

# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



## LAB REPORT

on

## UNIX SHELL AND PROGRAMMING

*Submitted by*

**EKTHA KARTHIK(1BM20CS045)**

*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 **EKTHA KARTHIK(1BM20CS045)**, 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.

**Madhavi R P**  
Assistant 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.
<b>1</b>		Shell script to find if the given year is leap or not	<b>4-5</b>
<b>2</b>		Shell script to find the area of a circle	<b>6</b>
<b>3</b>		Shell script to check whether the number is zero/ positive/ negative	<b>7</b>
<b>4</b>		Shell script to find the biggest of three numbers	<b>8</b>
<b>5</b>		Shell script to find the factorial of a number	<b>9</b>
<b>6</b>		Shell script to compute the gross salary of an employee	<b>10</b>
<b>7</b>		Shell script to convert the temperature Fahrenheit to Celsius	<b>11</b>
<b>8</b>		Shell script to perform arithmetic operations on given two numbers	<b>12-13</b>
<b>9</b>		Shell script to find the sum of even numbers upto n	<b>14</b>
<b>10</b>		Shell script to print the combinations of numbers 123	<b>15</b>
<b>11</b>		Shell script to find the power of a number	<b>16</b>
<b>12</b>		Shell script to find the sum of n natural numbers	<b>17</b>
<b>13</b>		Shell script to display the pass class of a student	<b>18-19</b>
<b>14</b>		Shell script to find the Fibonacci series up to n	<b>20</b>
<b>15</b>		Shell script to count the number of vowels of a string	<b>21</b>
<b>16</b>		Shell script to check number of lines, words, characters in a file	<b>22</b>
<b>17</b>		Write a C/C++ program to that outputs the contents of its environment list	<b>23</b>
<b>18</b>		Write a C/C++ program to emulate the Unix ln command	<b>24-25</b>
<b>19</b>		Write a C/C++ POSIX compliant program that prints the POSIX defined Configuration options supported on any given system using feature test macros.	<b>26-27</b>
<b>20</b>		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.	<b>28-29</b>

# Experiment 1

## 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 – Screen shot

```
^C
Karthiks-MacBook-Air:shellu karthikraveen$ sh file3.sh
Enter an Year:
443
443 is leap year
Karthiks-MacBook-Air:shellu karthikraveen$ sh file3.sh
Enter an Year:
444
444 is not leap year
Karthiks-MacBook-Air:shellu karthikraveen$ █
```

## Experiment 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
```

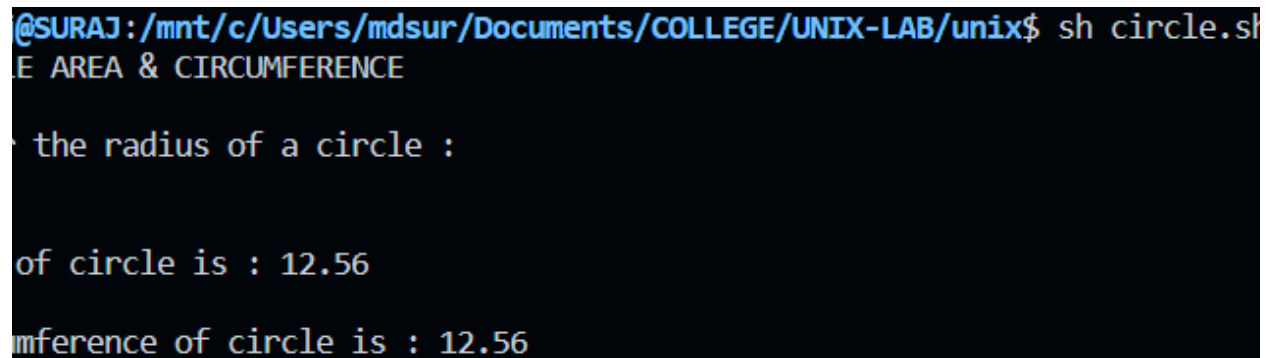
```
area=$(echo "scale=2; 22/7 * ($r * $r)" | bc)
```

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

```
echo "\nArea of circle is : $area"
```

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

### Output – Screen shot



