VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.

LAB REPORT
on

UNIX SHELL AND PROGRAMMING

Submitted by

Aravind Siddharth R(1BM20CS021)

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 **Aravind Siddharth R(1BM20CS021),** 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. Kayarvizhi N
Associate Professor
Department of CSE
BMSCE, Bengaluru

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

`

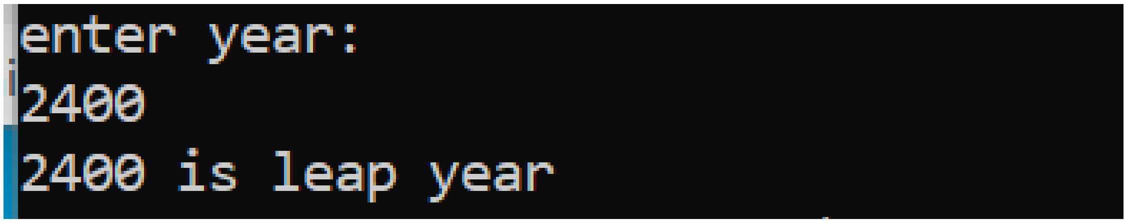
Index

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



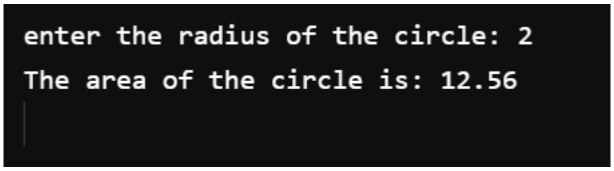
```
enter year:
2400
2400 is leap year
```

Experiment No 2

Shell script to find the area of a circle

```
echo "CIRCLE AREA & CIRCUMFERENCE"

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



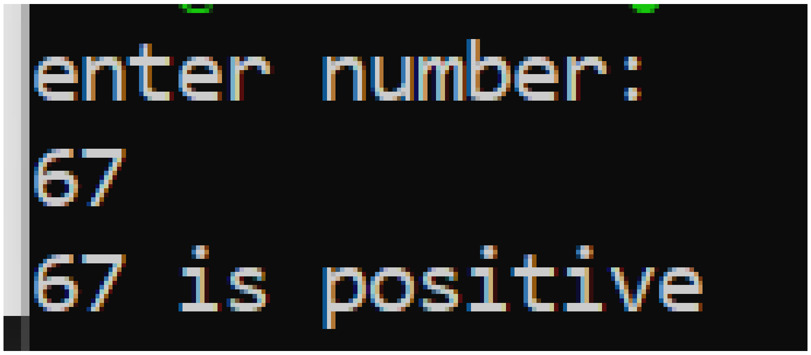
```
enter the radius of the circle: 2
The area of the circle is: 12.56
```

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



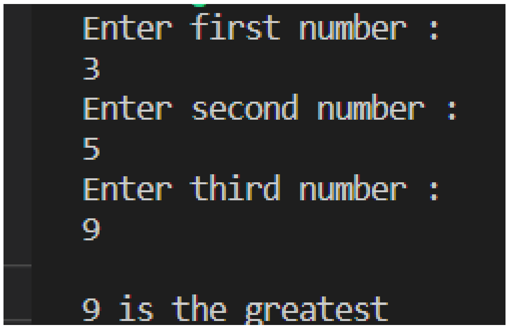
```
enter number:
67
67 is positive
```

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



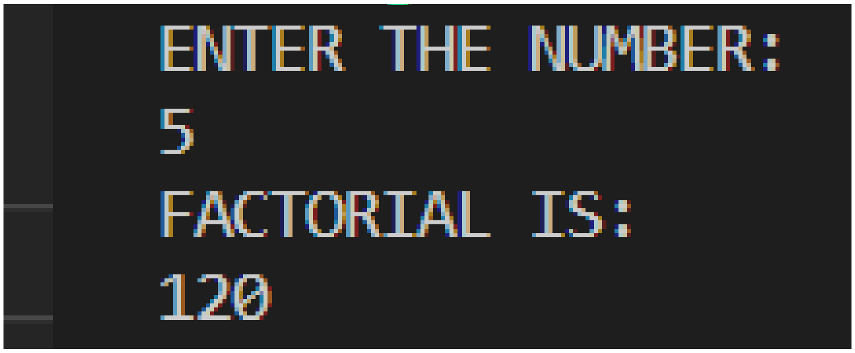
```
Enter first number :
3
Enter second number :
5
Enter third number :
9

