VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.

LAB REPORT
on

UNIX SHELL AND PROGRAMMING

Submitted by

ANKITH S(1BM20CS017)

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019
October-2022 to Feb-2023

B. M. S. College of Engineering,

Bull Temple Road, Bangalore 560019
(Affiliated To Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Lab work entitled “LAB COURSE **UNIX SHELL AND PROGRAMMING**” carried out by **ANKITH S (1BM20CS017),** who is bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2022. The Lab report has been approved as it satisfies the academic requirements in respect of a **Unix Shell and Programming - (20CS5PCUSP)** work prescribed for the said degree.

Dr. Kayarvizhy N
Associate Professor
Department of CSE
BMSCE, Bengaluru

Dr. Jyothi S Nayak
Professor and Head
Department of CSE
BMSCE, Bengaluru

`

Index

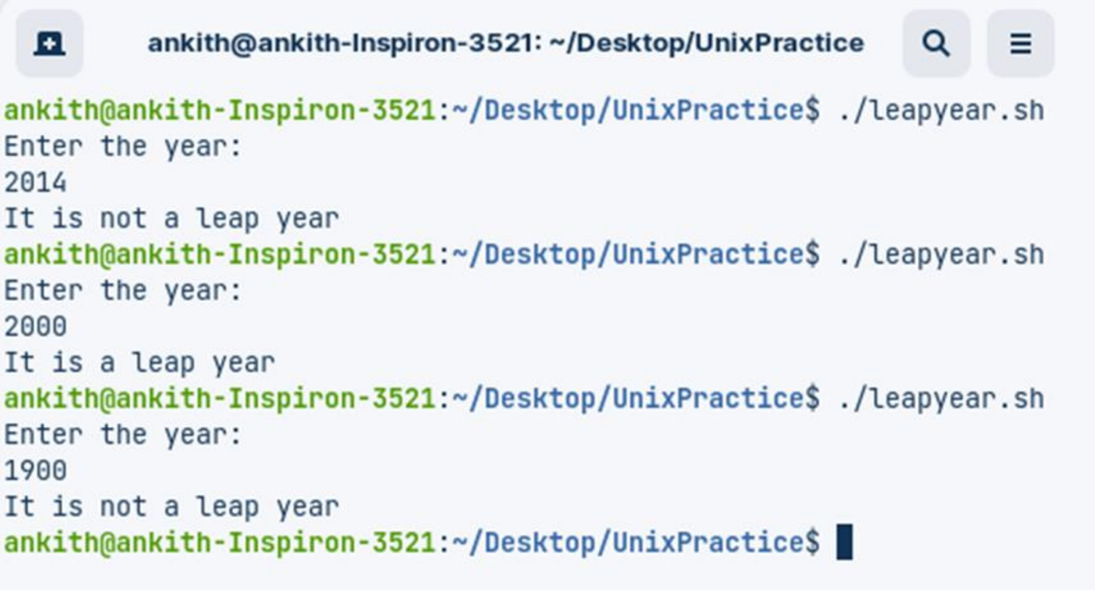
Sl. No.	Date	Experiment Title	Page No.
1.		Shell script to find if the given year is leap or not	4
2		Shell script to find the area of a circle	5
3		Shell script to check whether the number is zero/ positive/ negative	6
4		Shell script to find the biggest of three numbers	7
5		Shell script to find the factorial of a number	8
6		Shell script to compute the gross salary of an employee	9
7		Shell script to convert the temperature Fahrenheit to Celsius	10
8		Shell script to perform arithmetic operations on given two numbers	11
9		Shell script to find the sum of even numbers up to n	12
10		Shell script to print the combinations of numbers 123	13
11		Shell script to find the power of a number	14
12		Shell script to find the sum of n natural numbers	15
13		Shell script to display the pass class of a student	16
14		Shell script to find the Fibonacci series up to n	17
15		Shell script to count the number of vowels of a string	18
16		Shell script to check number of lines, words, characters in a file	19
17		Write a C/C++ program to that outputs the contents of its environment list	20
18		Write a C/C++ program to emulate the Unix ln command	21
19		Write a C/C++ POSIX compliant program that prints the POSIX defined Configuration options supported on any given system using feature test macros.	22
20		Write a C/C++ program which demonstrates Interprocess Communication between a reader process and a writer process. Use mkfifo, open, read, write and close apis in your program.	24

Experiment No 1

Shell script to find if the given year is leap or not

```
#!/bin/bash
echo "Enter an Year: "
read year
if [ $((year % 4)) -eq 0 ]
then
  if [ $((year % 100)) -eq 0 ]
    then
    if [ $((year % 400)) -eq 0 ]
          then
        echo "$year is a leap year"
    else
           echo "$year is not a leap year"
    fi
  else
  echo "$year is a leap year"
  fi
else
 echo "$year is not a leap year"
fi
```

Output



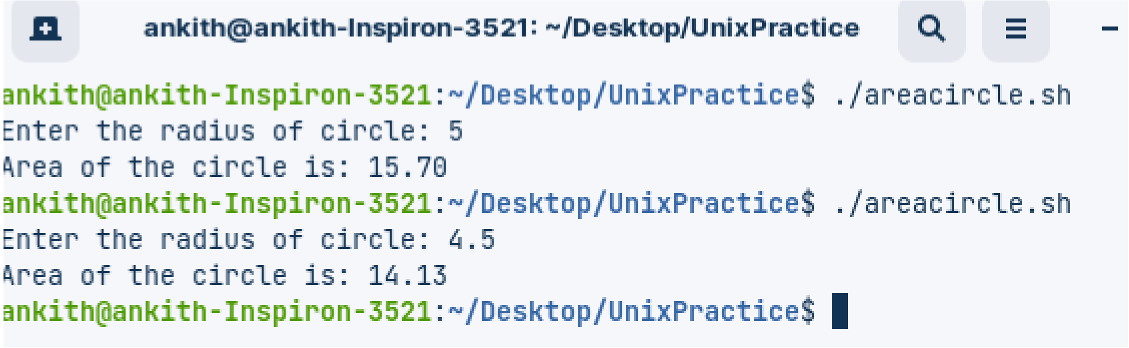
```
ankith@ankith-Inspiron-3521:~/Desktop/UnixPractice$ ./leapyear.sh
Enter the year:
2014
It is not a leap year
ankith@ankith-Inspiron-3521:~/Desktop/UnixPractice$ ./leapyear.sh
Enter the year:
2000
It is a leap year
ankith@ankith-Inspiron-3521:~/Desktop/UnixPractice$ ./leapyear.sh
Enter the year:
1900
It is not a leap year
ankith@ankith-Inspiron-3521:~/Desktop/UnixPractice$ 
```