```
@SURAJ:/mnt/c/Users/mdsur/Documents/COLLEGE/UNIX-LAB/unix$ sh circle.sh
CIRCLE AREA & CIRCUMFERENCE

Enter the radius of a circle : 4

Area of circle is : 12.56

Circumference of circle is : 25.12
```

## Experiment 3

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

```
echo "\n-----Pos Neg Zero-----\n"

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 – Screen shot



```
suraj@SURAJ:/mnt/c/Users/mdsur/Documents/COLLEGE/UNIX-LAB/unix/pos_neg$ sh pos_neg.sh
-----Pos Neg Zero-----
Enter the number :
2
2 is positive
suraj@SURAJ:/mnt/c/Users/mdsur/Documents/COLLEGE/UNIX-LAB/unix/pos_neg$ sh pos_neg.sh
-----Pos Neg Zero-----
Enter the number :
-1
-1 is negative
suraj@SURAJ:/mnt/c/Users/mdsur/Documents/COLLEGE/UNIX-LAB/unix/pos_neg$ sh pos_neg.sh
-----Pos Neg Zero-----
Enter the number :
0
0 is zero
```

## Experiment 4

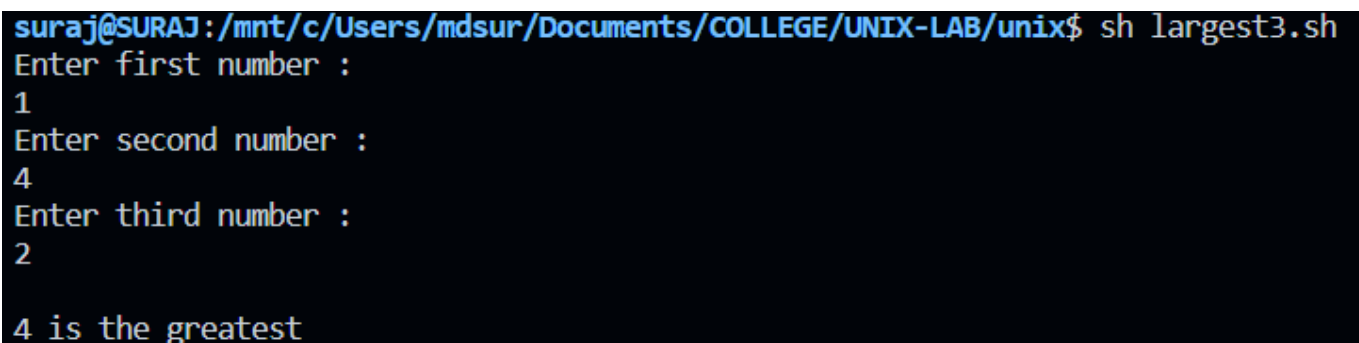
### Shell script to find the biggest of three numbers

```
#!/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 – Screen shot



```
suraj@SURAJ:/mnt/c/Users/mdsur/Documents/COLLEGE/UNIX-LAB/unix$ sh largest3.sh
Enter first number :
1
Enter second number :
4
Enter third number :
2
4 is the greatest
```



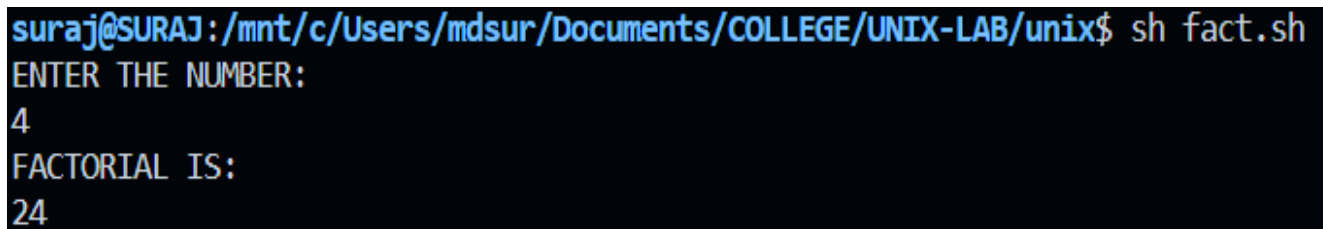
## Experiment 5

### 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 – Screen shot



