VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.

LAB REPORT
on

Unix Shell Programming Lab Report

Submitted by

HEMAMALA MN(1BM20CS056)

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 PROGRAMMING**" carried out by **HEMAMALA MN(1BM20CS056),** who is a 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 Programming - (20CS5PCUSP)** work prescribed for the said degree.

Prameetha Pai
Assistant Professor
Department of CSE
BMSCE, Bengaluru

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

`

Index

17	2/1/23	Shell script that file name as arguments and searches for a specific word on the file and stops when word is found	20
18	9/1/23	Shell script to check the number of line, words and characters on an entered file	21
19	9/1/23	Shell script to display count, sum of positive number and sum of negative numbers	
20			
21			

Program 1

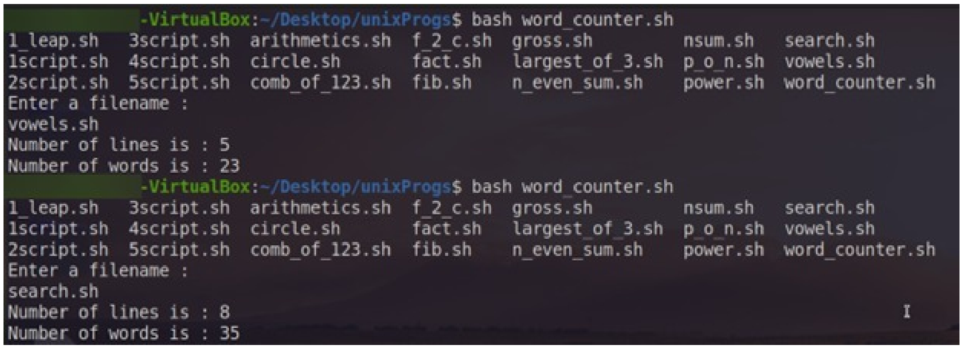
Aim of the program - A shell script which displays the number of words and lines in a file.

Program

```
#!/bin/bash
ls
echo Enter a filename :
read fname
echo -n Number of lines is :
wc -l $fname | head -c 3
echo
echo -n Number of words is :
wc -w $fname | head -c 3
```

```
echo
```

Output Screen shot



```
[redacted]-VirtualBox:~/Desktop/unixProgs$ bash word_counter.sh
1_leap.sh   3script.sh  arithmetics.sh  f_2_c.sh  gross.sh          nsum.sh   search.sh
1script.sh  4script.sh  circle.sh       fact.sh   largest_of_3.sh   p_o_n.sh  vowels.sh
2script.sh  5script.sh  comb_of_123.sh  fib.sh    n_even_sum.sh     power.sh  word_counter.sh
Enter a filename :
vowels.sh
Number of lines is : 5
Number of words is : 23
[redacted]-VirtualBox:~/Desktop/unixProgs$ bash word_counter.sh
1_leap.sh   3script.sh  arithmetics.sh  f_2_c.sh  gross.sh          nsum.sh   search.sh
1script.sh  4script.sh  circle.sh       fact.sh   largest_of_3.sh   p_o_n.sh  vowels.sh
2script.sh  5script.sh  comb_of_123.sh  fib.sh    n_even_sum.sh     power.sh  word_counter.sh
Enter a filename :
search.sh
Number of lines is : 8
Number of words is : 35
```