Experiment No 2

Shell script to find the area of a circle

```
#!/bin/bash

echo "\nEnter the radius of a circle : "

read r

d=$(echo "scale=2;2 * $r"| bc) #Diameter

area=$(echo "scale=2; 22/7 * ($r * $r)" | bc)

circumference=$(echo "scale=2; 22/7 * $d"| bc)

echo "\nArea of circle is : $area"

echo "\nCircumference of circle is : $circumference \n"
```

Output



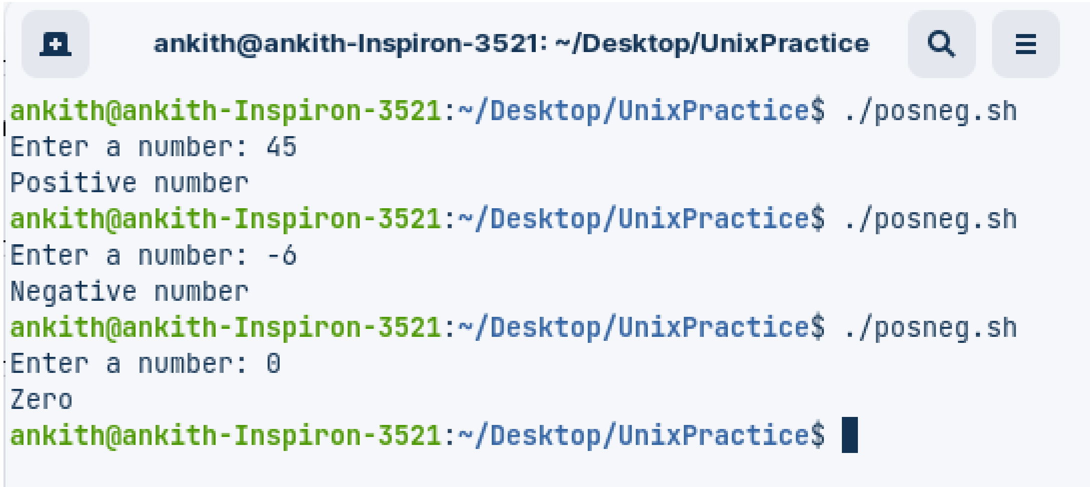
```
ankith@ankith-Inspiron-3521:~/Desktop/UnixPractice$ ./areacircle.sh
Enter the radius of circle: 5
Area of the circle is: 15.70
ankith@ankith-Inspiron-3521:~/Desktop/UnixPractice$ ./areacircle.sh
Enter the radius of circle: 4.5
Area of the circle is: 14.13
ankith@ankith-Inspiron-3521:~/Desktop/UnixPractice$
```

Experiment No 3

Shell script to check whether the number is zero/ positive/ negative

```
#!/bin/bash
echo "Enter the number : "
read num
if [ $num -gt 0 ]
then
   echo "$num is positive"
elif [ $num -lt 0 ]
then
   echo "$num is negative"
else
   echo "$num is zero"
fi
```

Output



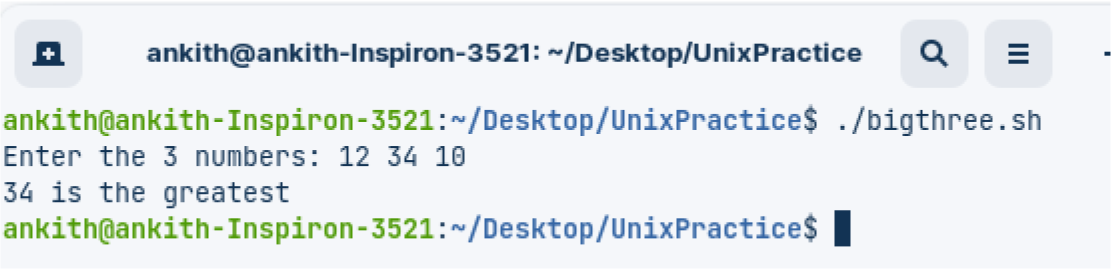
```
ankith@ankith-Inspiron-3521:~/Desktop/UnixPractice$ ./posneg.sh
Enter a number: 45
Positive number
ankith@ankith-Inspiron-3521:~/Desktop/UnixPractice$ ./posneg.sh
Enter a number: -6
Negative number
ankith@ankith-Inspiron-3521:~/Desktop/UnixPractice$ ./posneg.sh
Enter a number: 0
Zero
ankith@ankith-Inspiron-3521:~/Desktop/UnixPractice$ 
```

Experiment No 4

Shell script to find the biggest of three numbers

```
#!/bin/bash
echo "Enter first number : "
read num1
echo "Enter second number : "
read num2
echo "Enter third number : "
read num3
if [ $num1 -gt $num2 ] && [ $num1 -gt $num3 ]
then
   echo "\n$num1 is the greatest"
elif [ $num2 -gt $num1 ] && [ $num2 -gt $num3 ]
then
   echo "\n$num2 is the greatest"
else
   echo "\n$num3 is the greatest"
fi
```

Output



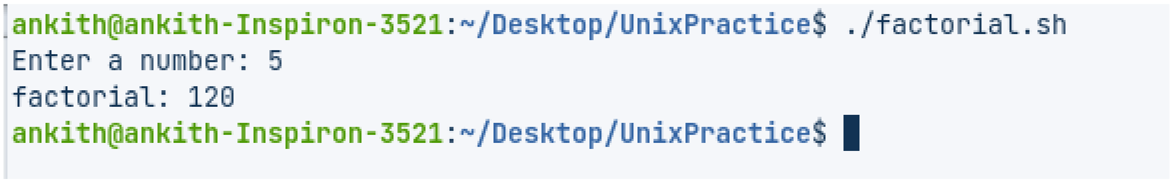
```
ankith@ankith-Inspiron-3521:~/Desktop/UnixPractice$ ./bigthree.sh
Enter the 3 numbers: 12 34 10
34 is the greatest
ankith@ankith-Inspiron-3521:~/Desktop/UnixPractice$ 
```

Experiment No 5

Shell script to find the factorial of a number

```
#!/bin/bash

echo "ENTER THE NUMBER: "

read n

fact=1

while [ $n -gt 1 ]

do

   fact=$(( fact * n))

   n=$((n-1 ))

done

echo "FACTORIAL IS: "

echo $fact
```