```
suraj@SURAJ:/mnt/c/Users/mdsur/Documents/COLLEGE/UNIX-LAB/unix$ sh fact.sh
ENTER THE NUMBER:
4
FACTORIAL IS:
24
```

## Experiment 6

### Shell script to compute the gross salary of an employee

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

```
echo "\nEnter name of Employee :"  
read name
```

```
echo "\nEnter DA :"  
read da
```

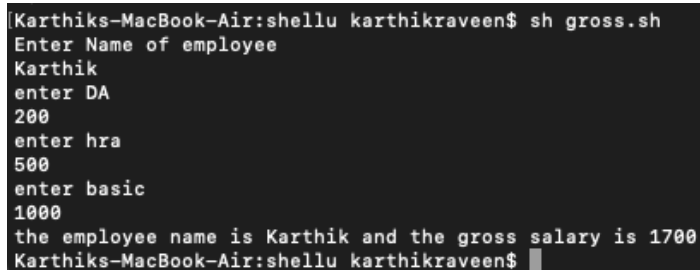
```
echo "\nEnter HRA:"  
read hra
```

```
echo "\nEnter basic"  
read basic
```

```
sal=$(( $da + $hra + $basic ))
```

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

### Output – Screen shot



```
[Karthiks-MacBook-Air:shellu karthikraveen$ sh gross.sh  
Enter Name of employee  
Karthik  
enter DA  
200  
enter hra  
500  
enter basic  
1000  
the employee name is Karthik and the gross salary is 1700  
Karthiks-MacBook-Air:shellu karthikraveen$
```

## Experiment 7

### Shell script to convert the temperature Fahrenheit to Celsius

```
echo "\n-----Fahrenheit to Celcius-----\n"
```

```
echo "Enter temperature in F : "
```

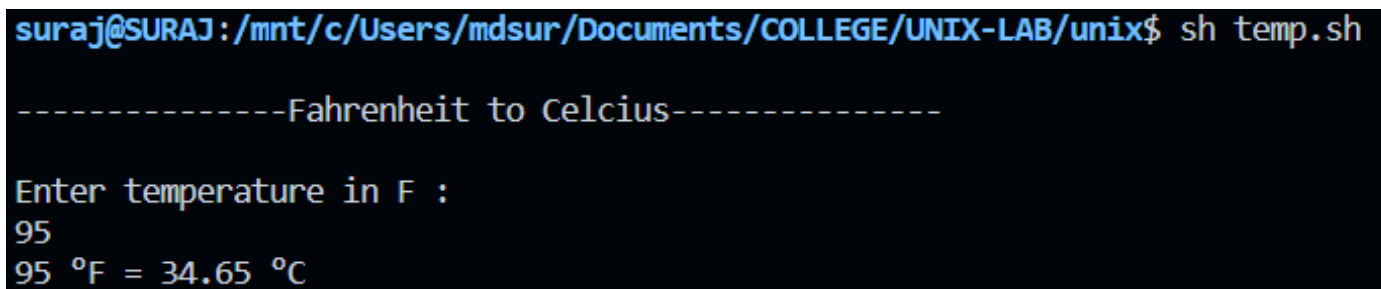
```
read f
```

```
# formula c=(5/9)*(f-32)
```

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

```
echo "$f °F = $c °C"
```

### Output – Screen shot



```
suraj@SURAJ:/mnt/c/Users/mdsur/Documents/COLLEGE/UNIX-LAB/unix$ sh temp.sh
-----Fahrenheit to Celcius-----
Enter temperature in F :
95
95 °F = 34.65 °C
```

## Experiment 8

### Shell script to perform arithmetic operations on given two numbers

```
echo "\n-----CALCULATOR-----\n"
```

```
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 – Screen shot

```
suraj@SURAJ:/mnt/c/Users/mdsur/Documents/COLLEGE/UNIX-LAB/unix$ sh calc.sh

-----CALCULATOR-----

Enter 2 Numbers :
3
4
Enter Operation :

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

1
7
```