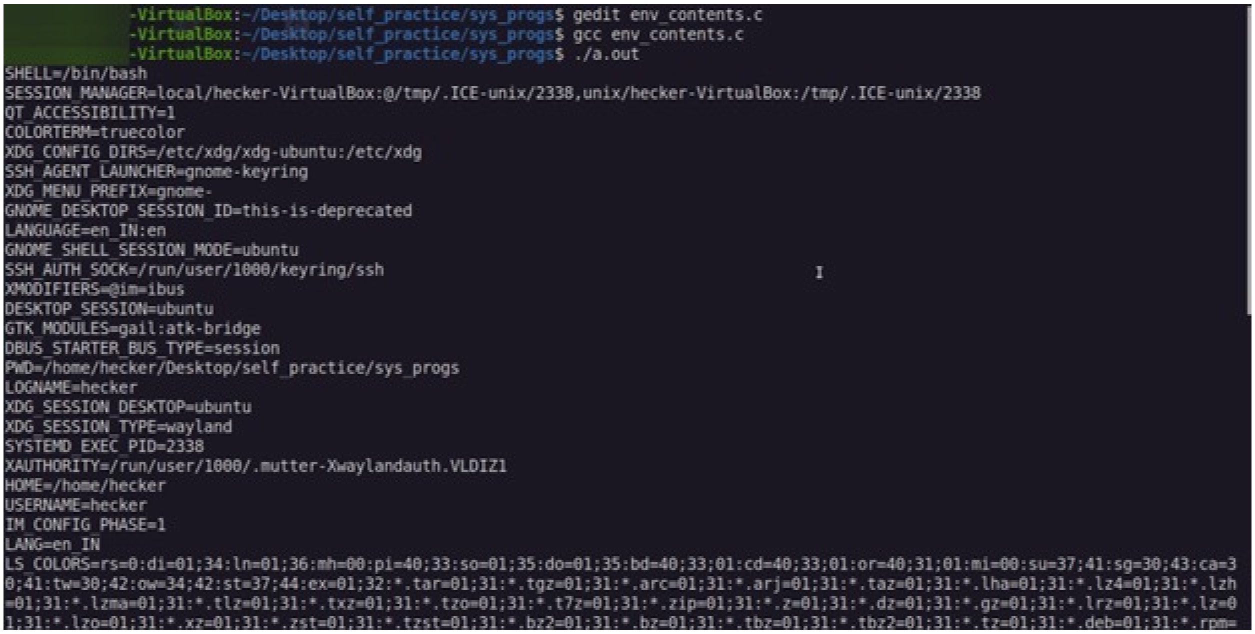
Program 2

Aim of the program - Shell script which displays list of files in a given directory.

Program

```
#!/bin/bash

echo “list of files present in this directory:”

echo “enter directory name”

read dir

echo “list of files in this directory”

ls $dir
```

Output screenshot



```
[redacted]-VirtualBox:~/Desktop/self_practice/sys_progs$ gedit env_contents.c
-VirtualBox:~/Desktop/self_practice/sys_progs$ gcc env_contents.c
-VirtualBox:~/Desktop/self_practice/sys_progs$ ./a.out
SHELL=/bin/bash
SESSION_MANAGER=local/hecker-VirtualBox:@/tmp/.ICE-unix/2338,unix/hecker-VirtualBox:/tmp/.ICE-unix/2338
QT_ACCESSIBILITY=1
COLORTERM=truecolor
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/etc/xdg
SSH_AGENT_LAUNCHER=gnome-keyring
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
DBUS_STARTER_BUS_TYPE=session
PWD=/home/hecker/Desktop/self_practice/sys_progs
LOGNAME=hecker
XDG_SESSION_DESKTOP=ubuntu
XDG_SESSION_TYPE=wayland
SYSTEMD_EXEC_PID=2338
XAUTHORITY=/run/user/1000/.mutter-Xwaylandauth.VLDIZ1
HOME=/home/hecker
USERNAME=hecker
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=37;41:sg=30;43:ca=3
0;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:*.lha=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=0
1;31:*.lzo=01;31:*.xz=01;31:*.zst=01;31:*.tzst=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.tz=01;31:*.deb=01;31:*.rpm=
```

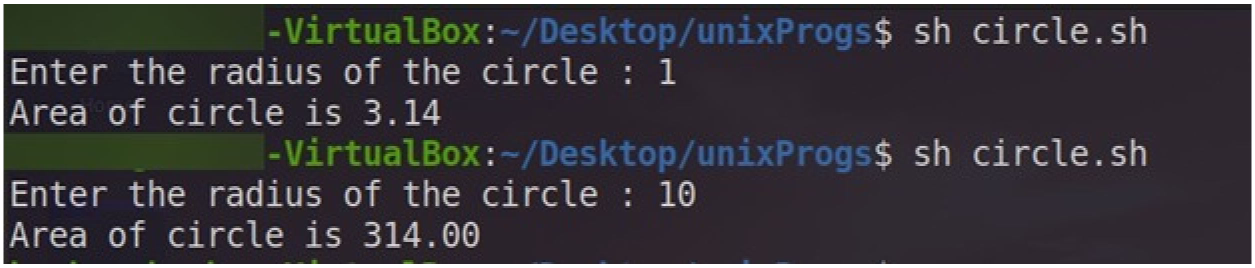
Program - 3

<u>Aim of the program -</u> To find the area of the circle

<u>Program</u>

```
#!/bin/bash

echo -n "Enter the radius of the circle : "

read r

echo "Area of circle is `echo 3.14*$r^2 | bc`"
```

<u>Output Screenshot</u>



```
[redacted]-VirtualBox:~/Desktop/unixProgs$ sh circle.sh
Enter the radius of the circle : 1
Area of circle is 3.14
[redacted]-VirtualBox:~/Desktop/unixProgs$ sh circle.sh
Enter the radius of the circle : 10
Area of circle is 314.00
```

Program 4

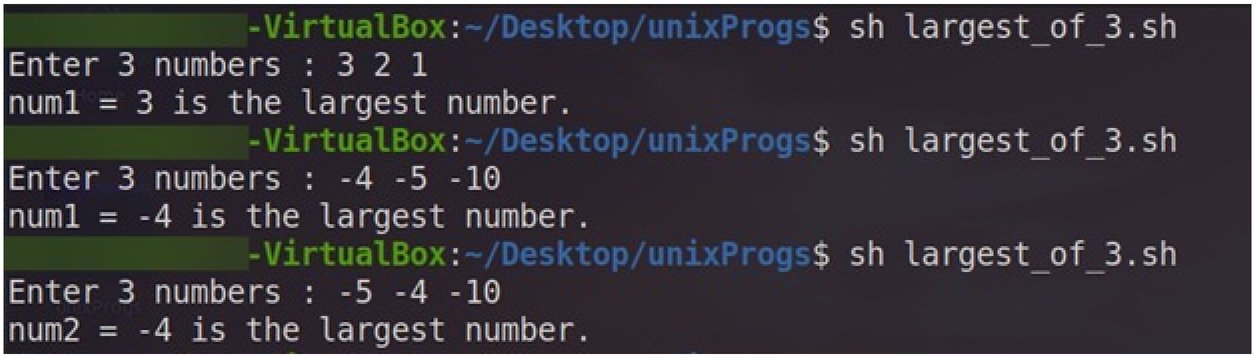
<u>Aim of the Program -</u> To find the largest of three numbers

