VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.

LAB REPORT
on

UNIX SHELL AND SYSTEM PROGRAMMING

Submitted by

ARNAV SHARMA (1B20CS023)

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 "**Unix Shell and System Programming**" carried out by **Arnav Sharma (1BM20CS023),** 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 System 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

Experiment 1

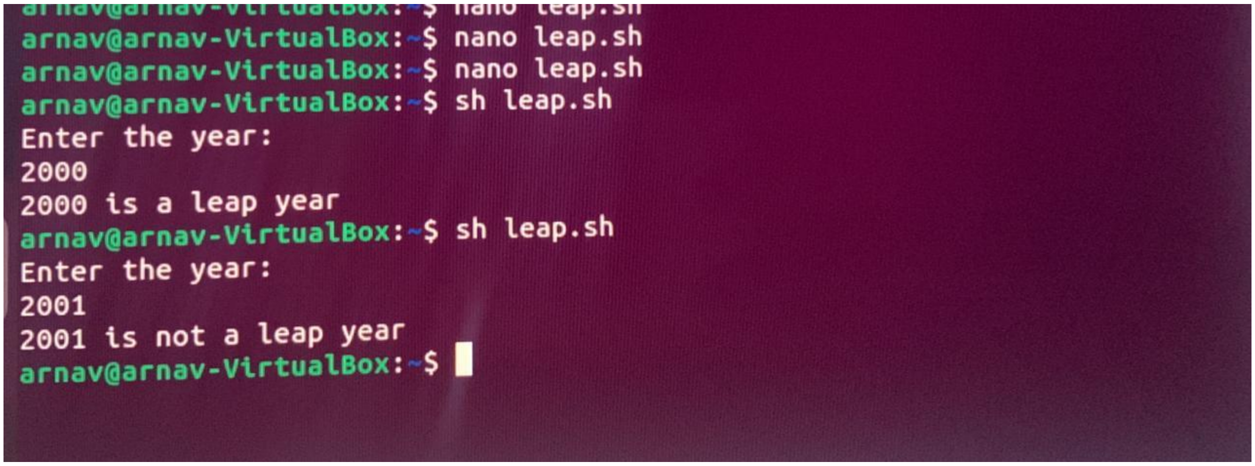
Aim of the program

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

Program

```
#!/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



```
arnav@arnav-VirtualBox:~$ nano leap.sh
arnav@arnav-VirtualBox:~$ nano leap.sh
arnav@arnav-VirtualBox:~$ nano leap.sh
arnav@arnav-VirtualBox:~$ sh leap.sh
Enter the year:
2000
2000 is a leap year
arnav@arnav-VirtualBox:~$ sh leap.sh
Enter the year:
2001
2001 is not a leap year
arnav@arnav-VirtualBox:~$ 
```

Experiment 2

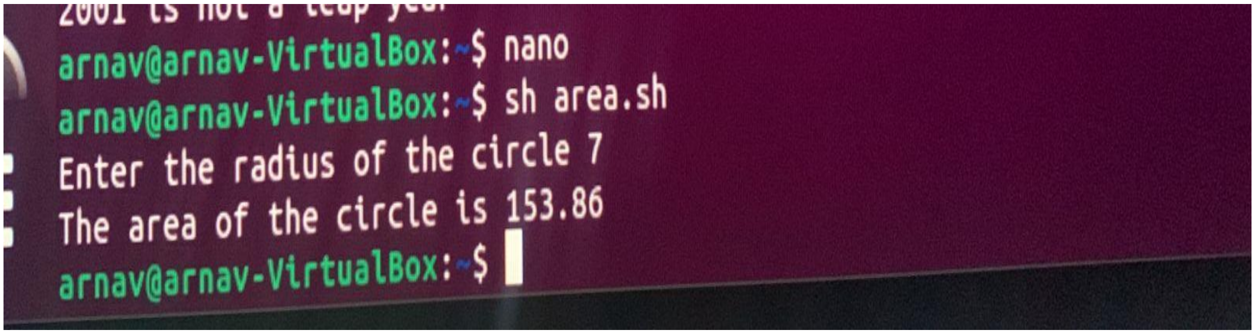
Aim of the program

Shell script to find the area of a circle

Program

```
#!/bin/bash


echo "\nEnter the radius of a circle : "
read r
echo "The area of the circle is : "
echo “3,14*$radius*$radius”|bc
```

Output



```
2001 is not a leap year
arnav@arnav-VirtualBox:~$ nano
arnav@arnav-VirtualBox:~$ sh area.sh
Enter the radius of the circle 7
The area of the circle is 153.86
arnav@arnav-VirtualBox:~$
```

Experiment 3

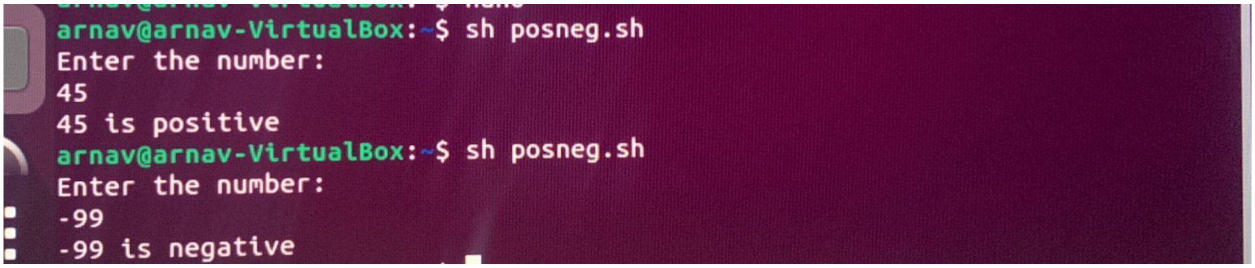
Aim of the program

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

Program