## Experiment 9

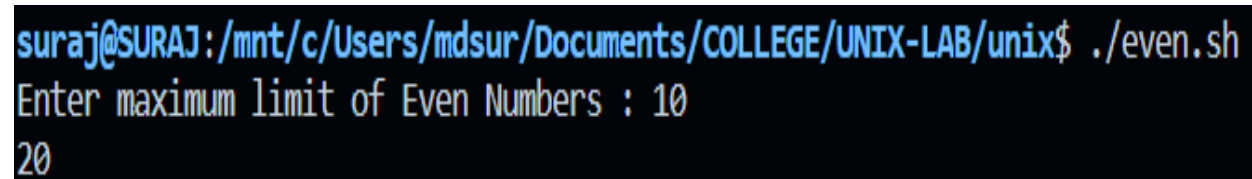
### Shell script to find the sum of even numbers upto n

```
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 – Screen shot



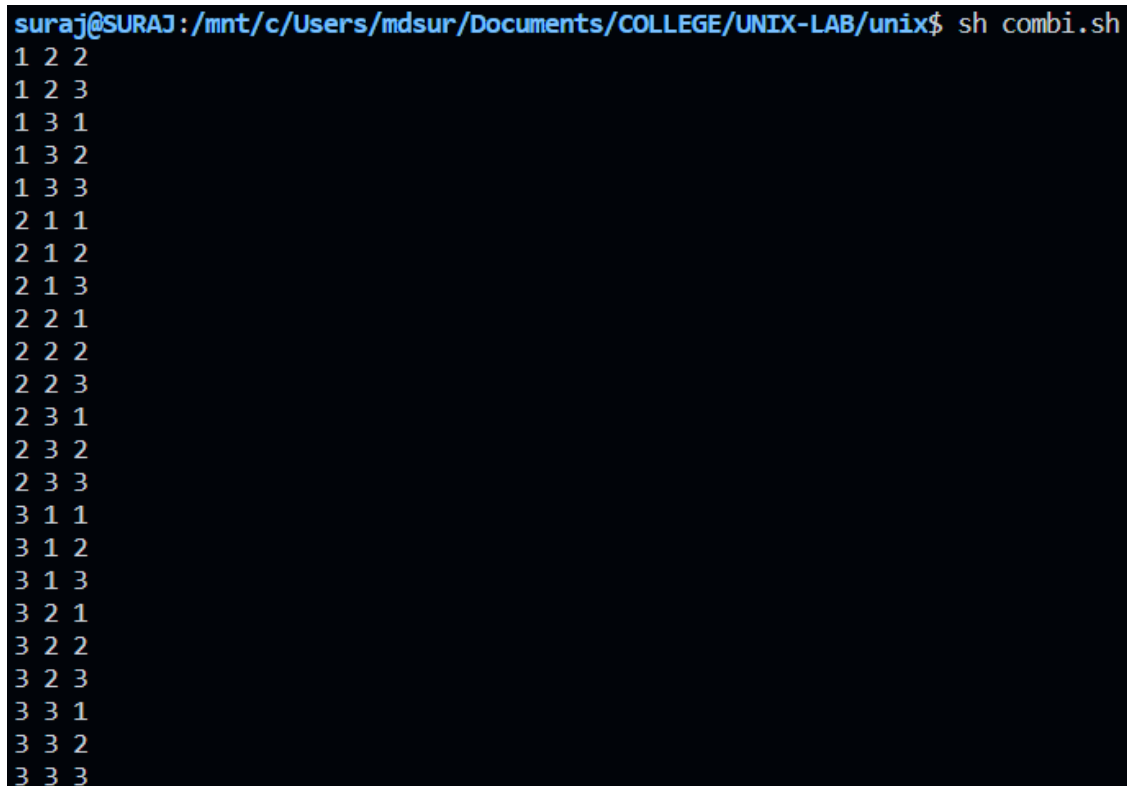
```
suraj@SURAJ:/mnt/c/Users/mdsur/Documents/COLLEGE/UNIX-LAB/unix$ ./even.sh
Enter maximum limit of Even Numbers : 10
20
```

## Experiment 10

### Shell script to print the combinations of numbers 123