Output



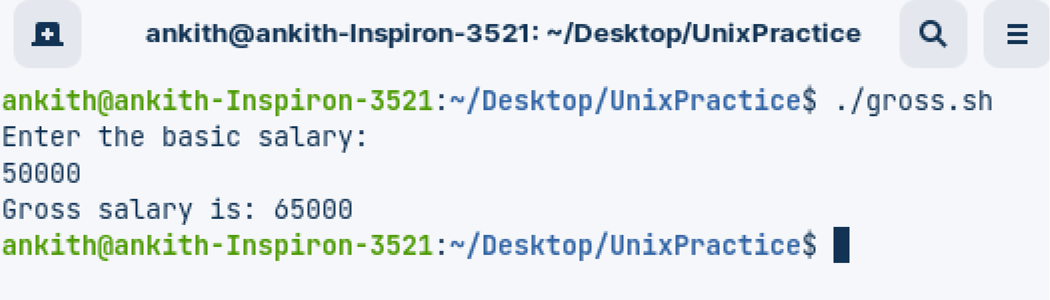
```
ankith@ankith-Inspiron-3521:~/Desktop/UnixPractice$ ./factorial.sh
Enter a number: 5
factorial: 120
ankith@ankith-Inspiron-3521:~/Desktop/UnixPractice$ 
```

Experiment No 6

Shell script to compute the gross salary of an employee

```
#!/bin/bash


echo "Enter the basic salary: "

read sal

grossSal=$(($sal + ($sal/5) + ($sal/10)))

echo "Gross salary is: $grossSal"
```

Output



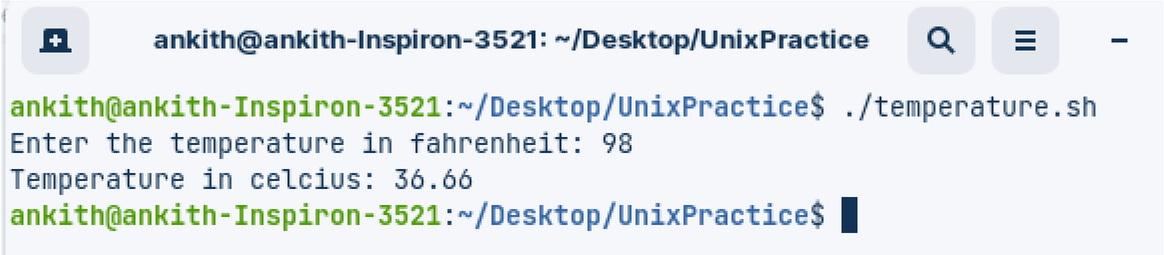
```
ankith@ankith-Inspiron-3521:~/Desktop/UnixPractice$ ./gross.sh
Enter the basic salary:
50000
Gross salary is: 65000
ankith@ankith-Inspiron-3521:~/Desktop/UnixPractice$ 
```

Experiment No 7

Shell script to convert the temperature Fahrenheit to Celsius

```
#!/bin/bash

echo "Enter temperature in F : "

read f

c=$(echo "scale=2;(5/9)*($f-32)"|bc)

echo "$f °F = $c °C"
```

Output:



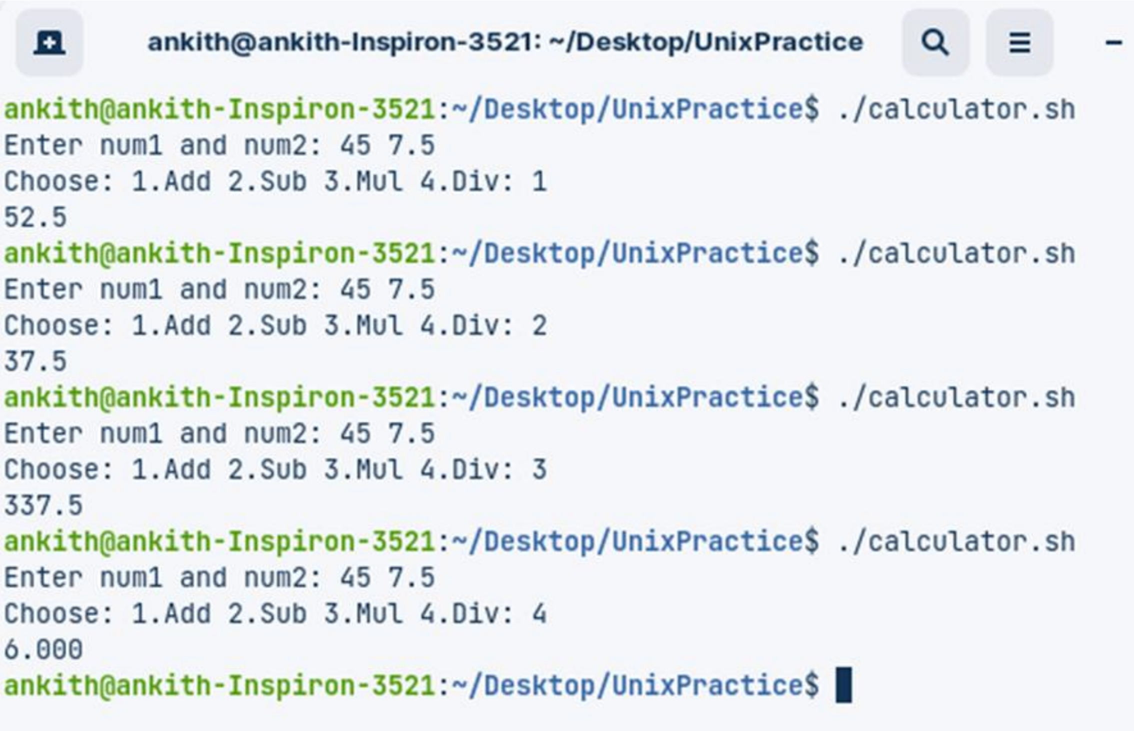
```
ankith@ankith-Inspiron-3521:~/Desktop/UnixPractice$ ./temperature.sh
Enter the temperature in fahrenheit: 98
Temperature in celcius: 36.66
ankith@ankith-Inspiron-3521:~/Desktop/UnixPractice$ 
```

Experiment No 8

Shell script to perform arithmetic operations on given two numbers

```
#!/bin/bash

read -p "Enter num1 and num2: " n1 n2

read -p "Choose: 1.Add 2.Sub 3.Mul 4.Div: " op

if [ $op -eq 1 ]

then echo "$n1 + $n2" | bc

elif [ $op -eq 2 ]

then echo "$n1 - $n2" | bc

elif [ $op -eq 3 ]

then echo "$n1 * $n2" | bc

elif [ $op -eq 4 ]

then echo "scale=3; $n1 / $n2" | bc

else echo "Invalid input"

fi
```

Output



```
ankith@ankith-Inspiron-3521:~/Desktop/UnixPractice$ ./calculator.sh
Enter num1 and num2: 45 7.5
Choose: 1.Add 2.Sub 3.Mul 4.Div: 1
52.5
ankith@ankith-Inspiron-3521:~/Desktop/UnixPractice$ ./calculator.sh
Enter num1 and num2: 45 7.5
Choose: 1.Add 2.Sub 3.Mul 4.Div: 2
37.5
ankith@ankith-Inspiron-3521:~/Desktop/UnixPractice$ ./calculator.sh
Enter num1 and num2: 45 7.5
Choose: 1.Add 2.Sub 3.Mul 4.Div: 3
337.5
ankith@ankith-Inspiron-3521:~/Desktop/UnixPractice$ ./calculator.sh
Enter num1 and num2: 45 7.5
Choose: 1.Add 2.Sub 3.Mul 4.Div: 4
6.000
ankith@ankith-Inspiron-3521:~/Desktop/UnixPractice$ 
```