```
#!/bin/bash
echo "Enter first 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



```
arnav@arnav-VirtualBox:~$ nano
arnav@arnav-VirtualBox:~$ sh posneg.sh
Enter the number:
45
45 is positive
arnav@arnav-VirtualBox:~$ sh posneg.sh
Enter the number:
-99
-99 is negative
```

Experiment 4

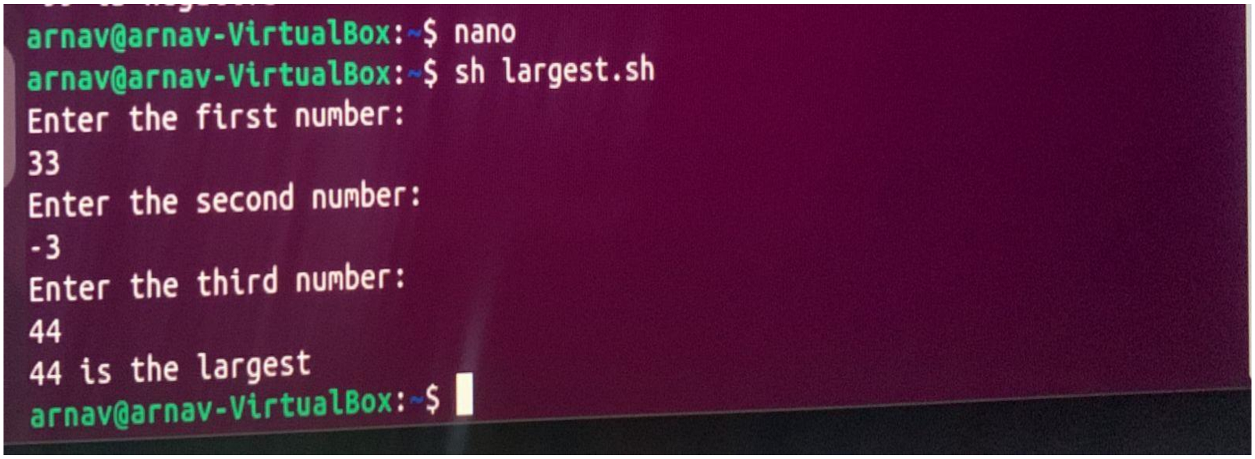
Aim of the program

Shell script to find the biggest of three numbers

Program

```
#!/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



```
arnav@arnav-VirtualBox:~$ nano
arnav@arnav-VirtualBox:~$ sh largest.sh
Enter the first number:
33
Enter the second number:
-3
Enter the third number:
44
44 is the largest
arnav@arnav-VirtualBox:~$ 
```

Experiment 5

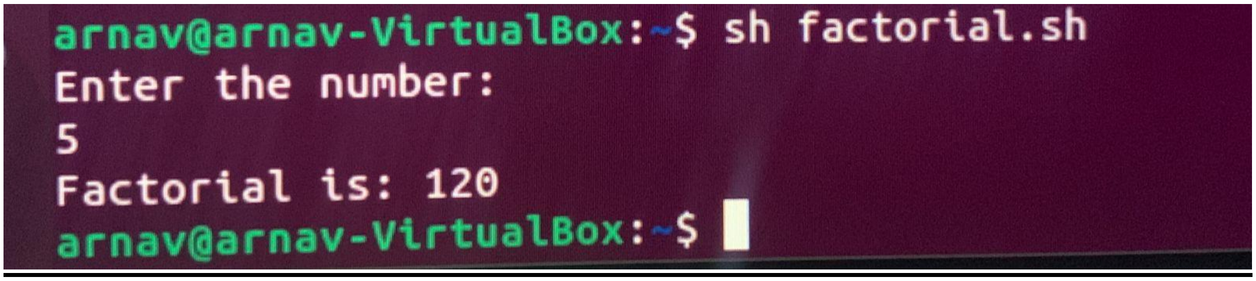
Aim of the program

Shell script to find the factorial of a number

Program

```
#!/bin/bash
read -p "Enter a number : " num
fact=1

for((i=2;i<=num;i++))
{
  fact=$((fact*i))
}

echo $Factorial is: fact
```

Output



```
arnav@arnav-VirtualBox:~$ sh factorial.sh
Enter the number:
5
Factorial is: 120
arnav@arnav-VirtualBox:~$ 
```

Experiment 6

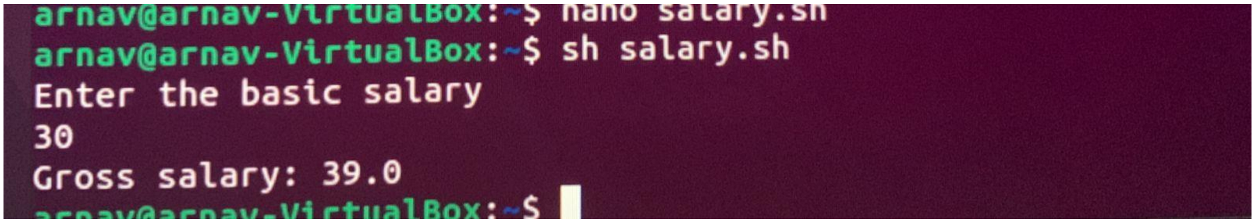
Aim of the program

Shell script to compute the gross salary of an employee

Program

```
#!/bin/bash
echo "\nEnter name of Employee :"
read name

echo "\nEnter DA :"
read da

echo "\nEnter HRA:"
read hra
echo "\nEnter basic"
read basic
sal=$(( $da + $hra + $basic ))
echo "Gross Salary: $sal"
```

Output