9 is the greatest
```

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



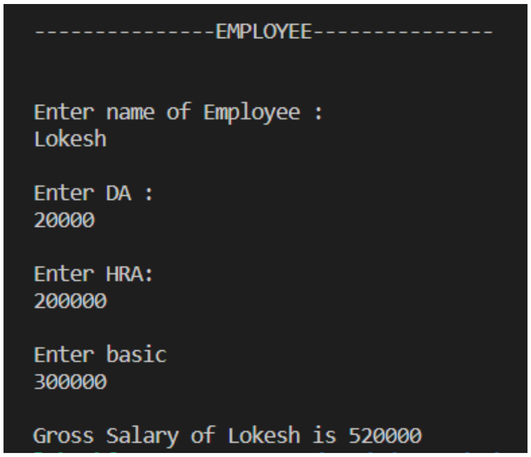
```
ENTER THE NUMBER:
5
FACTORIAL IS:
120
```

Experiment No 6

Shell script to compute the gross salary of an employee

```
#!/bin/bash

echo "\n---------------EMPLOYEE---------------\n"

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 "\nGross Salary of $name is $sal"
```

Output



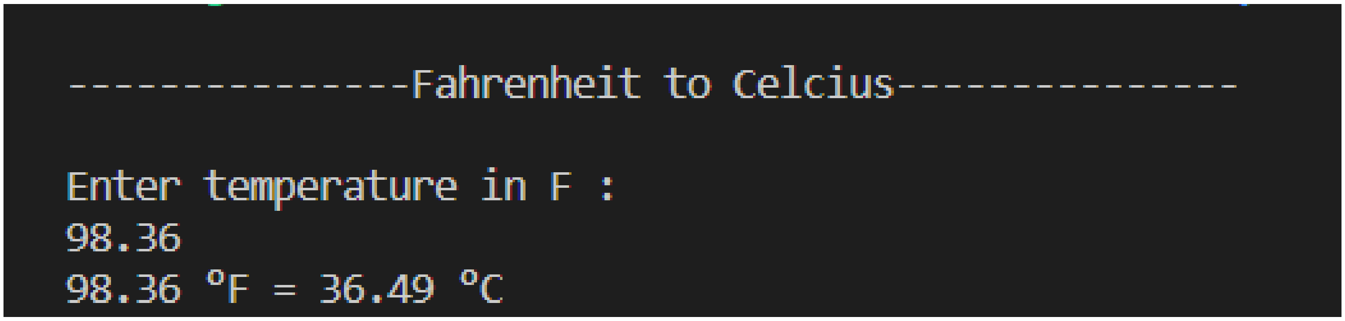
```
---------------EMPLOYEE---------------


Enter name of Employee :
Lokesh

Enter DA :
20000

Enter HRA:
200000

Enter basic
300000

Gross Salary of Lokesh is 520000
```

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:



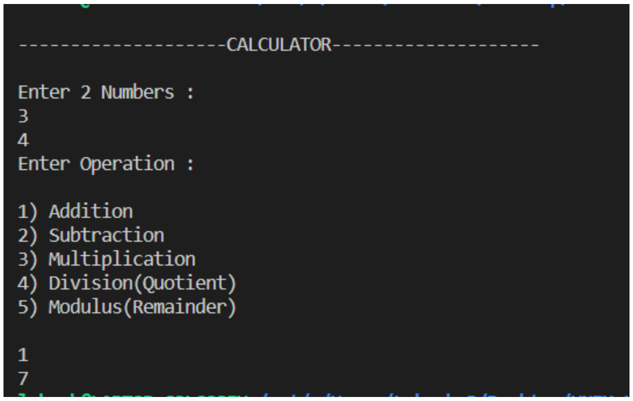
```
---------------Fahrenheit to Celcius---------------

Enter temperature in F :
98.36
98.36 °F = 36.49 °C
```

Experiment No 8

Shell script to perform arithmetic operations on given two numbers

```
#!/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



```
--------------------CALCULATOR-------------------

Enter 2 Numbers :
3
4
Enter Operation :

1) Addition
2) Subtraction
3) Multiplication
4) Division(Quotient)
5) Modulus(Remainder)

1
7
```

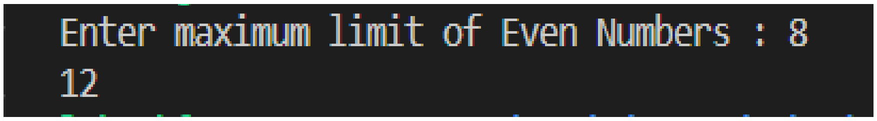
Experiment No 9

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



```
Enter maximum limit of Even Numbers : 8
12
```