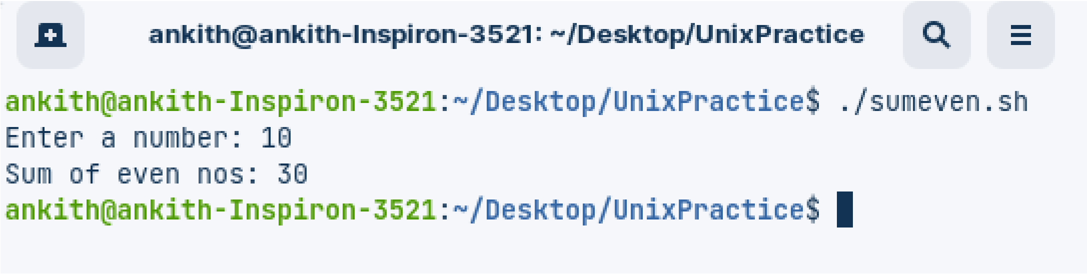
Experiment No 9

Shell script to find the sum of even numbers upto n

Program:

```
#!/bin/bash
sum=0
read -p "Enter maximum limit of Even Numbers : " m
for ((i = 0; i < m; i++)); do
    if [[ $i%2 -eq 0 ]]; then
        sum=$(expr $sum + $i)
    fi
done
echo $sum
```

Output



```
ankith@ankith-Inspiron-3521:~/Desktop/UnixPractice$ ./sumeven.sh
Enter a number: 10
Sum of even nos: 30
ankith@ankith-Inspiron-3521:~/Desktop/UnixPractice$ 
```

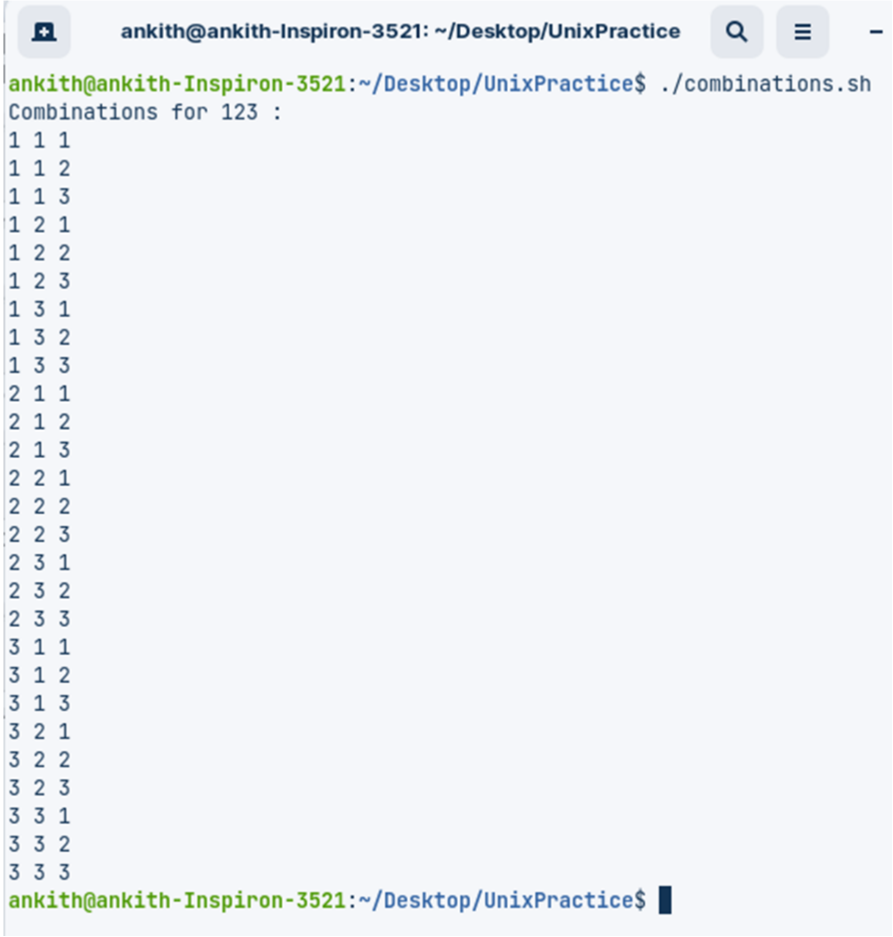
Experiment No 10

Shell script to print the combinations of numbers 123

Program:

```
#!/bin/bash

echo "Combinations for 123 :"

for ((i = 1; i <= 3; i++)); do

    for ((j = 1; j <= 3; j++)); do

        for ((k = 1; k <= 3; k++)); do

            echo $i $j $k

        done

    done

done
```

Output



```
ankith@ankith-Inspiron-3521:~/Desktop/UnixPractice$ ./combinations.sh
Combinations for 123 :
1 1 1
1 1 2
1 1 3
1 2 1
1 2 2
1 2 3
1 3 1
1 3 2
1 3 3
2 1 1
2 1 2
2 1 3
2 2 1
2 2 2
2 2 3
2 3 1
2 3 2
2 3 3
3 1 1
3 1 2
3 1 3
3 2 1
3 2 2
3 2 3
3 3 1
3 3 2
3 3 3
ankith@ankith-Inspiron-3521:~/Desktop/UnixPractice$
```

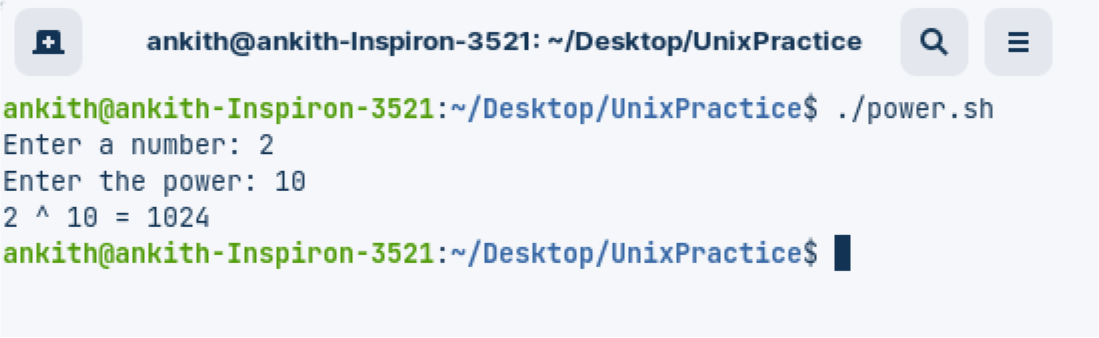
Experiment No 11

Shell script to find the power of a number

Program:

```
#!/bin/bash

echo "Enter base"

read a

echo "Enter power"

read b

res=1

for ((i = 1; i <= b; i++)); do

    res=`expr $res \* $a`

done

echo $res
```