```
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 – Screen shot



```
suraj@SURAJ:/mnt/c/Users/mdsur/Documents/COLLEGE/UNIX-LAB/unix$ sh combi.sh  
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 11

### Shell script to find the power of a number

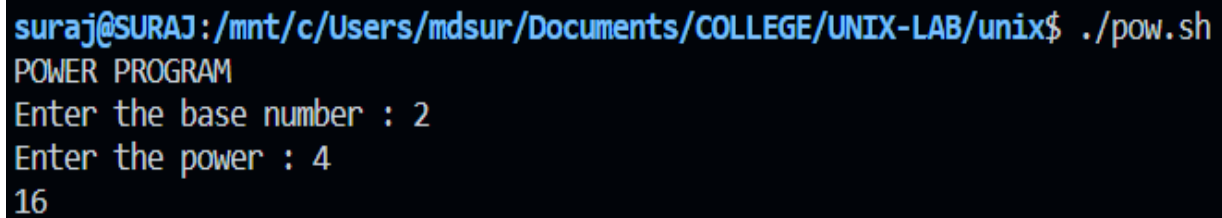
```
echo "POWER PROGRAM"
read -p "Enter the base number : " a
read -p "Enter the power : " b

res=1

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

echo $res
```

### Output – Screen shot



```
suraj@SURAJ:/mnt/c/Users/mdsur/Documents/COLLEGE/UNIX-LAB/unix$ ./pow.sh
POWER PROGRAM
Enter the base number : 2
Enter the power : 4
16
```



## Experiment 12

### Shell script to find the sum of n natural numbers

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

```
i=1
sum=0
echo "Digits : "
while [ $i -le $n ]
do
    echo "$i"
    sum=$(( $sum + $i ))
    i=$(( $i + 1 ))
done
echo "Sum=$sum"
```

### Output – Screen shot



```
suraj@SURAJ:/mnt/c/Users/mdsur/Documents/COLLEGE/UNIX-LAB/unix$ sh sum_of_n.sh
Enter number : 5
Digits :
1
2
3
4
5
Sum=15
```

## Experiment 13

### Shell script to display the pass class of a student

```
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 – Screen shot

```
suraj@SURAJ:/mnt/c/Users/mdsur/Documents/COLLEGE/UNIX-LAB/unix$ sh pass_class.sh
Enter m1:90
Enter m2:56
Enter m3:87
Enter m4:44
Enter m5:56
total : 333
  avg : 66
Grade: First Class
```

## Experiment 14

### Shell script to find the Fibonacci series up to n

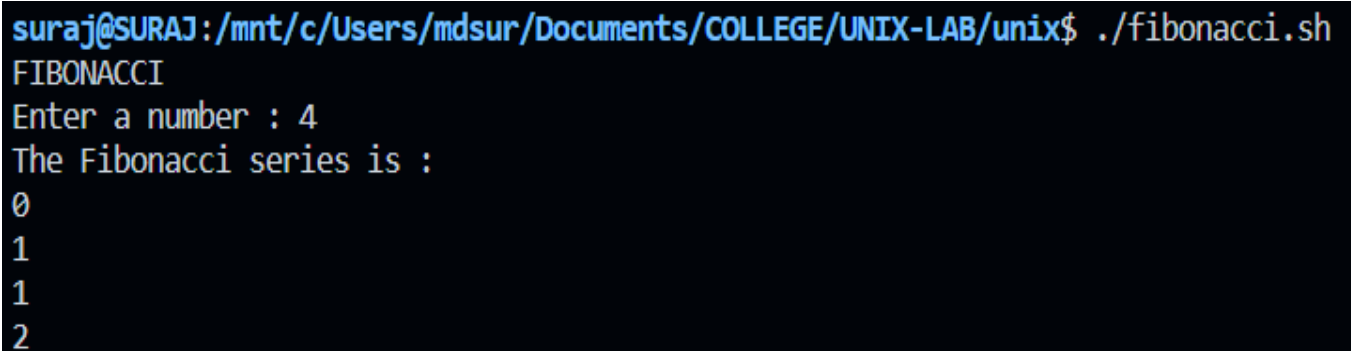
```
echo "FIBONACCI"
read -p "Enter a number : " N

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 – Screen shot