<u>Program</u>

```
#!/bin/bash
echo -n "Enter 3 numbers : "
read num1 num2 num3
if [ $num1 -gt $num2 ] && [ $num1 -gt $num3 ]
then
```

```
        echo num1 = $num1 is the largest number.
elif [ $num2 -gt $num1 ] && [ $num2 -gt $num3 ]
then
        echo num2 = $num2 is the largest number.
else
        echo num3 = $num3 is the largest number.
fi
```

<u>Output Screenshot</u>



```
[redacted]-VirtualBox:~/Desktop/unixProgs$ sh largest_of_3.sh
Enter 3 numbers : 3 2 1
num1 = 3 is the largest number.
[redacted]-VirtualBox:~/Desktop/unixProgs$ sh largest_of_3.sh
Enter 3 numbers : -4 -5 -10
num1 = -4 is the largest number.
[redacted]-VirtualBox:~/Desktop/unixProgs$ sh largest_of_3.sh
Enter 3 numbers : -5 -4 -10
num2 = -4 is the largest number.
```

Program 5

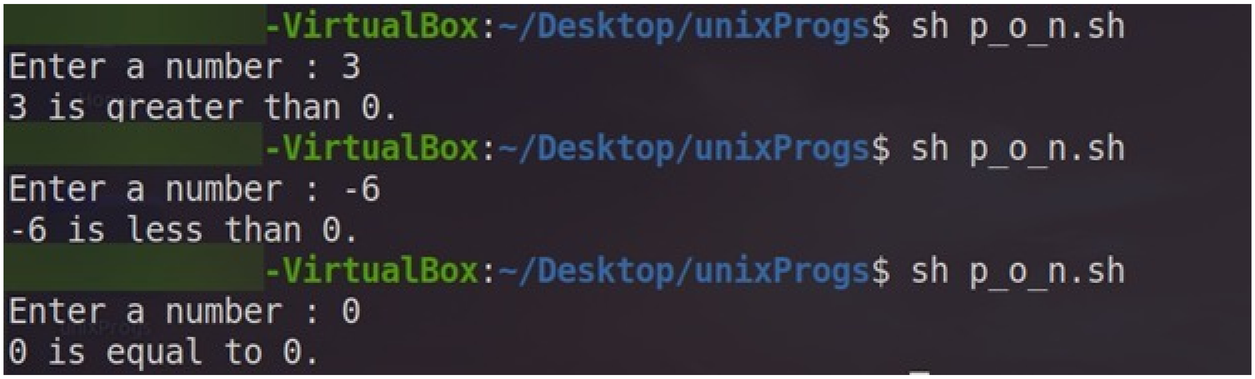
<u>Aim of the Program - </u>Shell program to find whether a number is a positive, negative or zero

<u>Program</u>

```
#!/bin/bash
echo “Enter a number”
read num
if [$num -lt 0]
then
```

```
echo “Negative”
elif [$num -gt 0]
echo “Positive”
else
echo “ It is 0”
fi
```

Output Screenshot



```
[redacted]-VirtualBox:~/Desktop/unixProgs$ sh p_o_n.sh
Enter a number : 3
3 is greater than 0.
[redacted]-VirtualBox:~/Desktop/unixProgs$ sh p_o_n.sh
Enter a number : -6
-6 is less than 0.
[redacted]-VirtualBox:~/Desktop/unixProgs$ sh p_o_n.sh
Enter a number : 0
0 is equal to 0.
```

Program 6

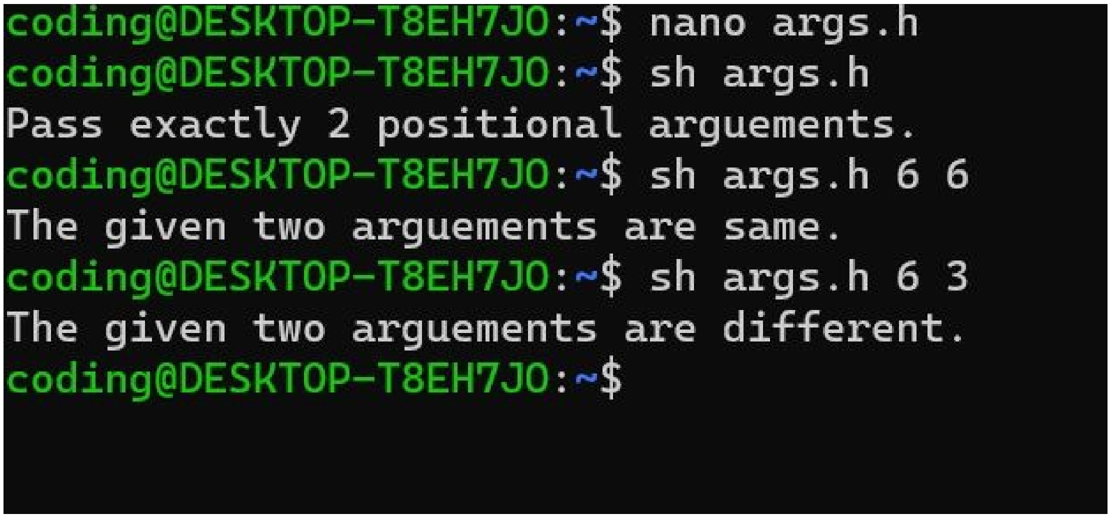
Aim of the program - Shell program to find whether the entered two arguments are equal or not