```
arnav@arnav-VirtualBox:~$ nano salary.sh
arnav@arnav-VirtualBox:~$ sh salary.sh
Enter the basic salary
30
Gross salary: 39.0
arnav@arnav-VirtualBox:~$
```

Experiment 7

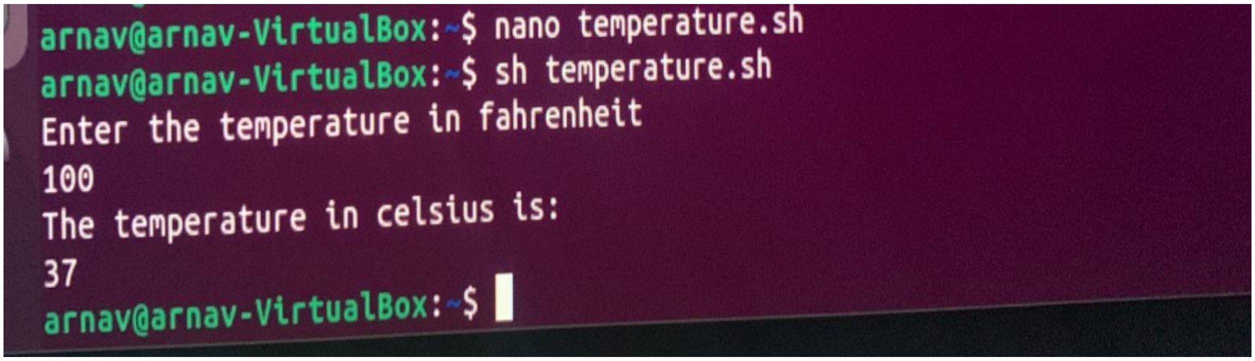
Aim of the program

Shell script to convert the temperature Fahrenheit to Celsius

Program

```
#!/bin/bash
echo "Enter temperature in fahrenheit : "
read f
echo “The temperature in celsius is: “
# formula c=(5/9)*(f-32)
c=$(echo "scale=2;(5/9)*($f-32)"|bc)
echo "$f °F = $c °C"
```

Output



```
arnav@arnav-VirtualBox:~$ nano temperature.sh
arnav@arnav-VirtualBox:~$ sh temperature.sh
Enter the temperature in fahrenheit
100
The temperature in celsius is:
37
arnav@arnav-VirtualBox:~$ 
```

Experiment 8

Aim of the program

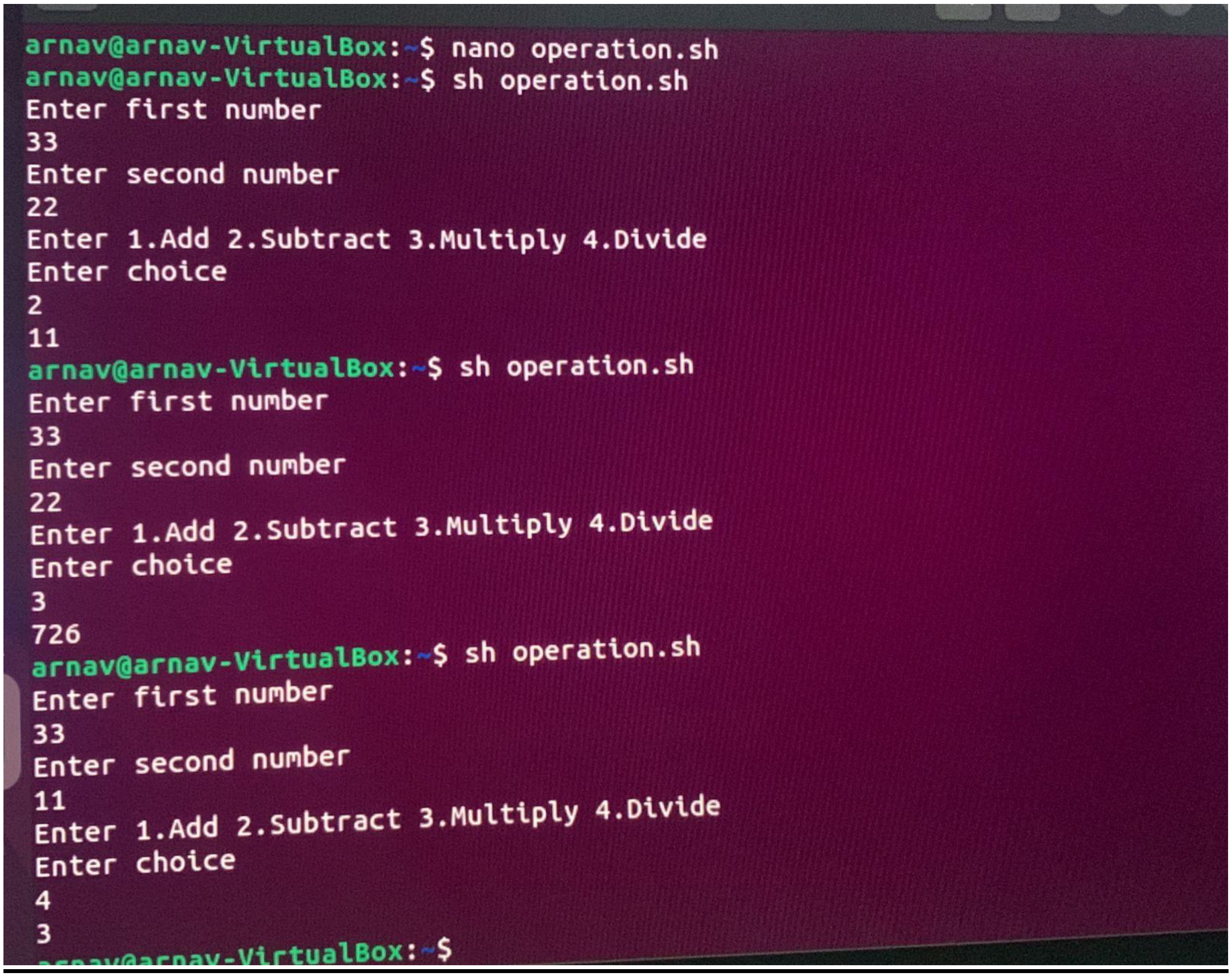
Shell script to perform arithmetic operations on given two numbers

Program