Output



```
ankith@ankith-Inspiron-3521:~/Desktop/UnixPractice$ ./power.sh
Enter a number: 2
Enter the power: 10
2 ^ 10 = 1024
ankith@ankith-Inspiron-3521:~/Desktop/UnixPractice$ 
```

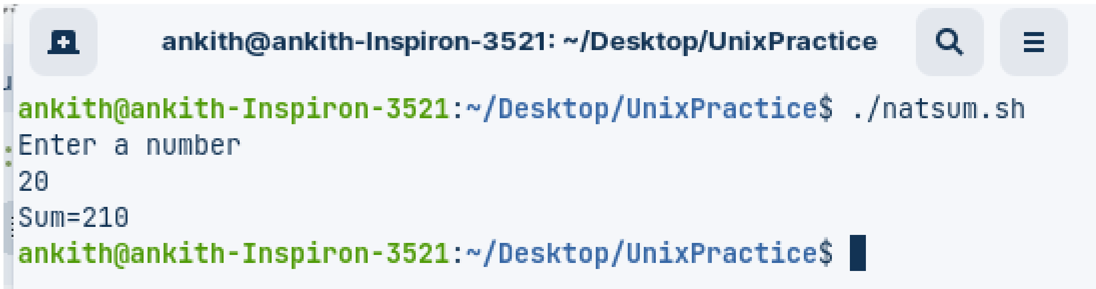
Experiment No 12

Shell script to find the sum of n natural numbers

Program:

```
#!/bin/bash
echo "Enter a number"
read n
i=1
sum=0
while [ $i -le $n ]
  do
   echo "$i"
   sum=$(( $sum + $i ))
        i=$(( $i + 1 ))
  done
echo "Sum=$sum"
```

Output



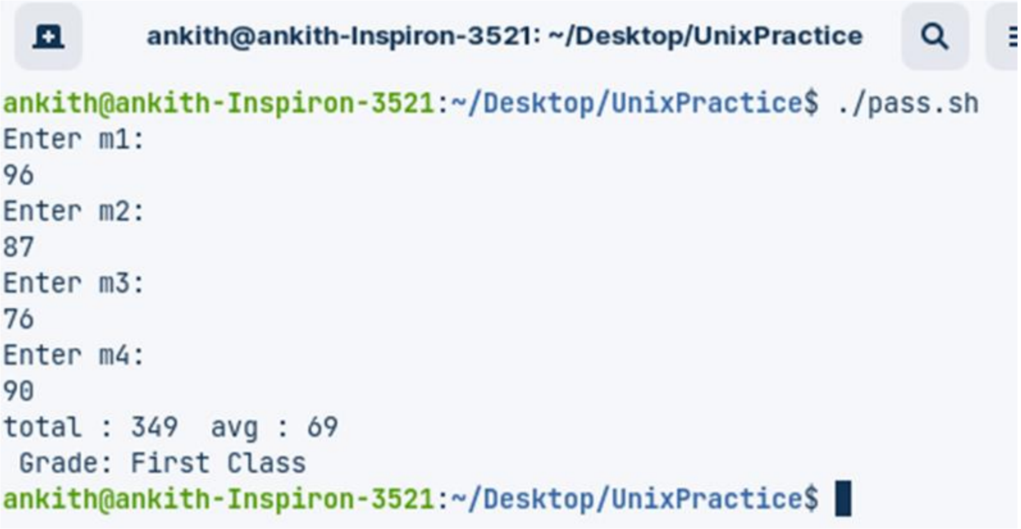
```
ankith@ankith-Inspiron-3521:~/Desktop/UnixPractice$ ./natsum.sh
Enter a number
20
Sum=210
ankith@ankith-Inspiron-3521:~/Desktop/UnixPractice$
```

Experiment No 13

Shell script to display the pass class of a student

```
#!/bin/bash
echo "Enter m1:\c and Enter m2:\c "
read m1
echo "Enter m3:\c"
read m3
echo "Enter m4:\c"
read m4
echo "Enter m5:\c"
read m5
tot=`expr $m1 + $m2 + $m3 + $m4 + $m5`;
avg=`expr $tot / 5`;
echo "total : $tot \n avg : $avg"
if [ $avg -gt 85 ];then
echo " Grade: Distinction "
elif [ $avg -gt 65 ];then
echo " Grade: First Class "
elif [ $avg -gt 50 ];then
echo " Grade: Second Class  "
elif [ $avg -gt 35 ];then
echo " Grade: Pass "
else echo " Grade: Fail"
fi
```

Output:



```
ankith@ankith-Inspiron-3521:~/Desktop/UnixPractice$ ./pass.sh
Enter m1:
96
Enter m2:
87
Enter m3:
76
Enter m4:
90
total : 349  avg : 69
 Grade: First Class
ankith@ankith-Inspiron-3521:~/Desktop/UnixPractice$ 
```

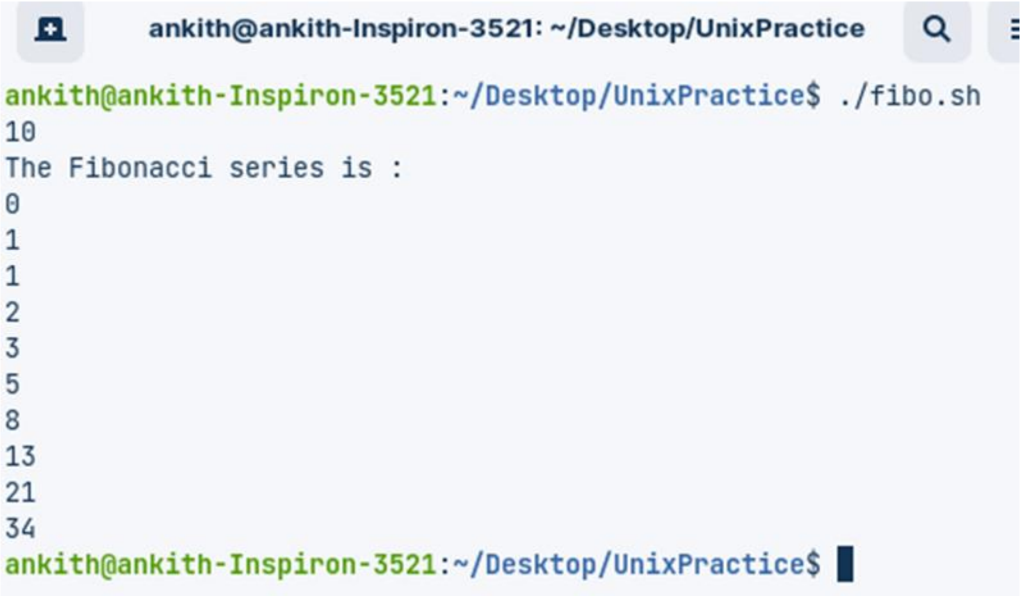
Experiment No 14

Shell script to find the Fibonacci series up to n

Program:

```
#!/bin/bash
read N
a=0
b=1
echo "The Fibonacci series is : "
for (( i=0; i<N; i++ ))
do
    echo "$a"
    fib=$((a + b))
    a=$b
    b=$fib
done
```