Program

```
#!/bin/bash
if [ $# -eq 2 ]
then
        if [ $1 = $2 ]
        then
```

```
                echo "The two arguments are the same."
        else
                echo "The given two arguments are different."
        fi
else
        echo "Pass exactly 2 positional arguments."
fi
```

Output Screenshot



```
coding@DESKTOP-T8EH7JO:~$ nano args.h
coding@DESKTOP-T8EH7JO:~$ sh args.h
Pass exactly 2 positional arguements.
coding@DESKTOP-T8EH7JO:~$ sh args.h 6 6
The given two arguements are same.
coding@DESKTOP-T8EH7JO:~$ sh args.h 6 3
The given two arguements are different.
coding@DESKTOP-T8EH7JO:~$
```

Program 7

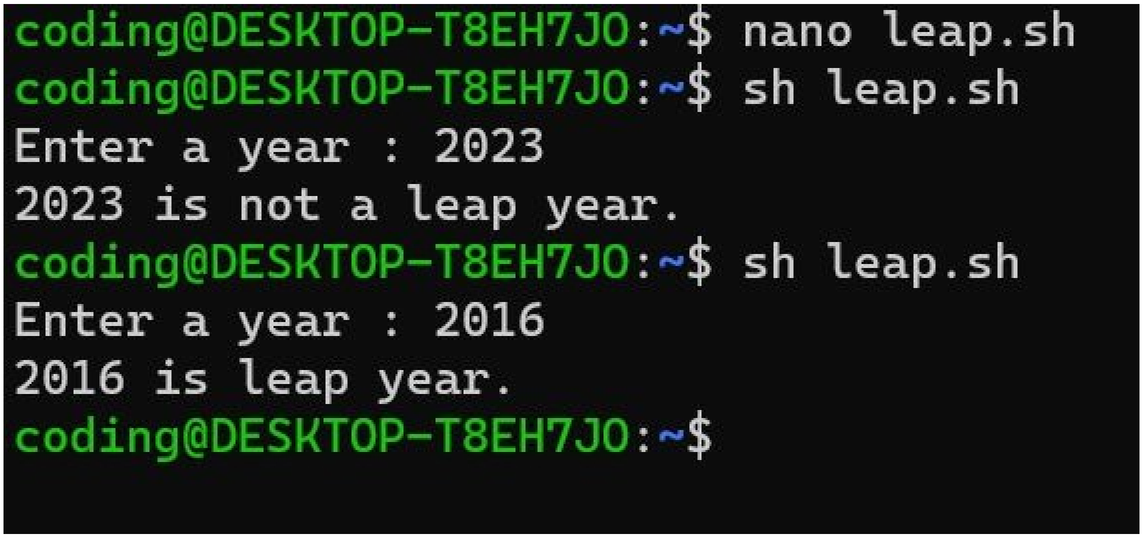
Aim of the program - Shell programming to find whether a given year is a leap year or not

Program

```
#!/bin/bash
echo -n "Enter a year : "
read y
if [ `expr $y % 400` -eq 0 ]
then
```

```
        echo $y is leap year.
elif [ `expr $y % 4` -eq 0 ] && [ `expr $y % 100` -ne 0 ]
then
        echo $y is leap year.
else
        echo $y is not a leap year.
fi
```

Output Screenshot



```
coding@DESKTOP-T8EH7JO:~$ nano leap.sh
coding@DESKTOP-T8EH7JO:~$ sh leap.sh
Enter a year : 2023
2023 is not a leap year.
coding@DESKTOP-T8EH7JO:~$ sh leap.sh
Enter a year : 2016
2016 is leap year.
coding@DESKTOP-T8EH7JO:~$
```