```
#!/bin/bash
echo "Enter 2 Numbers : "
read a
read b

echo "Enter Operation : \n"
echo "1) Addition"
echo "2) Subtraction"
echo "3) Multiplication"
echo "4) Division(Quotient)"
echo "5) Modulus(Remainder)\n"
read op

case $op in
  1)echo "scale=3; $a + $b" | bc -l;;
  2)echo "scale=3; $a - $b" | bc -l  ;;
  3)echo "scale=3; $a \* $b" | bc -l;;
  4)echo "scale=3; $a / $b" | bc -l ;;
  5)echo "scale=3; $a % $b" | bc -l;;
  *)echo "Choose a valid option"
```

esac

Output



```
arnav@arnav-VirtualBox:~$ nano operation.sh
arnav@arnav-VirtualBox:~$ sh operation.sh
Enter first number
33
Enter second number
22
Enter 1.Add 2.Subtract 3.Multiply 4.Divide
Enter choice
2
11
arnav@arnav-VirtualBox:~$ sh operation.sh
Enter first number
33
Enter second number
22
Enter 1.Add 2.Subtract 3.Multiply 4.Divide
Enter choice
3
726
arnav@arnav-VirtualBox:~$ sh operation.sh
Enter first number
33
Enter second number
11
Enter 1.Add 2.Subtract 3.Multiply 4.Divide
Enter choice
4
3
arnav@arnav-VirtualBox:~$
```

Experiment 9

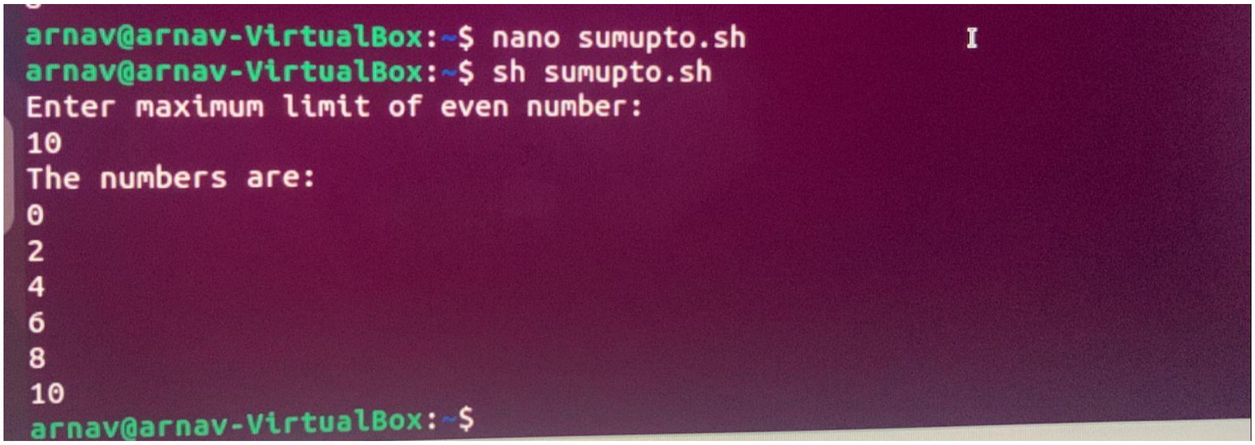
Aim of the program

Shell script to find the sum of even numbers up to 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
```

Output



```
arnav@arnav-VirtualBox:~$ nano sumupto.sh
arnav@arnav-VirtualBox:~$ sh sumupto.sh
Enter maximum limit of even number:
10
The numbers are:
0
2
4
6
8
10
arnav@arnav-VirtualBox:~$
```

