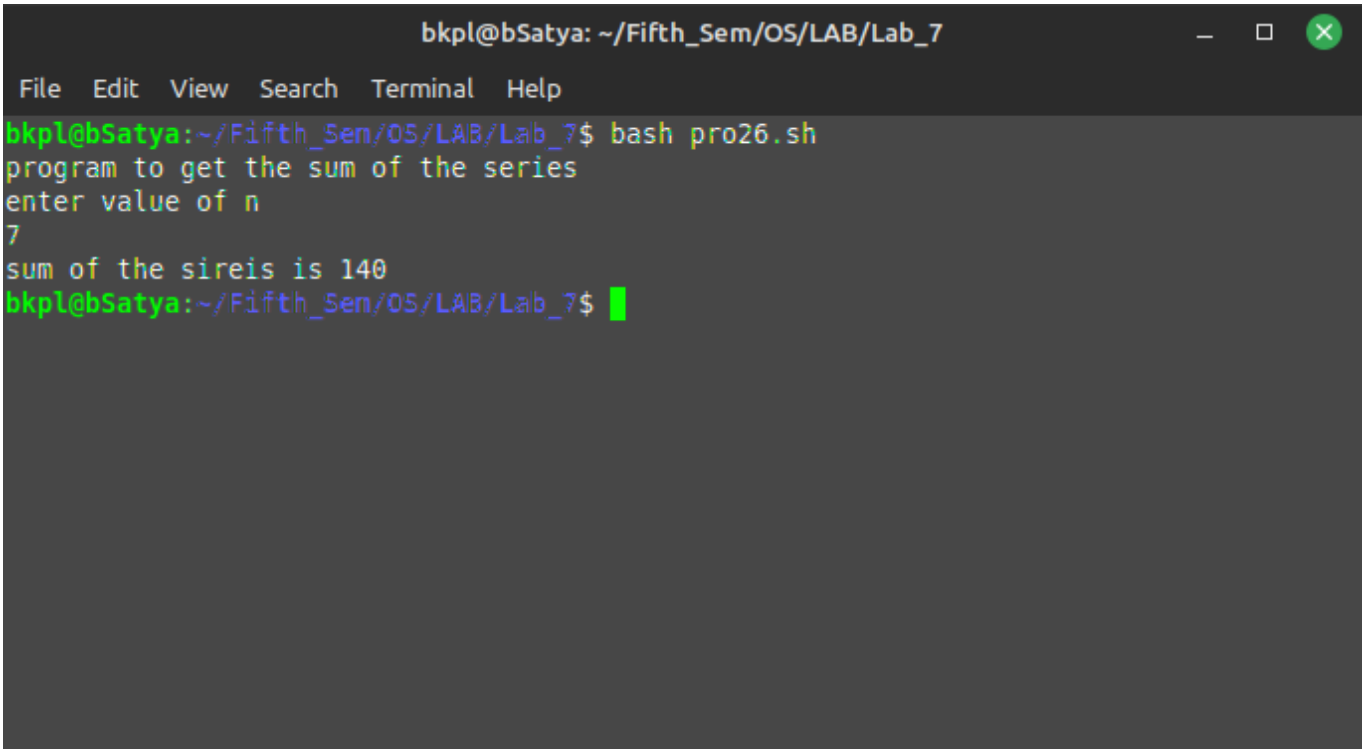


25.

```
echo program to generate prime no. between two given inputs
echo enter low range
read low
echo enter max range
read max
while [ $low -le $max ]
do
i=2
while [ $i -le `expr $low - 1 ` ]
Do
if [ ` expr $low % $i ` -eq 0 ]
then
break
else
i=` expr $i + 1 `
fi
done
if [ $low -eq $i ]
then
echo $low
fi
low=` expr $low + 1 `
done
```