Program - 8

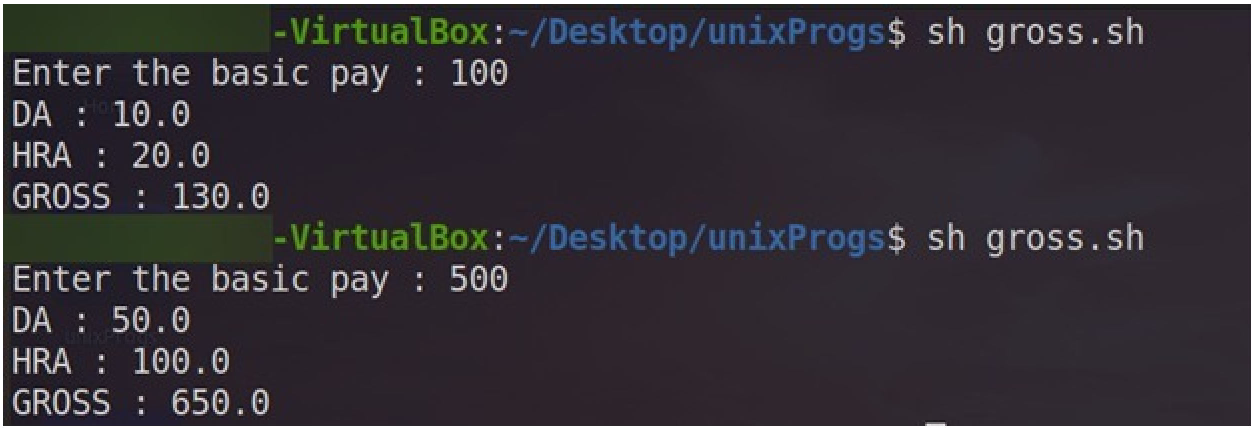
Aim of the program - Shell programming to find the gross salary of an employee

Program

```
#!/bin/bash
echo -n "Enter the basic pay : "
read b
da=`echo "scale=4; $b * 0.1" | bc`
hra=`echo "scale=4; $b * 0.2" | bc`
```

```
echo "DA : $da"
echo "HRA : $hra"
echo "GROSS : `echo "scale=4; $b + $da + $hra" | bc` "
```

<u>Output Screenshot</u>



```
[redacted]-VirtualBox:~/Desktop/unixProgs$ sh gross.sh
Enter the basic pay : 100
DA : 10.0
HRA : 20.0
GROSS : 130.0
[redacted]-VirtualBox:~/Desktop/unixProgs$ sh gross.sh
Enter the basic pay : 500
DA : 50.0
HRA : 100.0
GROSS : 650.0
```

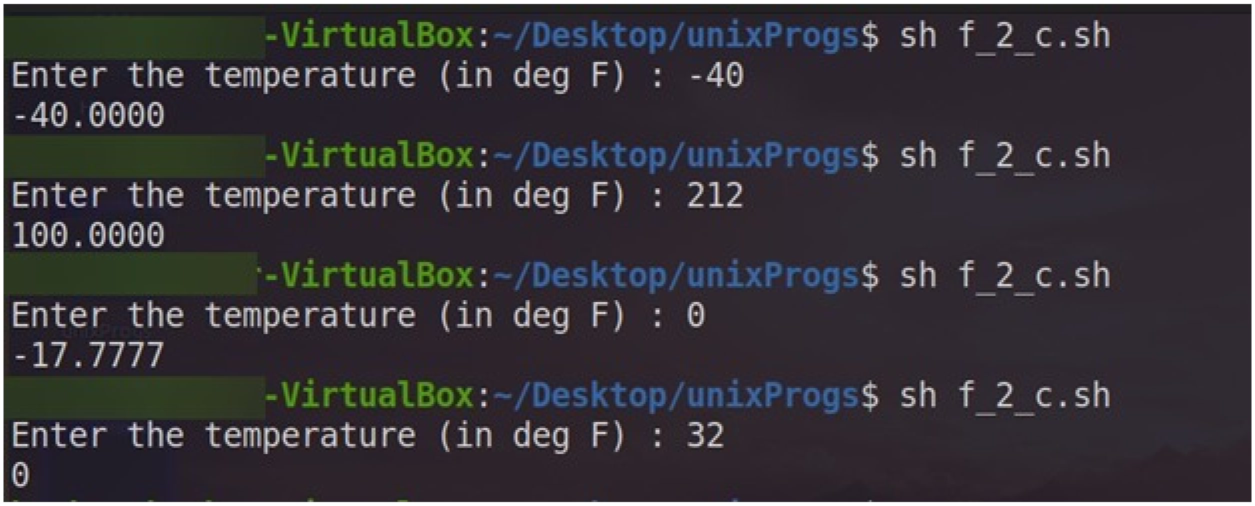
Program 9

<u>Aim of program - </u>Shell program to convert from fahrenheit to celsius

<u>Program</u>

```
#!/bin/bash
echo -n "Enter the temperature (in deg F) : "
read t
echo "scale=4; ($t - 32)*5/9" | bc
```

<u>Output Screenshot</u>



```
[redacted]-VirtualBox:~/Desktop/unixProgs$ sh f_2_c.sh
Enter the temperature (in deg F) : -40
-40.0000
[redacted]-VirtualBox:~/Desktop/unixProgs$ sh f_2_c.sh
Enter the temperature (in deg F) : 212
100.0000
[redacted]-VirtualBox:~/Desktop/unixProgs$ sh f_2_c.sh
Enter the temperature (in deg F) : 0
-17.7777
[redacted]-VirtualBox:~/Desktop/unixProgs$ sh f_2_c.sh
Enter the temperature (in deg F) : 32
0
```