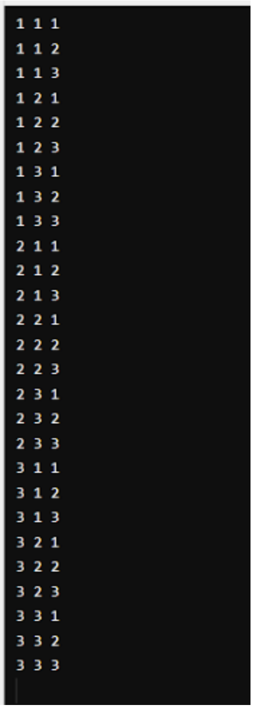
Experiment No 10

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



```
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
```

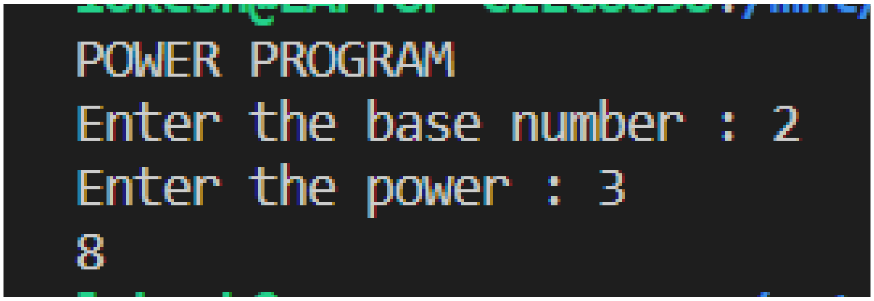
Experiment No 11

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



```
POWER PROGRAM
Enter the base number : 2
Enter the power : 3
8
```

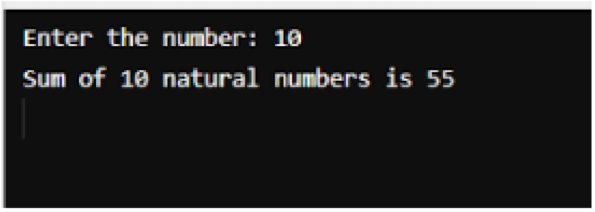
Experiment No 12

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



```
Enter the number: 10
Sum of 10 natural numbers is 55
```

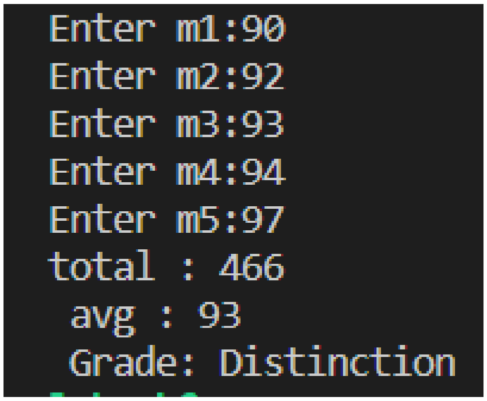
Experiment No 13

Shell script to display the pass class of a student

```
#!/bin/bash
echo "Enter m1:\c"
read m1
echo "Enter m2:\c"
read m2
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



```
Enter m1:90
Enter m2:92
Enter m3:93
Enter m4:94
Enter m5:97
total : 466
 avg : 93
 Grade: Distinction
```

Experiment No 14

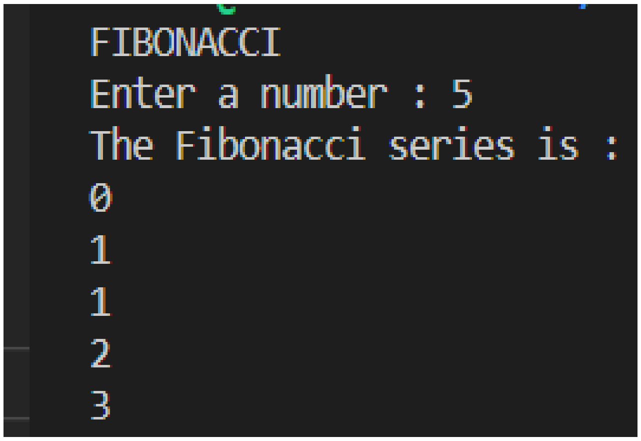
Shell script to find the Fibonacci series up to n

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