Output



```
ankith@ankith-Inspiron-3521:~/Desktop/UnixPractice$ ./fibo.sh
10
The Fibonacci series is :
0
1
1
2
3
5
8
13
21
34
ankith@ankith-Inspiron-3521:~/Desktop/UnixPractice$ 
```

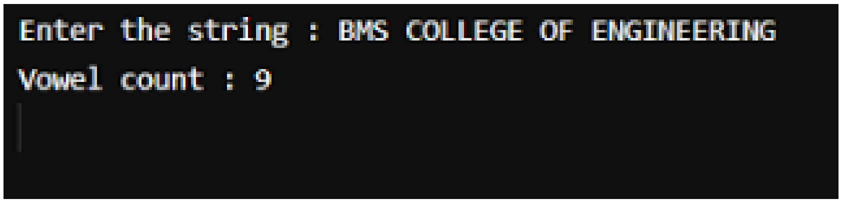
Experiment No 15

Shell script to count the number of vowels of a string

Program:

```
#!/bin/bash

echo "enter filename"

read filename

vowels=`cat $filename | tr -cd 'aeiouAEIOU' | wc -c`

echo "Number of vowels in $filename: $vowels"
```

Output



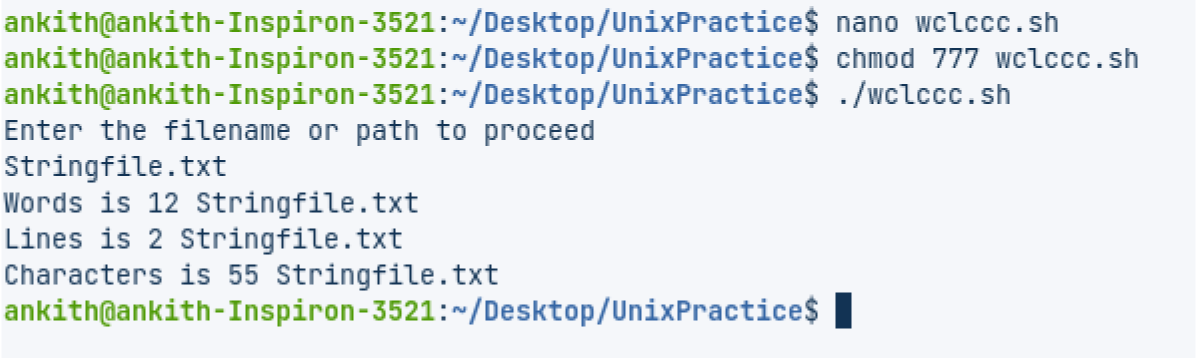
```
Enter the string : BMS COLLEGE OF ENGINEERING
Vowel count : 9
```

Experiment No 16

Shell script to check number of lines, words, characters in a file

```
#!/bin/bash
echo "Enter the filename or path to proceed"
read filename
words=`wc -w $filename`
lines=`wc -l $filename`
chars=`wc -c $filename`
echo "Words is $words"
echo "Lines is $lines"
echo "Characters is $chars"
```

Output

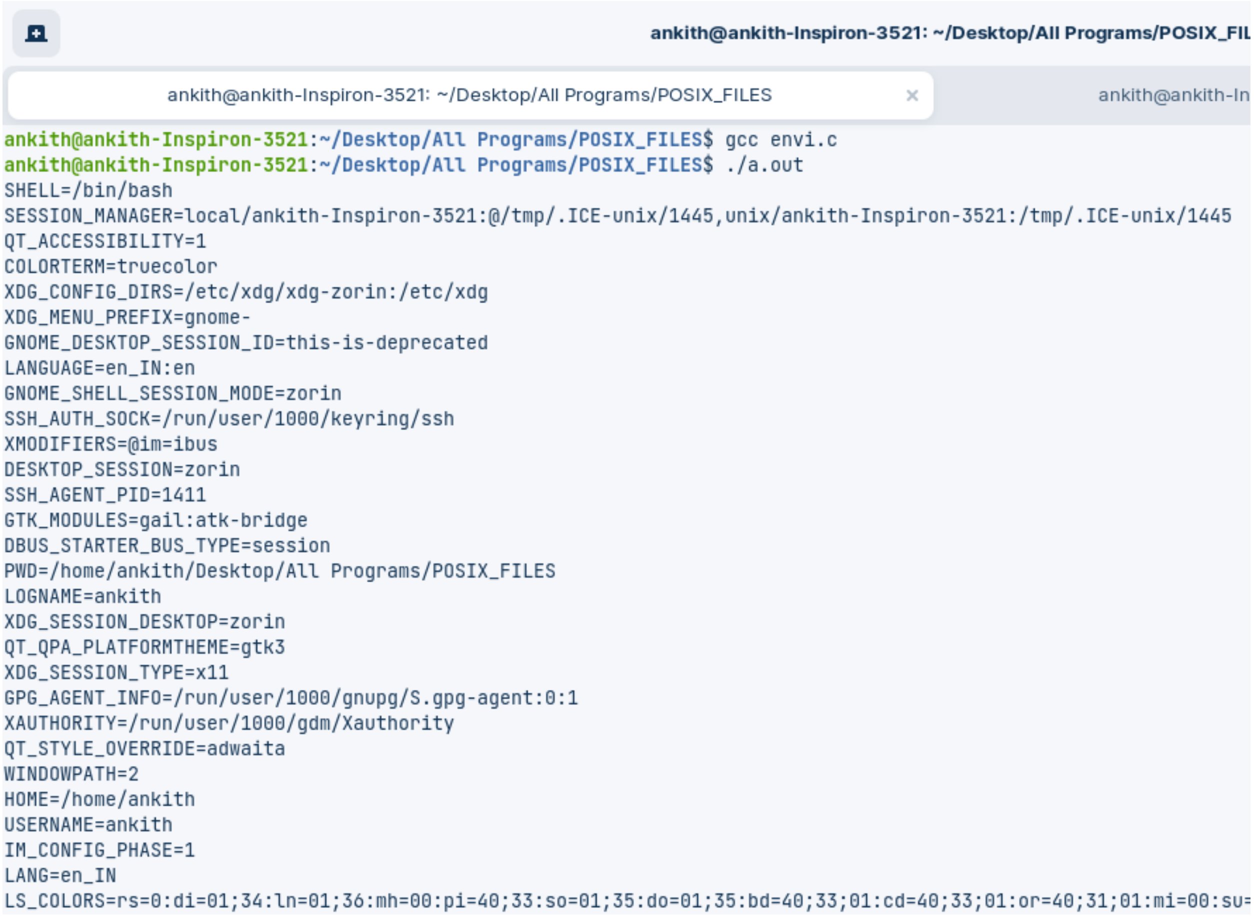


```
ankith@ankith-Inspiron-3521:~/Desktop/UnixPractice$ nano wclccc.sh
ankith@ankith-Inspiron-3521:~/Desktop/UnixPractice$ chmod 777 wclccc.sh
ankith@ankith-Inspiron-3521:~/Desktop/UnixPractice$ ./wclccc.sh
Enter the filename or path to proceed
Stringfile.txt
Words is 12 Stringfile.txt
Lines is 2 Stringfile.txt
Characters is 55 Stringfile.txt
ankith@ankith-Inspiron-3521:~/Desktop/UnixPractice$ 
```