Experiment 10

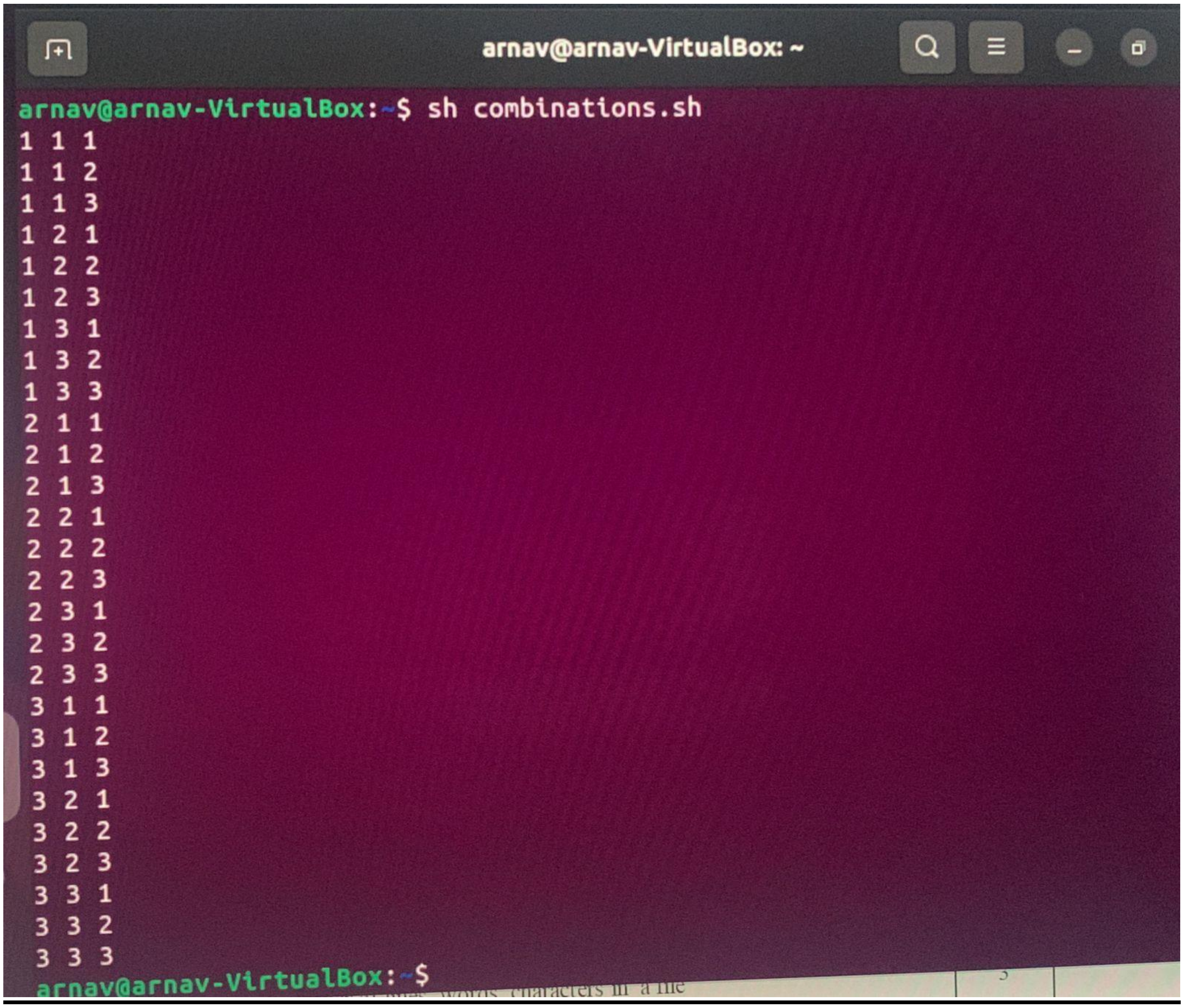
Aim of the program

Shell script to print the combinations of numbers 123

Program

```
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



```
arnav@arnav-VirtualBox:~$ sh combinations.sh
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
arnav@arnav-VirtualBox:~$
```

Experiment 11

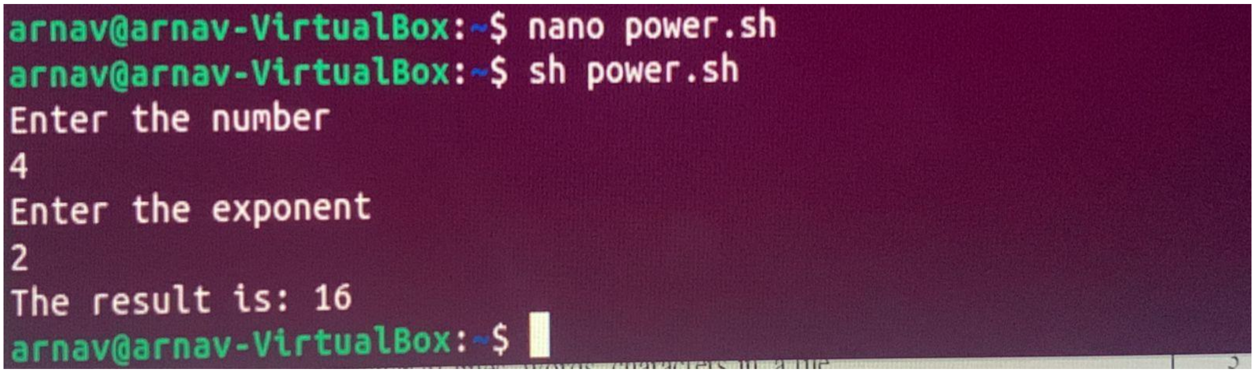
Aim of the program

Shell script to find the power of a number

Program

```
#!/bin/bash
read -p "Enter the number : " a
read -p "Enter the exponent : " b

res=1
for ((i = 1; i <= b; i++)); do
    res=$(expr $res \* $a)
done
echo $res
```

Output



```
arnav@arnav-VirtualBox:~$ nano power.sh
arnav@arnav-VirtualBox:~$ sh power.sh
Enter the number
4
Enter the exponent
2
The result is: 16
arnav@arnav-VirtualBox:~$
```

Experiment 12

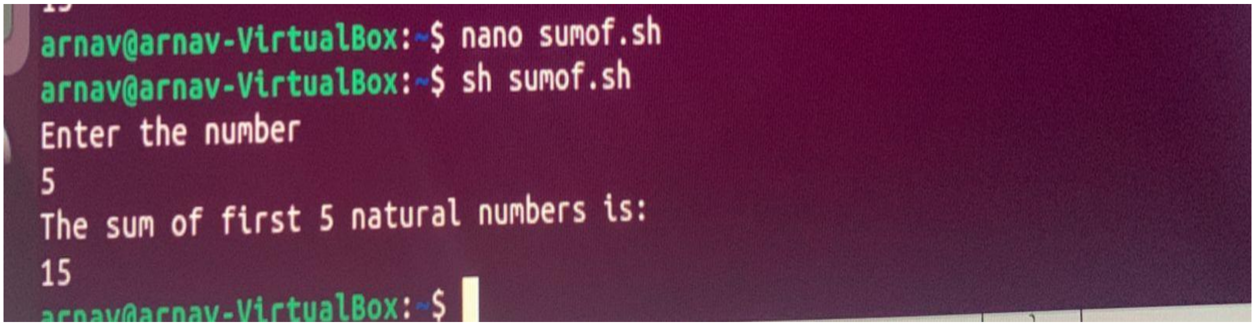
Aim of the program

Shell script to find the sum of n natural numbers

Program

```
#!/bin/bash
echo "Enter the number"
read x
y=0
while [$x -gt 0 ]
do
y = `expr $y + $x`
x = `expr $x - 1`
echo "The sum of first $x natural numbers  is: $y"
```

Output