```
suraj@SURAJ:/mnt/c/Users/mdsur/Documents/COLLEGE/UNIX-LAB/unix$ ./fibonacci.sh
FIBONACCI
Enter a number : 4
The Fibonacci series is :
0
1
1
2
```

## Experiment 15

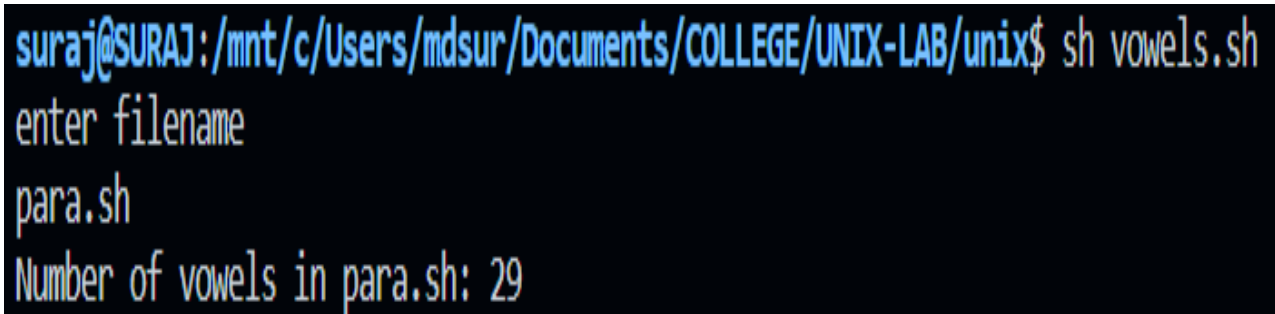
### Shell script to count the number of vowels of a string

```
echo "enter filename"  
read filename
```

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

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

### Output – Screen shot



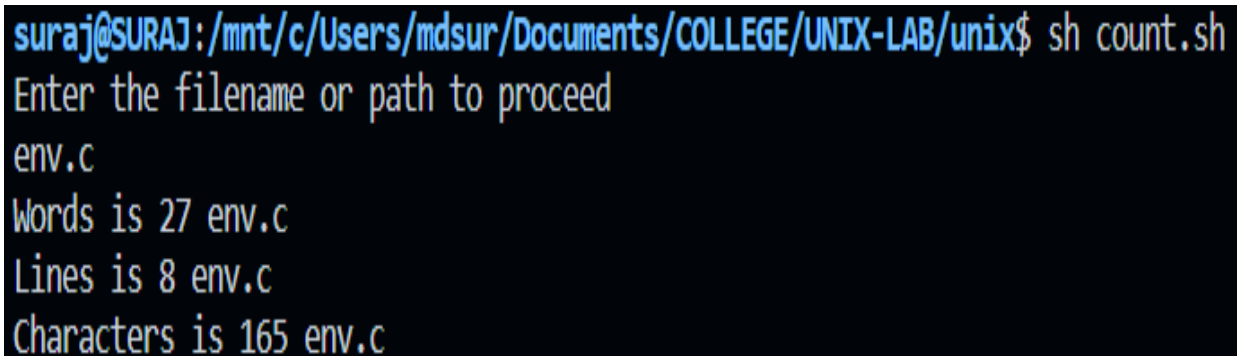
```
suraj@SURAJ:/mnt/c/Users/mdsur/Documents/COLLEGE/UNIX-LAB/unix$ sh vowels.sh  
enter filename  
para.sh  
Number of vowels in para.sh: 29
```

## Experiment 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 – Screen shot



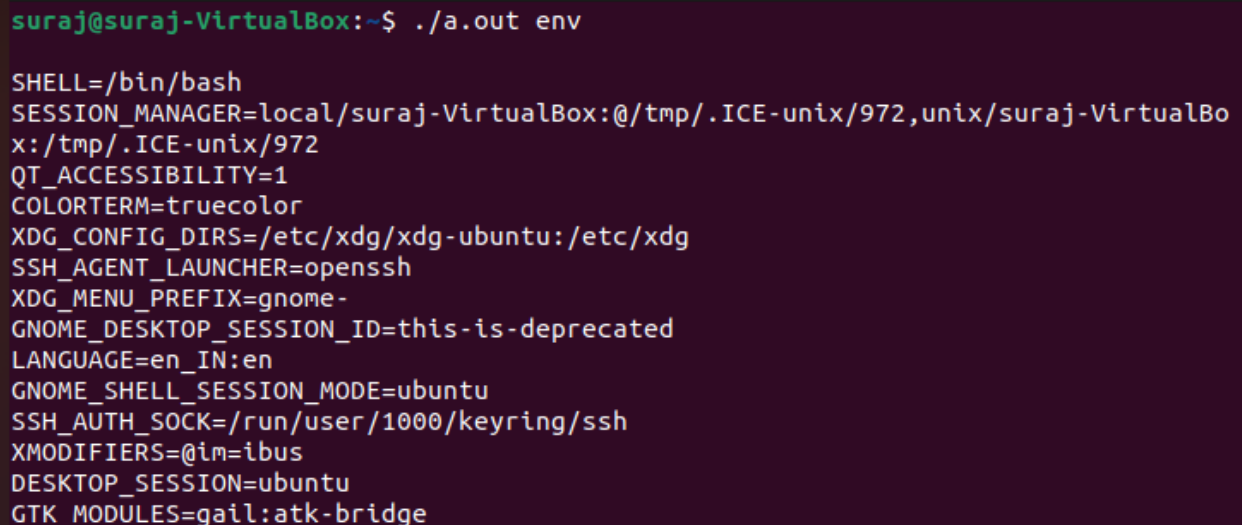
```
suraj@SURAJ:/mnt/c/Users/mdsur/Documents/COLLEGE/UNIX-LAB/unix$ sh count.sh
Enter the filename or path to proceed
env.c
Words is 27 env.c
Lines is 8 env.c
Characters is 165 env.c
```

## Experiment 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 – Screen shot**