Program 10

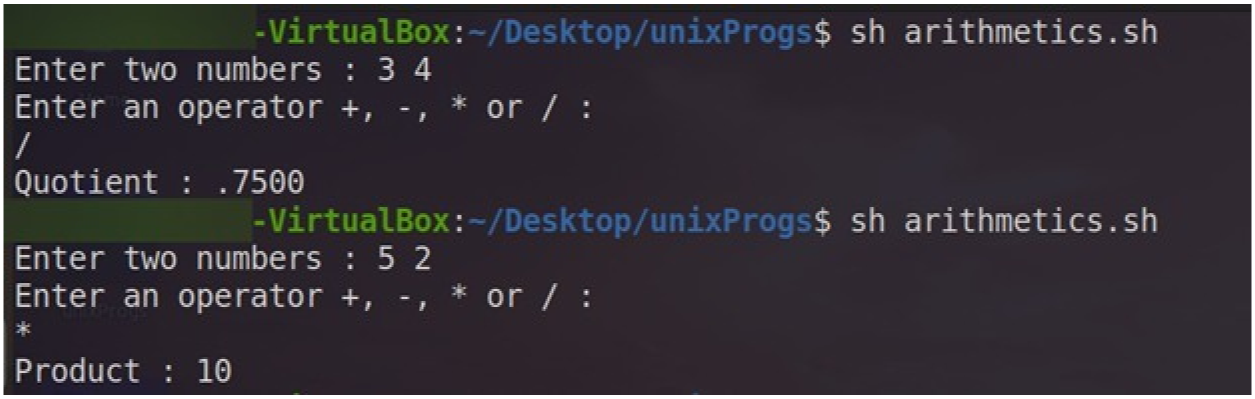
<u>Aim of the project -</u> Shell program to do basic arithmetic operations using CASE.

<u>Program</u>

```
#!/bin/bash

echo -n "Enter two numbers : "

read x y

echo "Enter an operator +, -, * or / : "

read sign
```

```
echo "scale=4; $x $sign $y" | bc
```

Output Screenshot



```
[redacted]-VirtualBox:~/Desktop/unixProgs$ sh arithmetics.sh
Enter two numbers : 3 4
Enter an operator +, -, * or / :
/
Quotient : .7500
[redacted]-VirtualBox:~/Desktop/unixProgs$ sh arithmetics.sh
Enter two numbers : 5 2
Enter an operator +, -, * or / :
*
Product : 10
```

Program 11

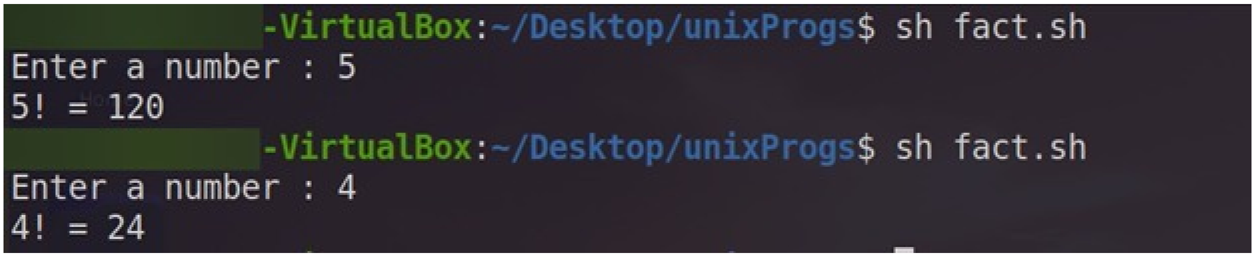
Aim of the program - Shell script to find the factorial of a number

Program

```
#!/bin/bash
echo -n "Enter a number : "
read num
num_copy=$num
ans=1
while [ $num -gt 0 ]
```

```
do
        ans=`expr $ans \* $num `
        num=`expr $num - 1`
done
echo "$num_copy! = $ans"
```

<u>Output Screenshot</u>



```
[redacted]-VirtualBox:~/Desktop/unixProgs$ sh fact.sh
Enter a number : 5
5! = 120
[redacted]-VirtualBox:~/Desktop/unixProgs$ sh fact.sh
Enter a number : 4
4! = 24
```

Program 12

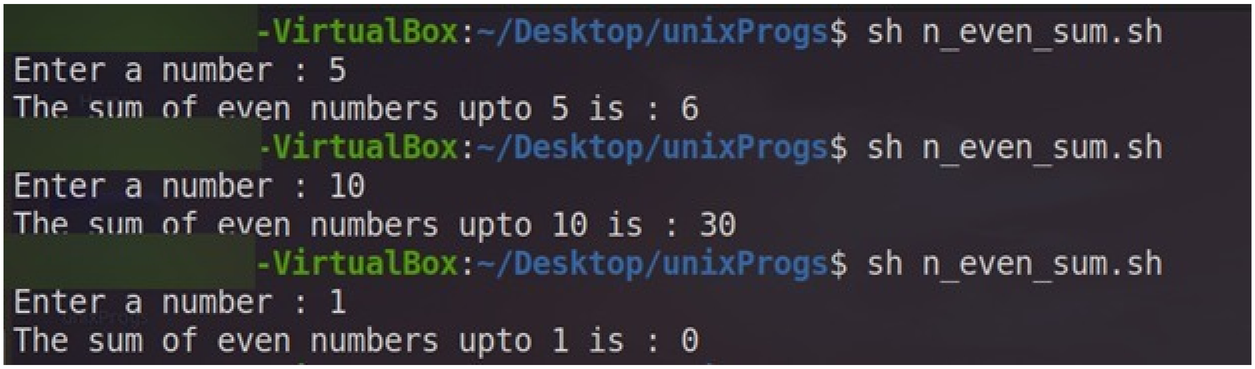
<u>Aim of the program</u> - Shell script to print the sum of even numbers