```
FIBONACCI
Enter a number : 5
The Fibonacci series is :
0
1
1
2
3
```

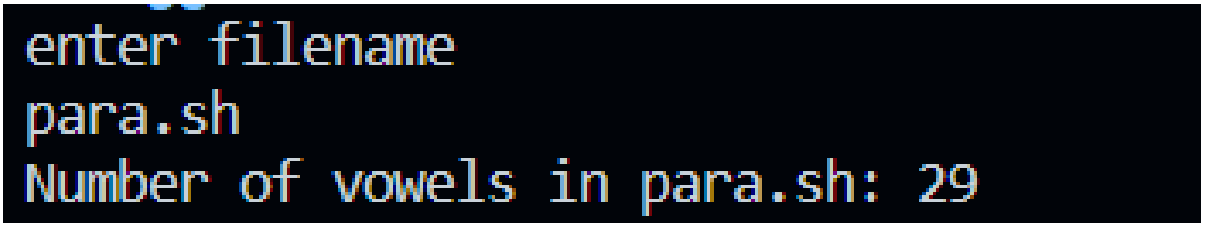
Experiment No 15

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



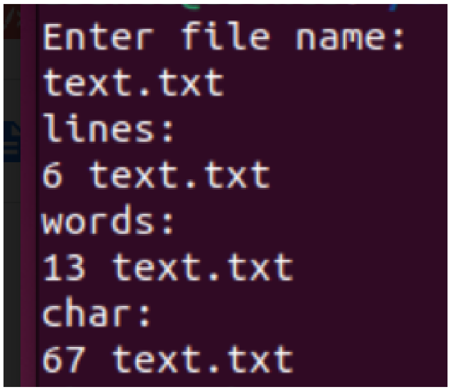
```
enter filename
para.sh
Number of vowels in para.sh: 29
```

Experiment No 16

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

```
#!/bin/sh
echo "Enter file name: "
read name
echo "lines:"
echo `wc -l $name`
echo "words:"
echo `wc -w $name`
echo "char:"
echo `wc -c $name`
```

Output



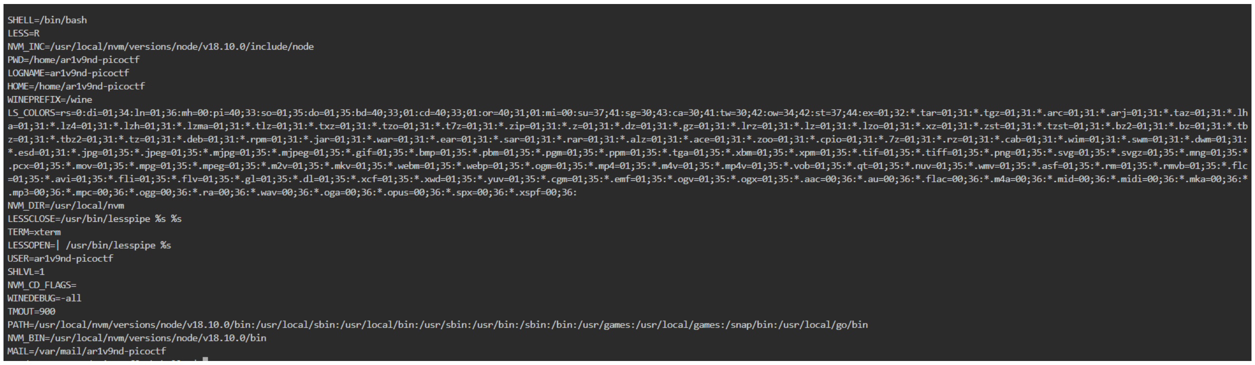
```
Enter file name:
text.txt
lines:
6 text.txt
words:
13 text.txt
char:
67 text.txt
```

Experiment No 17

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

```
#include<stdio.h>
int main(int argc, char *argv[], char * envp[])
{
int i;
for (i = 0; envp[i] != NULL; i++)
printf("\n%s", envp[i]);
getchar();
return 0;
}
```

Output