```
arnav@arnav-VirtualBox:~$ nano sumof.sh
arnav@arnav-VirtualBox:~$ sh sumof.sh
Enter the number
5
The sum of first 5 natural numbers is:
15
arnav@arnav-VirtualBox:~$
```

Experiment 13

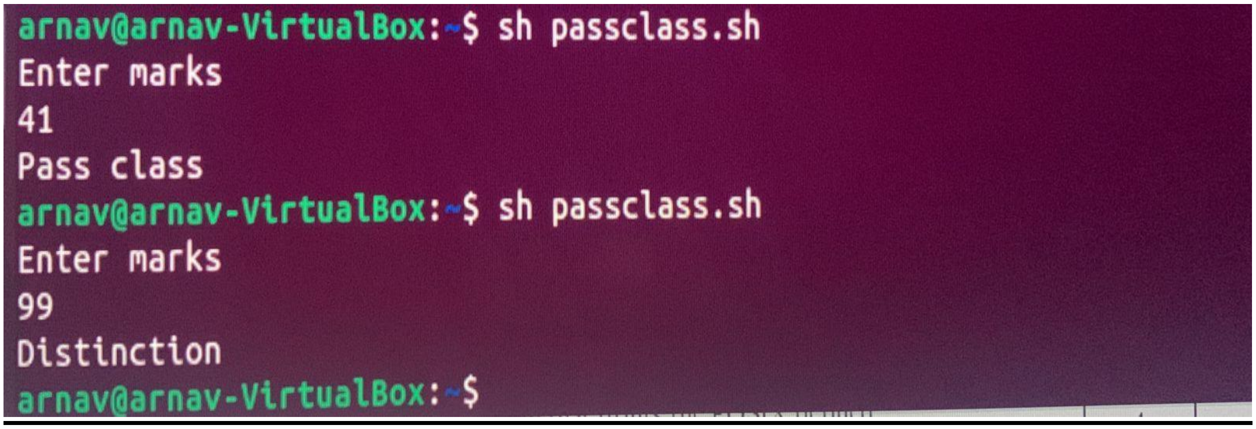
Aim of the program

Shell script to display the pass class of a student

Program

```
read -p "Enter marks : " marks
if [ $marks-ge 90 ]
then
echo "S Grade"
elif [ $per -ge 75 ]
then
echo "A Grade"
elif [ $per -ge  60 ]
then
echo "B Grade"
elif [ $per -le  40 ]
then
echo "Fail"
else
echo "C Grade"
fi
```

Output



```
arnav@arnav-VirtualBox:~$ sh passclass.sh
Enter marks
41
Pass class
arnav@arnav-VirtualBox:~$ sh passclass.sh
Enter marks
99
Distinction
arnav@arnav-VirtualBox:~$
```

Experiment 14

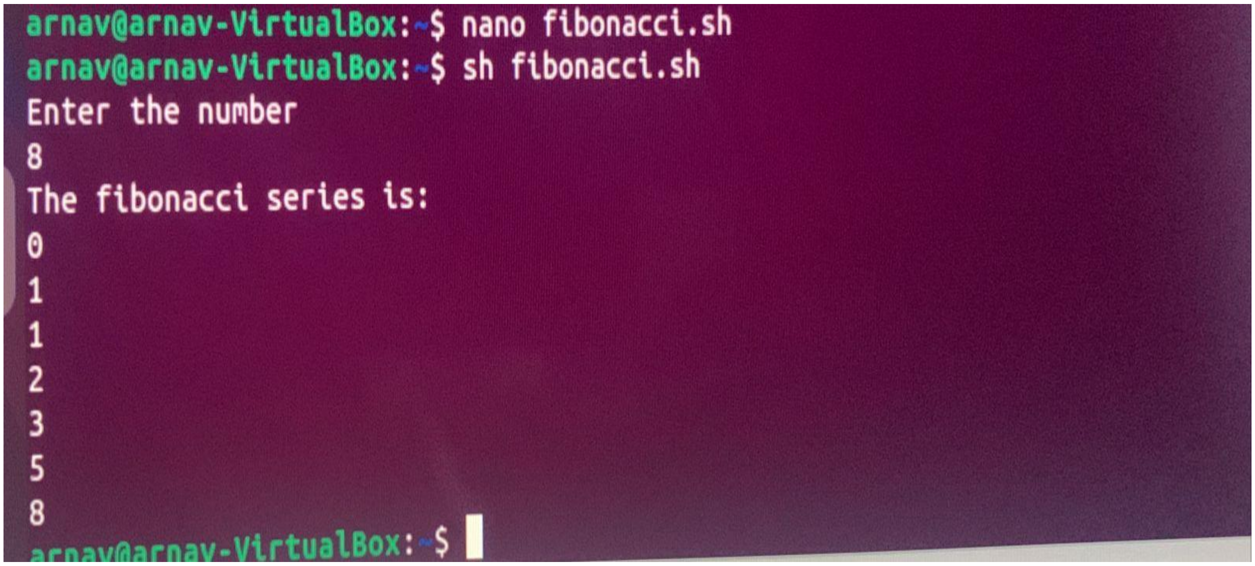
Aim of the program

Shell script to find the Fibonacci series up to n

Program

```
read -p "Enter a number : " 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



```
arnav@arnav-VirtualBox:~$ nano fibonacci.sh
arnav@arnav-VirtualBox:~$ sh fibonacci.sh
Enter the number
8
The fibonacci series is:
0
1
1
2
3
5
8
arnav@arnav-VirtualBox:~$
```