<u>Program</u>

```
#!/bin/bash
echo -n "Enter a number : "
read n
sum=0
n2=$n
if [ $(( $n % 2 )) -ne 0 ]
```

```
then
        n=$(($n-1))
fi
while [ $n -gt 0 ]
do
        sum=$(( $sum + $n ))
        n=$(( $n - 2 ))
done
echo "The sum of even numbers up to $n2 is : $sum"
```

Output Screenshot



```
[redacted]-VirtualBox:~/Desktop/unixProgs$ sh n_even_sum.sh
Enter a number : 5
The sum of even numbers upto 5 is : 6
[redacted]-VirtualBox:~/Desktop/unixProgs$ sh n_even_sum.sh
Enter a number : 10
The sum of even numbers upto 10 is : 30
[redacted]-VirtualBox:~/Desktop/unixProgs$ sh n_even_sum.sh
Enter a number : 1
The sum of even numbers upto 1 is : 0
```

Program 13

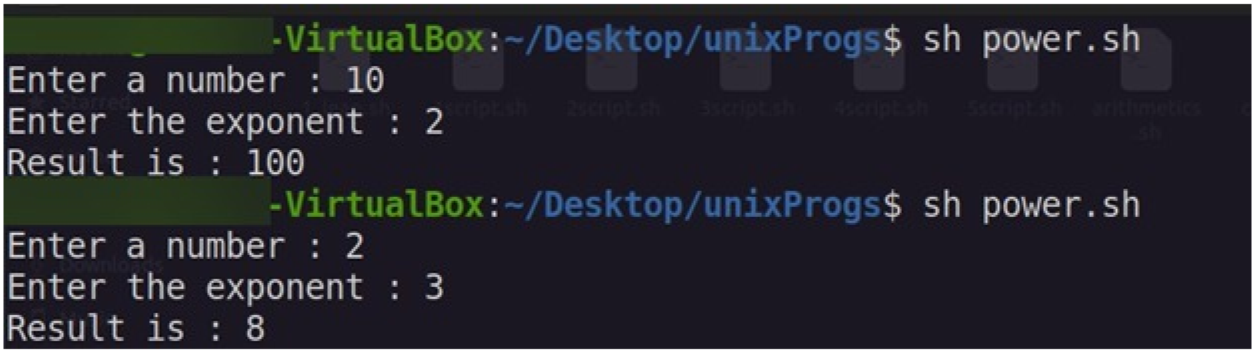
Aim of the Program - Shell script to find the power of a given number

Program

```
#!/bin/bash
echo -n "Enter a number : "
read n
echo -n "Enter the exponent : "
read exp
res=1
```

```
while [ $exp -gt 0 ]

do

        res=`expr $res \* $n`

        exp=`expr $exp - 1`

done

echo "Result is : $res"
```

Output Screenshot



```
[redacted]-VirtualBox:~/Desktop/unixProgs$ sh power.sh
Enter a number : 10
Enter the exponent : 2
Result is : 100
[redacted]-VirtualBox:~/Desktop/unixProgs$ sh power.sh
Enter a number : 2
Enter the exponent : 3
Result is : 8
```

Program 14

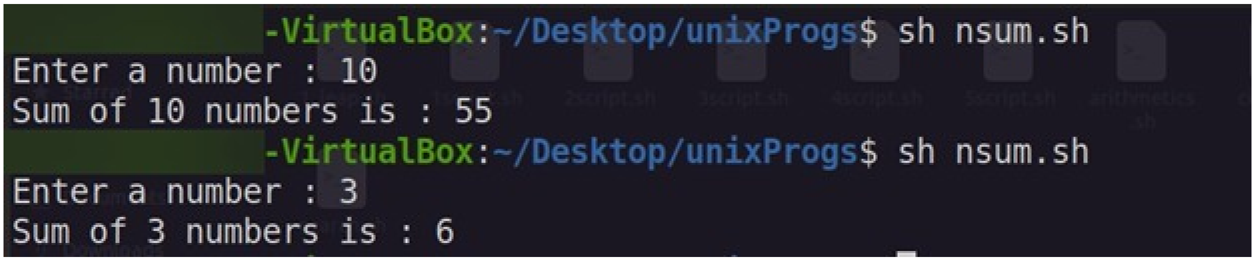
Aim of the program - Shell script to find the sum of N numbers up to N numbers

Program

```
#!/bin/bash

echo -n "Enter a number : "

read num

ans=0

i=$num
```

```
while [ $i -gt 0 ]
do
        ans=`expr $i + $ans`
        i=`expr $i - 1`
done
echo "Sum of $num numbers is : $ans"
```