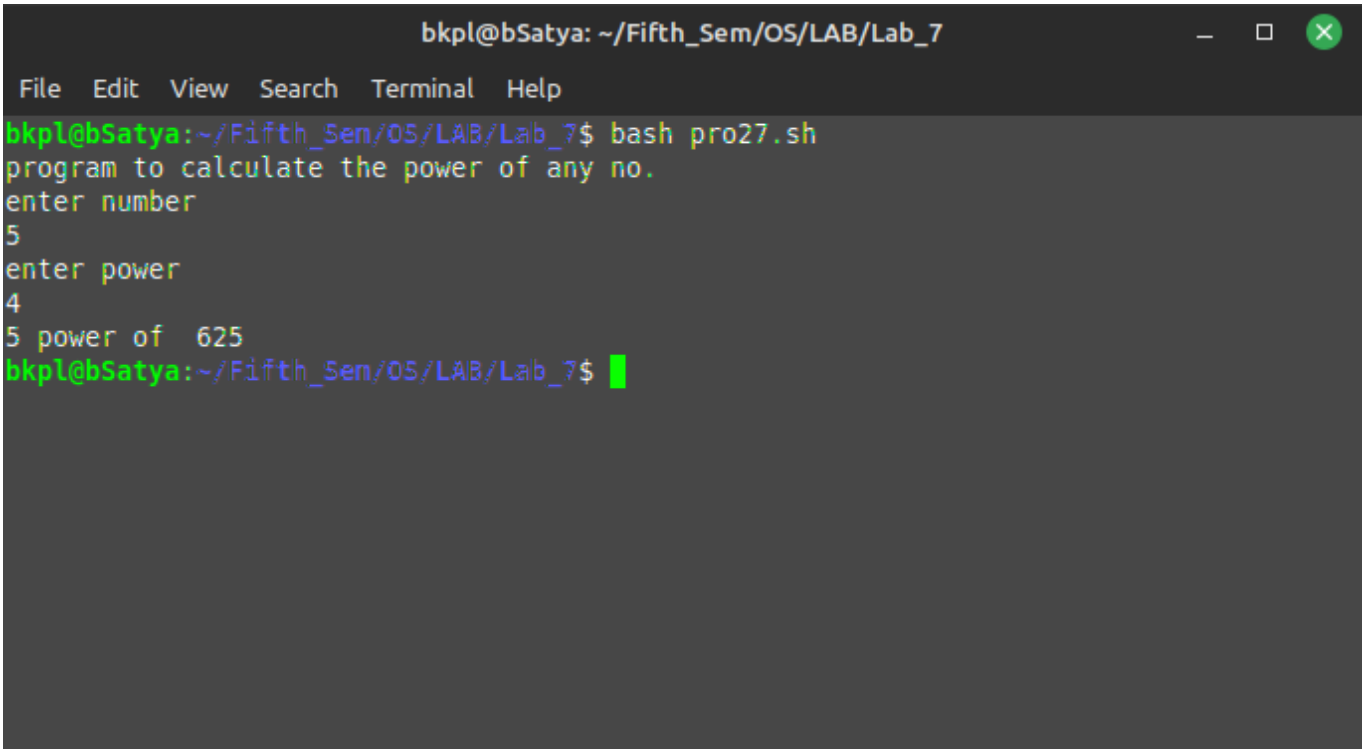
26.

```
echo "program to get the sum of the series"
echo enter value of n
read n
a=` expr 2 \* $n + 1 `
b=` expr $n + 1 `
c=` expr $a \* $b `
d=` expr $c \* $n `
sum=` expr $d / 6 `
echo "sum of the sireis is $sum"
```



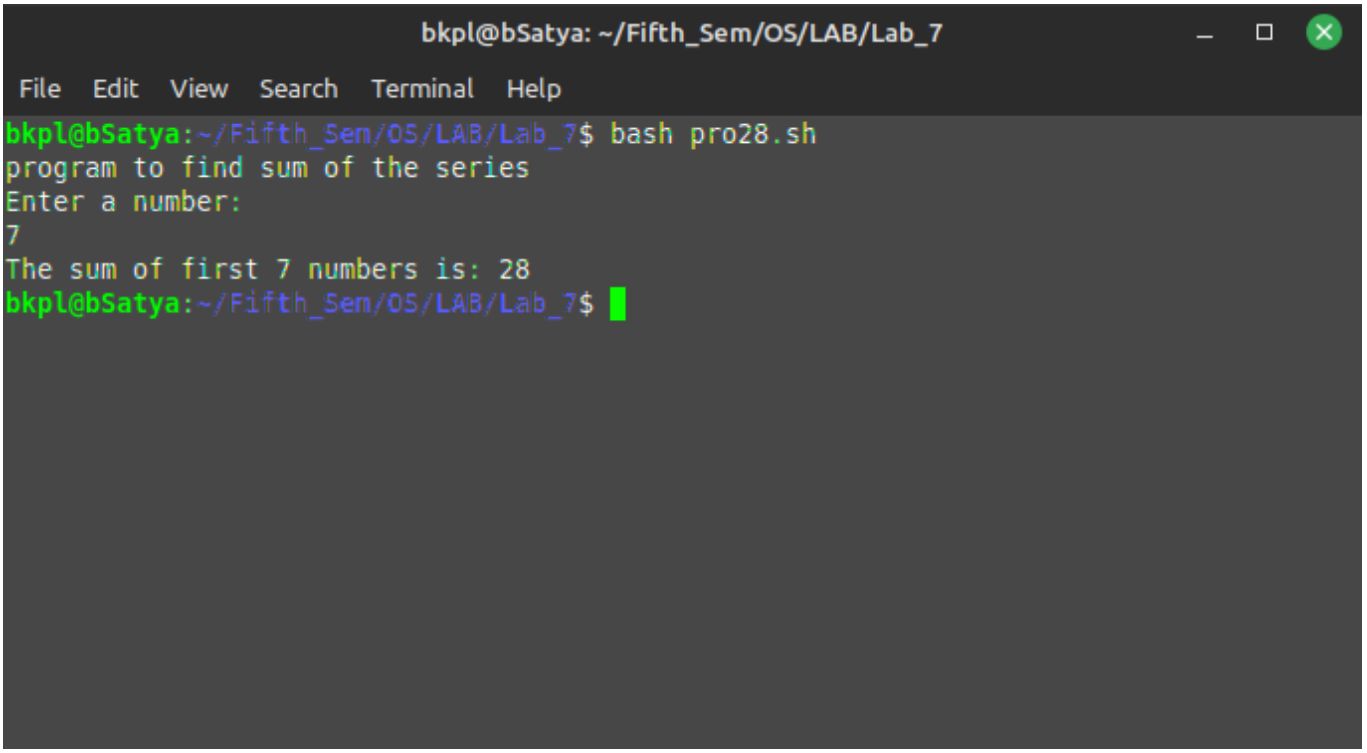
27.

```
echo program to calculate the power of any no.
echo enter number
read num
echo enter power
read pow
counter=0
ans=1
while [ $pow -ne $counter ]
do
ans=` expr $ans \* $num `
counter=` expr $counter + 1 `
done
echo "$num power of $power $ans"
```