Experiment 15

Aim of the program

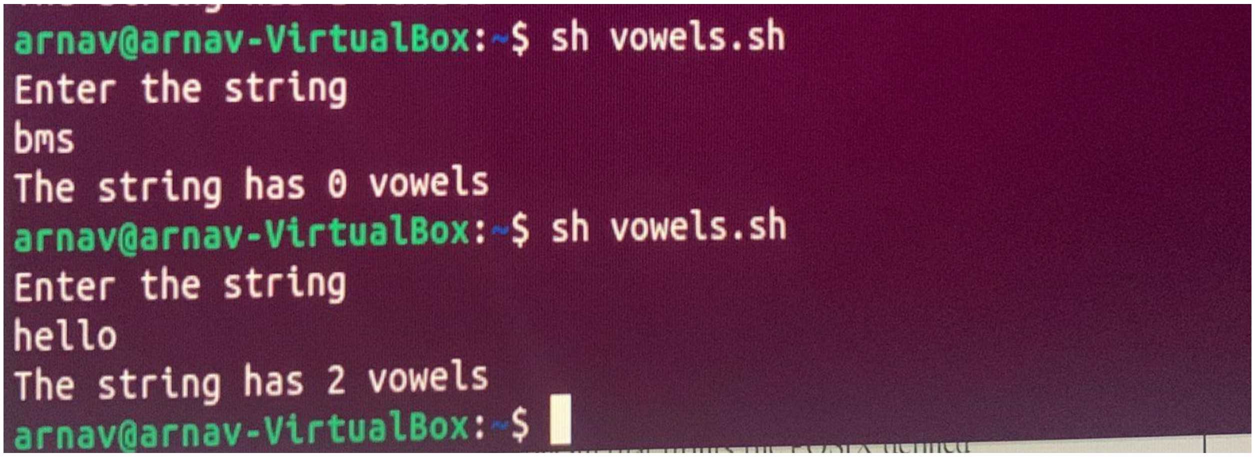
Shell script to count the number of vowels of a string

Program

```
#!/bin/bash
echo "Enter string"
read str
vowel = 0
i=`expr length $str`

while [ $i -gt 0 ]
do
temp = `expr $str|cut -c $i`

case $temp in
a|A) vowel = `expr $vowel + 1`;;
e|E) vowel = `expr $vowel + 1`;;
i|I) vowel = `expr $vowel + 1`;;
o|O) vowel = `expr $vowel + 1`;;
u|U) vowel = `expr $vowel + 1`;;
esac

i = `expr $i - 1`
done
```

```
echo "Number of vowels in string $vowel"
```

Output



```
arnav@arnav-VirtualBox:~$ sh vowels.sh
Enter the string
bms
The string has 0 vowels
arnav@arnav-VirtualBox:~$ sh vowels.sh
Enter the string
hello
The string has 2 vowels
arnav@arnav-VirtualBox:~$ 
```

Experiment 16

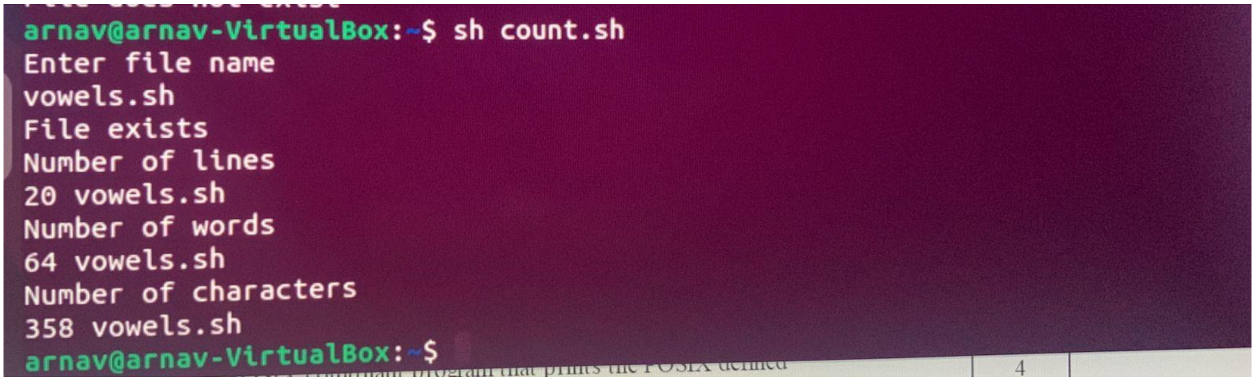
Aim of the program

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

Program

```
#!/bin/bash
echo "Enter file name"
read fname
echo "Number of lines in file"
wc -l $fname
echo "Number of words in file"
wc -w $fname
echo "Number of charcaters in file"
wc -m $fname
else
echo "Non existent file"
fi
```

Output



```
arnav@arnav-VirtualBox:~$ sh count.sh
Enter file name
vowels.sh
File exists
Number of lines
20 vowels.sh
Number of words
64 vowels.sh
Number of characters
358 vowels.sh
arnav@arnav-VirtualBox:~$
```

Experiment 17

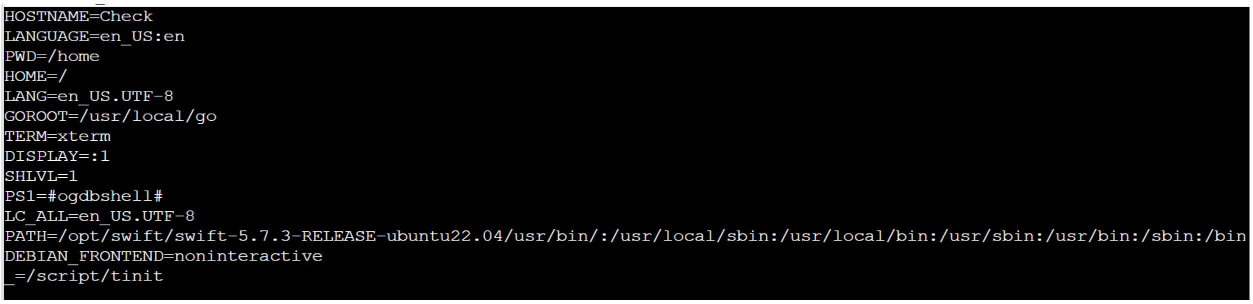
Aim of the program

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

Program

```
#include<stdio.h>
int main(int
argc, char* argv[ ])
{
int i;
char **ptr;
extern char
**environ;
for( ptr =
environ; *ptr != 0; ptr++ ) /*echo all env strings*/
printf(“%s\n”,
*ptr);
return 0;
}
```