```
suraj@suraj-VirtualBox:~$ ./a.out env
SHELL=/bin/bash
SESSION_MANAGER=local/suraj-VirtualBox:@/tmp/.ICE-unix/972,unix/suraj-VirtualBo
x:/tmp/.ICE-unix/972
QT_ACCESSIBILITY=1
COLORTERM=truecolor
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/etc/xdg
SSH_AGENT_LAUNCHER=openssh
XDG_MENU_PREFIX=gnome-
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
LANGUAGE=en_IN:en
GNOME_SHELL_SESSION_MODE=ubuntu
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
XMODIFIERS=@im=ibus
DESKTOP_SESSION=ubuntu
GTK_MODULES=gail:atk-bridge
```

## Experiment 18

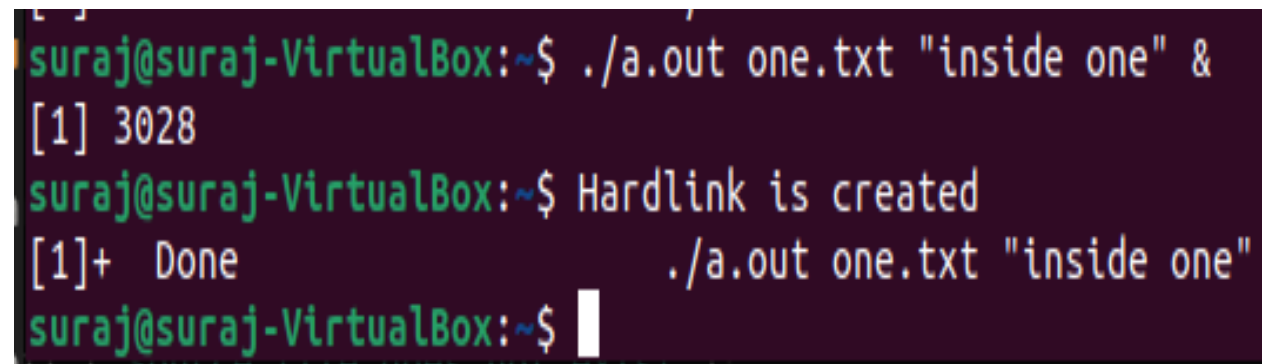
**Write a C/C++ program to emulate the Unix ln command**