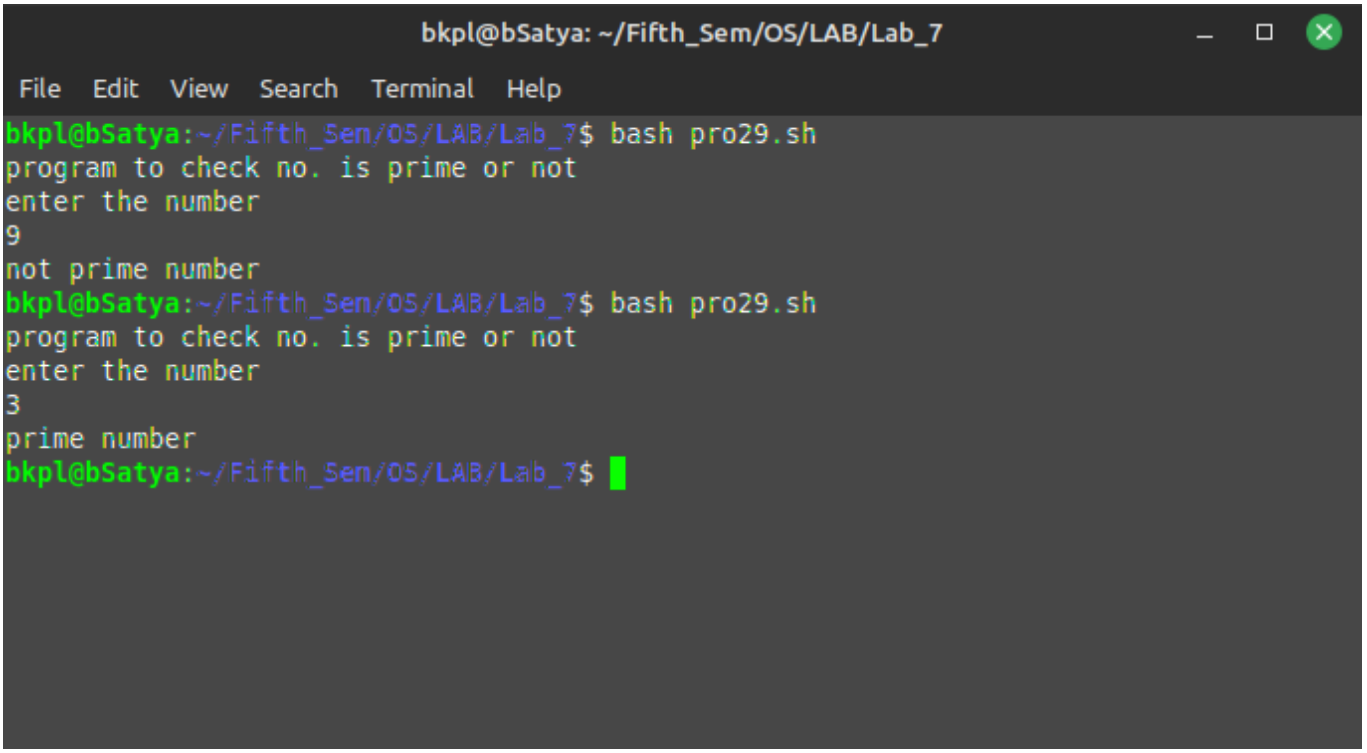
28.

```
echo program to find sum of the series
echo "Enter a number:"
read num
i=1
sum=0
while [ $i -le $num ]
do
sum=`expr $sum + $i`
i=`expr $i + 1`
done
echo "The sum of first $num numbers is: $sum"
```