Output



```
HOSTNAME=Check
LANGUAGE=en_US:en
PWD=/home
HOME=/
LANG=en_US.UTF-8
GOROOT=/usr/local/go
TERM=xterm
DISPLAY=:1
SHLVL=1
PS1=#ogdbshell#
LC_ALL=en_US.UTF-8
PATH=/opt/swift/swift-5.7.3-RELEASE-ubuntu22.04/usr/bin/:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
DEBIAN_FRONTEND=noninteractive
_=/script/tinit
```

Experiment 18

Aim of the program

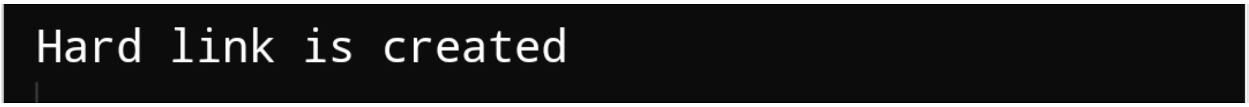
Write a C/C++ program to emulate the Unix **ln** command

Program

```
#include<stdio.h>
#include<sys/types.h>
#include<unistd.h>
#include<string.h>
int main(int argc, char * argv[])
{
if(argc < 3 || argc > 4 || (argc == 4 && strcmp(argv[1],”-s”)))
{
printf(“Usage: ./a.out [-s] <org_file> <new_link>\n”);
return 1;
}
if(argc == 4)
{
if((symlink(argv[2],argv[3])) == -1)
printf(“Cannot create symbolic link\n”) ;
else
printf(“Symbolic link created\n”) ;
}
else
{
```

```
if((link(argv[1],argv[2])) == -1)
printf(“Cannot create hard link\n”) ;
else
printf(“Hard link created\n”) ;
}
return 0;
}
```

<u>Output</u>



```
Hard link is created
```

Experiment 19

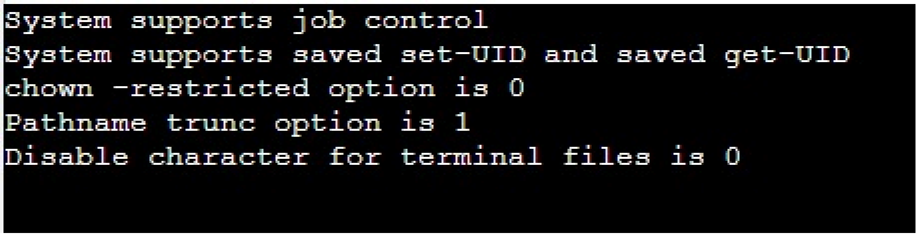
Aim of the program

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

Program

```
#define _POSIX_SOURCE
#define _POSIX_C_SOURCE 199309L
#include<stdio.h>
#include<unistd.h>
int main()
{
#ifdef _POSIX_JOB_CONTROL
printf(“System supports job control\n”);
#else
printf(“System does not support job control \n”);
#endif
#ifdef _POSIX_SAVED_IDS
printf(“System supports saved set-UID and saved set-GID\n”);
#else
printf(“System does not support saved set-UID and saved set-GID \n”);
#endif
#ifdef _POSIX_CHOWN_RESTRICTED
printf(“chown_restricted option is %d\n”,_POSIX_CHOWN_RESTRICTED);
#else
printf("System does not support chown_restricted option \n");
#endif
#ifdef _POSIX_NO_TRUNC
printf("Pathname trunc option is %d\n",_POSIX_NO_TRUNC);
#else
printf("System does not support system-wide pathname trunc option \n");
#endif
#ifdef _POSIX_VDISABLE
printf("Disable character for terminal files is %d\n",_POSIX_VDISABLE);
#else
printf("System does not support _POSIX_VDISABLE \n");
#endif
return 0;
}
```

Output



```
System supports job control
System supports saved set-UID and saved get-UID
chown -restricted option is 0
Pathname trunc option is 1
Disable character for terminal files is 0
```

Experiment 20

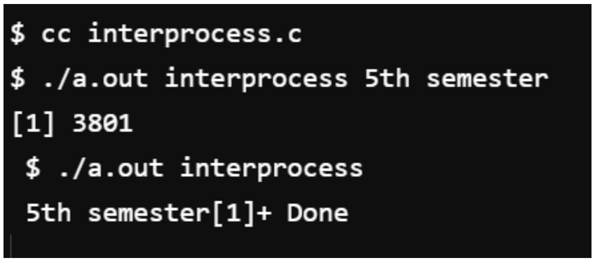
Aim of the program

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.

Program

```
#include<sys/types.h>
#include<unistd.h>
#include<fcntl.h>
#include<sys/stat.h>
#include<string.h>
#include<errno.h>
#include<stdio.h>
int main(int argc, char* argv[])
{
int fd;
char buf[256];
if(argc != 2 && argc != 3)
{
printf(“USAGE %s <file> [<arg>]\n”,argv[0]);
return 0;
}
mkfifo(argv[1],S_IFIFO | S_IRWXU | S_IRWXG | S_IRWXO );
if(argc == 2)
{
fd = open(argv[1], O_RDONLY|O_NONBLOCK);
while(read(fd,buf, sizeof(buf)) &gt; 0)
printf(“%s”,buf);
}
else
{
fd = open(argv[1], O_WRONLY);
write(fd,argv[2],strlen(argv[2]));
}
close(fd);
}
```

Output



```
$ cc interprocess.c
$ ./a.out interprocess 5th semester
[1] 3801
 $ ./a.out interprocess
 5th semester[1]+ Done
```