Experiment No 17

Write a C/C++ program to that outputs the contents of its environment list

```
#include&lt;stdio.h&gt;

#include&lt;unistd.h&gt;

int main(int argc,char *argv[])

{

char **ptr;

extern char **environ;

for(ptr=environ; *ptr; ptr++)

printf(&quot;%s\n&quot;,*ptr);

return 0;

}
```


ankith@ankith-Inspiron-3521: ~/Desktop/All Programs/POSIX_FIL
ankith@ankith-Inspiron-3521: ~/Desktop/All Programs/POSIX_FILES
ankith@ankith-In

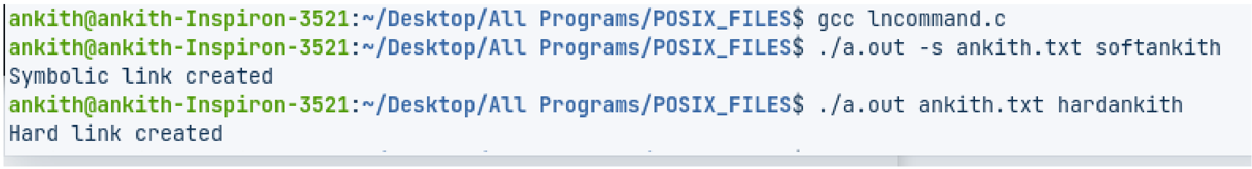
```
ankith@ankith-Inspiron-3521:~/Desktop/All Programs/POSIX_FILES$ gcc envi.c
ankith@ankith-Inspiron-3521:~/Desktop/All Programs/POSIX_FILES$ ./a.out
SHELL=/bin/bash
SESSION_MANAGER=local/ankith-Inspiron-3521:@/tmp/.ICE-unix/1445,unix/ankith-Inspiron-3521:/tmp/.ICE-unix/1445
QT_ACCESSIBILITY=1
COLORTERM=truecolor
XDG_CONFIG_DIRS=/etc/xdg/xdg-zorin:/etc/xdg
XDG_MENU_PREFIX=gnome-
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
LANGUAGE=en_IN:en
GNOME_SHELL_SESSION_MODE=zorin
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
XMODIFIERS=@im=ibus
DESKTOP_SESSION=zorin
SSH_AGENT_PID=1411
GTK_MODULES=gail:atk-bridge
DBUS_STARTER_BUS_TYPE=session
PWD=/home/ankith/Desktop/All Programs/POSIX_FILES
LOGNAME=ankith
XDG_SESSION_DESKTOP=zorin
QT_QPA_PLATFORMTHEME=gtk3
XDG_SESSION_TYPE=x11
GPG_AGENT_INFO=/run/user/1000/gnupg/S.gpg-agent:0:1
XAUTHORITY=/run/user/1000/gdm/Xauthority
QT_STYLE_OVERRIDE=adwaita
WINDOWPATH=2
HOME=/home/ankith
USERNAME=ankith
IM_CONFIG_PHASE=1
LANG=en_IN
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33;01:or=40;31;01:mi=00:su=
```

Experiment No 18

Write a C/C++ program to emulate the Unix ln command

```
#include<unistd.h>
#include<stdio.h>
#include<string.h>
int main(int argc , char * argv[]){
    if(argc<3 || argc>4){
        printf("Error in usage\n");
        return -1;
    }
    if(argc==4 && strcmp(argv[1],"-s")!=0){
        printf("for symbolic link use -s option");
        return -1;
    }
    if(argc==4 && access(argv[2] , F_OK)==-1){
        printf("Source file does not exist");
        return -1;
    }
    if(argc==3 && access(argv[1] , F_OK)==-1){
        printf("Source file does not exist");
        return -1;
    }
    if(argc==4){
        symlink(argv[2] , argv[3]);
        printf("Symbolic link is created");
        return 0;
    }
    if(argc==3){
        link(argv[1] , argv[2]);
        printf("Hard link is created");
        return 0;
    }
}
```

Output:



```
ankith@ankith-Inspiron-3521:~/Desktop/All Programs/POSIX_FILES$ gcc lncommand.c
ankith@ankith-Inspiron-3521:~/Desktop/All Programs/POSIX_FILES$ ./a.out -s ankith.txt softankith
Symbolic link created
ankith@ankith-Inspiron-3521:~/Desktop/All Programs/POSIX_FILES$ ./a.out ankith.txt hardankith
Hard link created
```