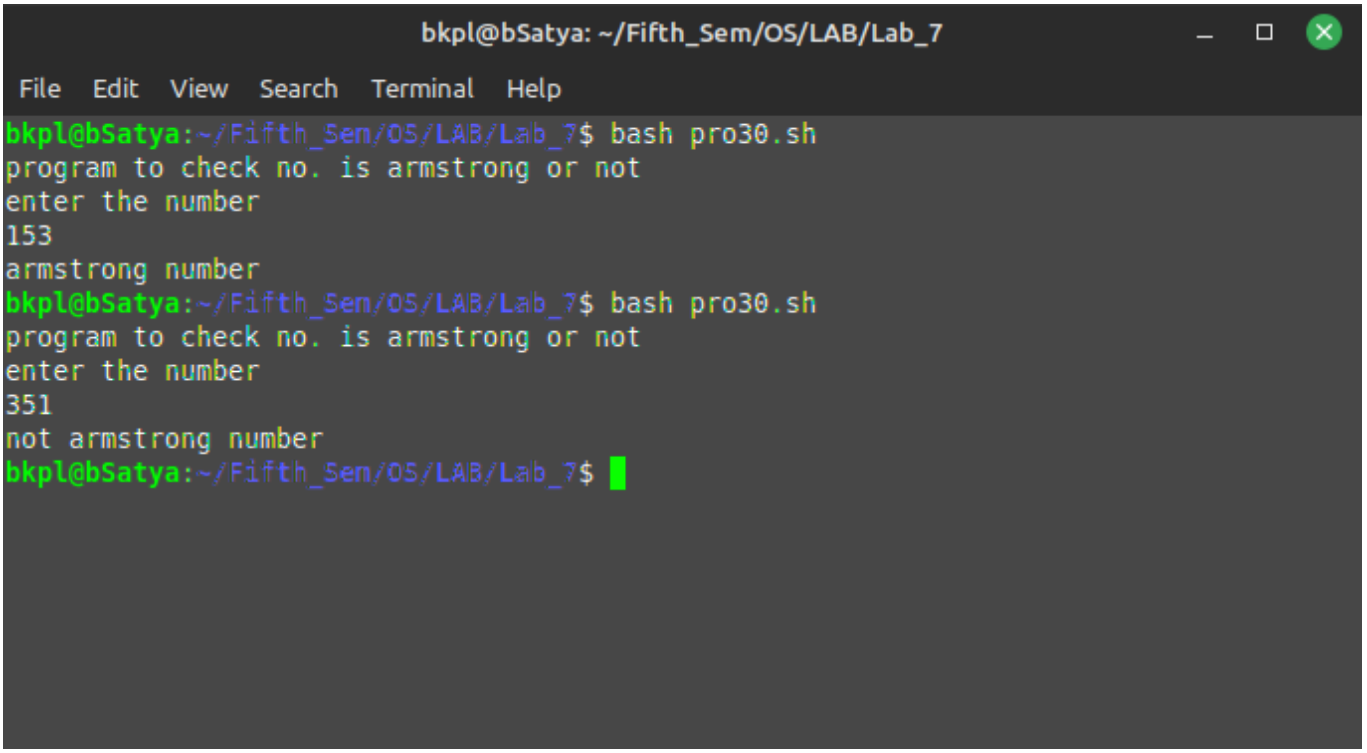
29.

```
echo program to check no. is prime or not
echo enter the number
read no
count=2
while [ $count -le `expr $no / 2 + 1 ` ]
do
if [ ` expr $no % $count ` -eq 0 ]
then
f=1
break
else
f=0
fi
count=` expr $count + 1 `
done
if [ $f -eq 0 ]
then
echo prime number
else
echo not prime number
fi
```



30.

```
echo program to check no. is armstrong or not
echo enter the number
read no
sum=0
dup=$no
while [ $no -gt 0 ]
do
d=`expr $no % 10 `
d=`expr $d \* $d \* $d `
sum=`expr $sum + $d `
no=`expr $no / 10 `
done
if [ $sum -eq $dup ]
then
echo "armstrong number"
else
echo not armstrong number
fi
```



Conclusion

- Learned about shell scripting
- Learned about different shell scripting commands
- Learned to program on the shell