```
SHELL=/bin/bash
LESS=R
NVM_INC=/usr/local/nvm/versions/node/v18.10.0/include/node
PWD=/home/ar1v9nd-picoctf
LOGNAME=ar1v9nd-picoctf
HOME=/home/ar1v9nd-picoctf
WINEPREFIX=/wine
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=37;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arc=01;31:*.arj=01;31:*.taz=01;31:*.lh
a=01;31:*.lz4=01;31:*.lzh=01;31:*.lzma=01;31:*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.t7z=01;31:*.zip=01;31:*.z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lz=01;31:*.lzo=01;31:*.xz=01;31:*.zst=01;31:*.tzst=01;31:*.bz2=01;31:*.bz=01;31:*.tb
z=01;31:*.tbz2=01;31:*.tz=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01;31:*.war=01;31:*.ear=01;31:*.sar=01;31:*.rar=01;31:*.alz=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.cab=01;31:*.wim=01;31:*.swm=01;31:*.dwm=01;31:
*.esd=01;31:*.jpg=01;35:*.jpeg=01;35:*.mjpg=01;35:*.mjpeg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=01;35:*.mng=01;35:*
.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01;35:*.webm=01;35:*.webp=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc
=01;35:*.avi=01;35:*.fli=01;35:*.flv=01;35:*.gl=01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.ogv=01;35:*.ogx=01;35:*.aac=00;36:*.au=00;36:*.flac=00;36:*.m4a=00;36:*.mid=00;36:*.midi=00;36:*.mka=00;36:*
.mp3=00;36:*.mpc=00;36:*.ogg=00;36:*.ra=00;36:*.wav=00;36:*.oga=00;36:*.opus=00;36:*.spx=00;36:*.xspf=00;36:
NVM_DIR=/usr/local/nvm
LESSCLOSE=/usr/bin/lesspipe %s %s
TERM=xterm
LESSOPEN=| /usr/bin/lesspipe %s
USER=ar1v9nd-picoctf
SHLVL=1
NVM_CD_FLAGS=
WINEDEBUG=-all
TMOUT=900
PATH=/usr/local/nvm/versions/node/v18.10.0/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/usr/local/go/bin
NVM_BIN=/usr/local/nvm/versions/node/v18.10.0/bin
MAIL=/var/mail/ar1v9nd-picoctf
```

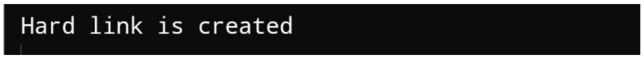
Experiment No 18

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



```
Hard link is created
```

Experiment No 19

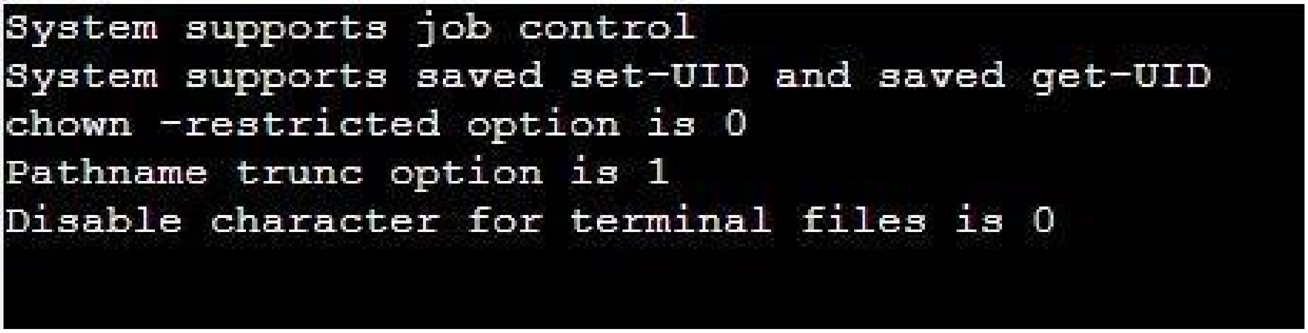
Write a C/C++ POSIX compliant program that prints the POSIX defined Configuration options supported on any given system using feature test macros.

```
#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



```
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
```

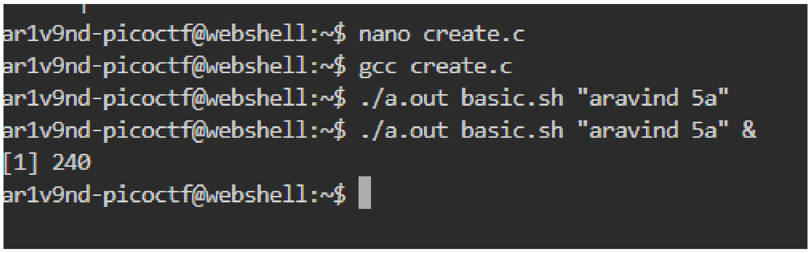
Experiment No 20

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



```
ar1v9nd-picoctf@webshell:~$ nano create.c
ar1v9nd-picoctf@webshell:~$ gcc create.c
ar1v9nd-picoctf@webshell:~$ ./a.out basic.sh "aravind 5a"
ar1v9nd-picoctf@webshell:~$ ./a.out basic.sh "aravind 5a" &
[1] 240
ar1v9nd-picoctf@webshell:~$
```