```
#include <stdio.h>
#include <unistd.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("Hardlink is created");  
return 0;  
}  
return 0;  
}
```

### Output – Screen shot



A terminal window screenshot showing the execution of a program. The prompt is `suraj@suraj-VirtualBox:~$`. The user enters `./a.out one.txt "inside one" &`. The output shows `[1] 3028` on the next line, followed by `suraj@suraj-VirtualBox:~$ Hardlink is created`. Then, `[1]+ Done` is shown, followed by `./a.out one.txt "inside one"`. Finally, the prompt `suraj@suraj-VirtualBox:~$` is shown with a cursor.

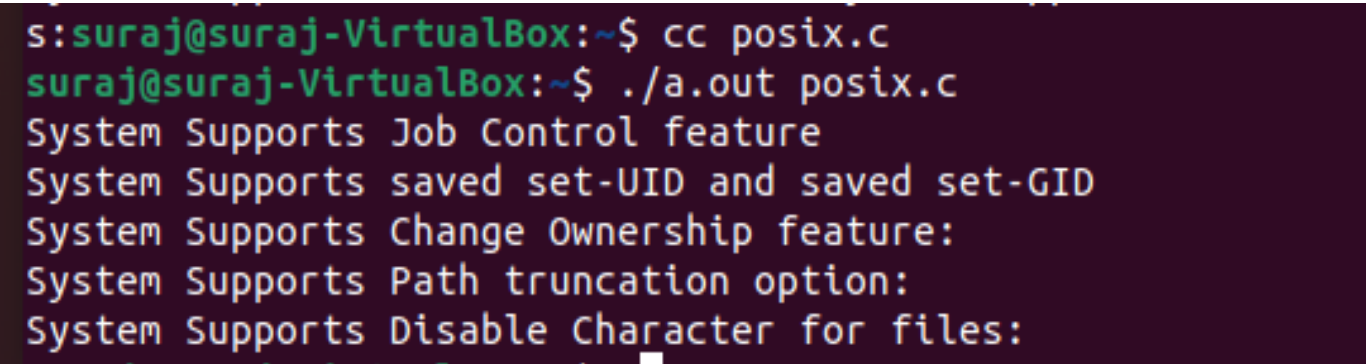
## Experiment 19

**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<unistd.h>
#include<stdio.h>
int main()
{
#ifdef _POSIX_JOB_CONTROL
printf("System Supports Job Control feature\n");
#else
printf("System doesnot support job control\n");
#endif
#ifdef _POSIX_SAVED_IDS
printf("System Supports saved set-UID and saved set-GID\n");
#else
printf("System doesnot support saved set-UID\n");
#endif
#ifdef _POSIX_CHOWN_RESTRICTED
printf("System Supports Change Ownership feature:\n");
#else
printf("System doesnot support change Ownership feature\n");
#endif
#ifdef _POSIX_NO_TRUNC
printf("System Supports Path truncation option:\n");
#else
printf("System doesnot support Path truncation \n");
#endif
}
```

```
#ifdef _POSIX_VDISABLE
printf("System Supports Disable Character for files:\n");
#else
printf("System doesnot support Disable Characters \n");
#endif
return 0;
}
```

### Output – Screen shot



```
s:suraj@suraj-VirtualBox:~$ cc posix.c
suraj@suraj-VirtualBox:~$ ./a.out posix.c
System Supports Job Control feature
System Supports saved set-UID and saved set-GID
System Supports Change Ownership feature:
System Supports Path truncation option:
System Supports Disable Character for files:
```

## Experiment 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 – Screen shot

```
suraj@suraj-VirtualBox:~$ cc interprocess.c
suraj@suraj-VirtualBox:~$ ./a.out interprocess "5th semester" &
[1] 3801
suraj@suraj-VirtualBox:~$ ./a.out interprocess
5th semester[1]+  Done                  ./a.out interprocess "5th semester"
suraj@suraj-VirtualBox:~$ s
```