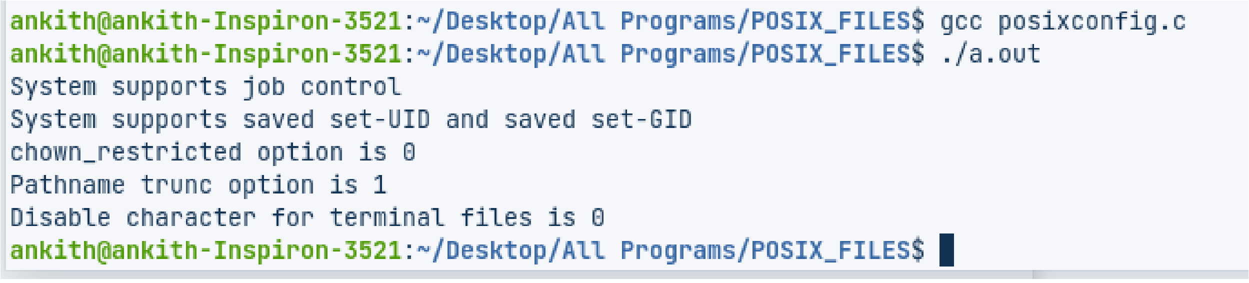
Experiment No 19

Write a C/C++ POSIX compliant program that prints the POSIX defined Configuration options supported on any given system using feature test macros.

```
#define _POSIX_SOURCE
#define _POSIX_C_SOURCE 199309L
#include&lt;iostream&gt;
#include&lt;unistd.h&gt;
int main()
{
using namespace std;
#ifdef _POSIX_JOB_CONTROL
cout&lt;&lt;&quot;System Supports Job Control feature&quot;&lt;&lt;endl;
#else
cout&lt;&lt;&quot;System doesnot support job control\n&quot;;
#endif
#ifdef _POSIX_SAVED_IDS
cout&lt;&lt;&quot;System Supports saved set-UID and saved set-GID&quot;&lt;&lt;endl;
#else
cout&lt;&lt;&quot;System doesnot support saved set-UID\n&quot;;
#endif
#ifdef _POSIX_CHOWN_RESTRICTED
cout&lt;&lt;&quot;System Supports Change Ownership feature:&quot;&lt;&lt;endl;
#else
cout&lt;&lt;&quot;System doesnot support change Ownership feature\n&quot;;
#endif
#ifdef _POSIX_NO_TRUNC
cout&lt;&lt;&quot;System Supports Path truncation option:&quot;&lt;&lt;endl;
```

```
#else
cout&lt;&lt;&quot;System doesnot support Path truncation \n&quot;;
#endif
#ifdef _POSIX_VDISABLE
cout&lt;&lt;&quot;System Supports Disable Character for files:&quot;&lt;&lt;endl;
#else
cout&lt;&lt;&quot;System doesnot support Disable Characters \n&quot;;
#endif
return 0;
```

Output:



```
ankith@ankith-Inspiron-3521:~/Desktop/All Programs/POSIX_FILES$ gcc posixconfig.c
ankith@ankith-Inspiron-3521:~/Desktop/All Programs/POSIX_FILES$ ./a.out
System supports job control
System supports saved set-UID and saved set-GID
chown_restricted option is 0
Pathname trunc option is 1
Disable character for terminal files is 0
ankith@ankith-Inspiron-3521:~/Desktop/All Programs/POSIX_FILES$
```

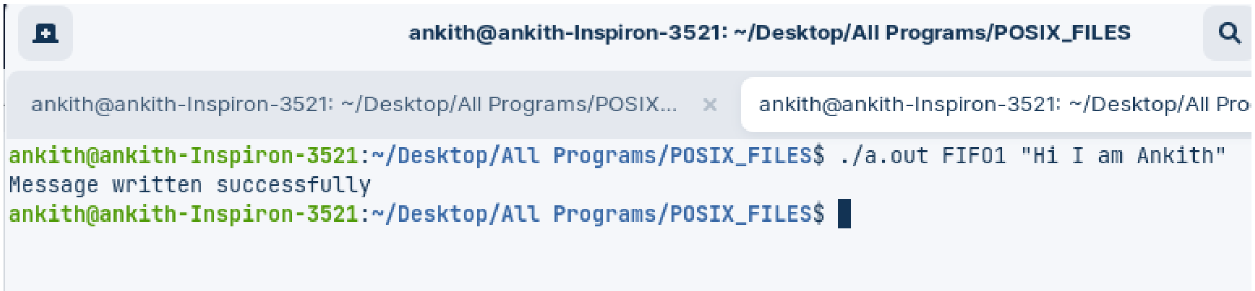
Experiment No 20

Write a C/C++ program which demonstrates Interprocess Communication between a reader process and a writer process. Use mkfifo, open, read, write and close apis in your program.

```
#include <sys/stat.h>
        #include <string.h>
        #include <fcntl.h>
        #include <stdio.h>
        #include <unistd.h>
        int main(int argc, char *argv[])
        {
        char buf[100];
        int fd,n;
        mkfifo (argv[1], S_IFIFO |0777);
        if (argc == 3){
        fd =  open (argv[1], O_WRONLY);
        write (fd, argv[2], strlen(argv[2]));
        close(fd);}
        if (argc ==2){
        fd = open (argv[1], O_RDONLY);
        n= read (fd, buf, sizeof(buf));
        buf[n]='\0';
        printf ("%s", buf);
        close(fd);
        }
```

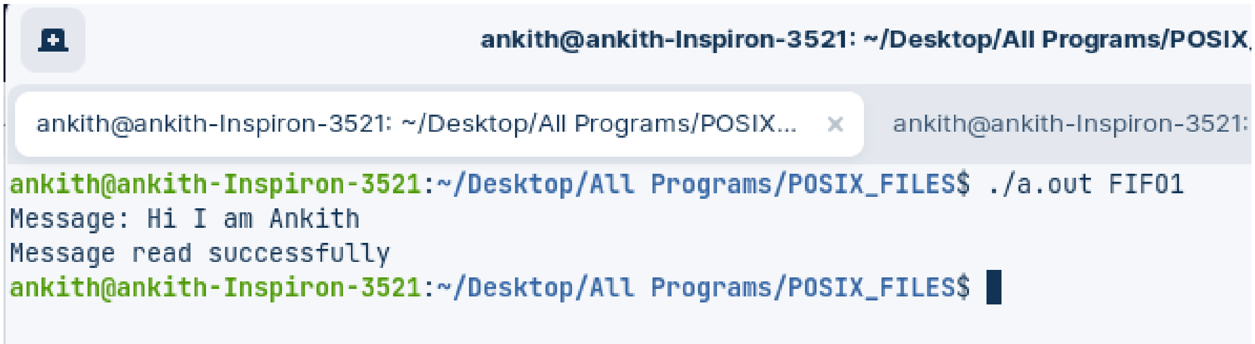
OUTPUT:

Writer process:


ankith@ankith-Inspiron-3521: ~/Desktop/All Programs/POSIX_FILES
ankith@ankith-Inspiron-3521: ~/Desktop/All Programs/POSIX...
ankith@ankith-Inspiron-3521: ~/Desktop/All Pro

```
ankith@ankith-Inspiron-3521:~/Desktop/All Programs/POSIX_FILES$ ./a.out FIFO1 "Hi I am Ankith"
Message written successfully
ankith@ankith-Inspiron-3521:~/Desktop/All Programs/POSIX_FILES$
```

Reader process:


ankith@ankith-Inspiron-3521: ~/Desktop/All Programs/POSIX
ankith@ankith-Inspiron-3521: ~/Desktop/All Programs/POSIX...
ankith@ankith-Inspiron-3521:

```
ankith@ankith-Inspiron-3521:~/Desktop/All Programs/POSIX_FILES$ ./a.out FIFO1
Message: Hi I am Ankith
Message read successfully
ankith@ankith-Inspiron-3521:~/Desktop/All Programs/POSIX_FILES$
```