Output Screenshot



```
[redacted]-VirtualBox:~/Desktop/unixProgs$ sh nsum.sh
Enter a number : 10
Sum of 10 numbers is : 55
[redacted]-VirtualBox:~/Desktop/unixProgs$ sh nsum.sh
Enter a number : 3
Sum of 3 numbers is : 6
```

Program 15

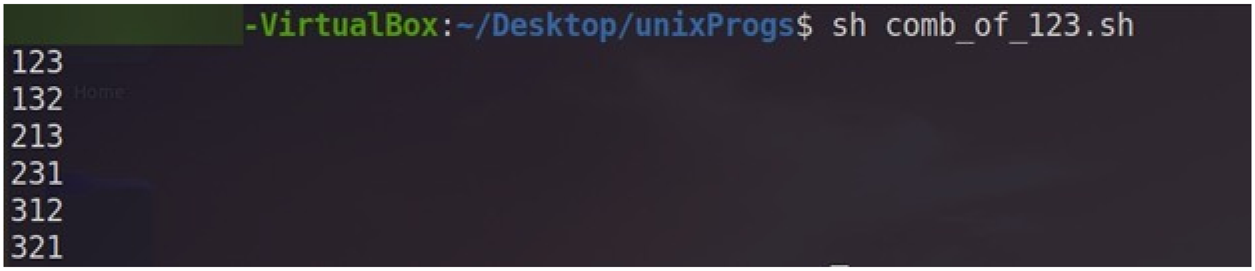
Aim of the Program - Shell program to find all the combinations of the numbers ‘1 2 3’

Program

```
#!/bin/bash
for i in 1 2 3;do
        for j in 1 2 3;do
                for k in 1 2 3;do
                                if [ $i -ne $j ] && [ $j -ne $k ] && [ $k -ne $i ]
```

```
                        then
                        echo "$i$j$k"
                        fi
                done
        done
done
```

Output Screenshot



```
[redacted]-VirtualBox:~/Desktop/unixProgs$ sh comb_of_123.sh
123
132
213
231
312
321
```

Program 16

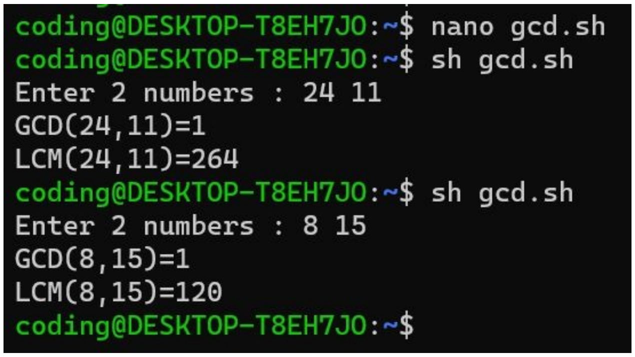
<u>Aim of Program</u> - Shell program to find GCD and LCM of a number

<u>Program</u>

```
#!/bin/bash
echo -n "Enter 2 numbers : "
read num1 num2
n1=$num1
n2=$num2
#echo $n1 $n2
```

```
rem=`expr $num1 % $num2`
while [ $rem -ne 0 ]
do
        rem=`expr $num1 % $num2`
        num1=$num2
        num2=$rem
done
lcm=$(($n1*$n2/$num1))
echo "GCD($n1,$n2)=$num1"
echo "LCM($n1,$n2)=$lcm"
```

Output Screenshot



```
coding@DESKTOP-T8EH7JO:~$ nano gcd.sh
coding@DESKTOP-T8EH7JO:~$ sh gcd.sh
Enter 2 numbers : 24 11
GCD(24,11)=1
LCM(24,11)=264
coding@DESKTOP-T8EH7JO:~$ sh gcd.sh
Enter 2 numbers : 8 15
GCD(8,15)=1
LCM(8,15)=120
coding@DESKTOP-T8EH7JO:~$
```

Program 17

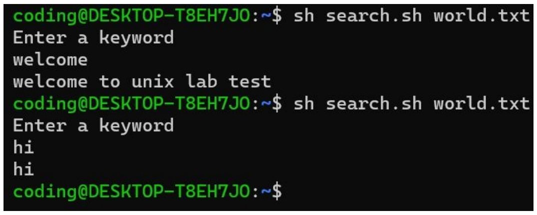
Aim of the Program - Shell script that takes a filename as arguments and searches for a specific word on the file and stops when the word is found

Program

```
#!/bin/bash
if [ $# -lt 2 ]
then
        echo "Insufficient parameters"
else
```

```
    pattern=$1
    shift
    for filename in "$@"
    do
        grep "$pattern" $filename
        if [ $? -eq 0 ]
        then
            echo "Pattern found in $filename"
            exit
        fi
    done
fi
```

Output Screenshot



```
coding@DESKTOP-T8EH7JO:~$ sh search.sh world.txt
Enter a keyword
welcome
welcome to unix lab test
coding@DESKTOP-T8EH7JO:~$ sh search.sh world.txt
Enter a keyword
hi
hi
coding@DESKTOP-T8EH7JO:~$
```

Program 18

Aim of the Program - Shell script to find the number of lines, words and characters

Program

```
#!/bin/bash
echo "Enter the name of the file:" \c
read name
echo "The number of lines are:"
wc -l $name
```

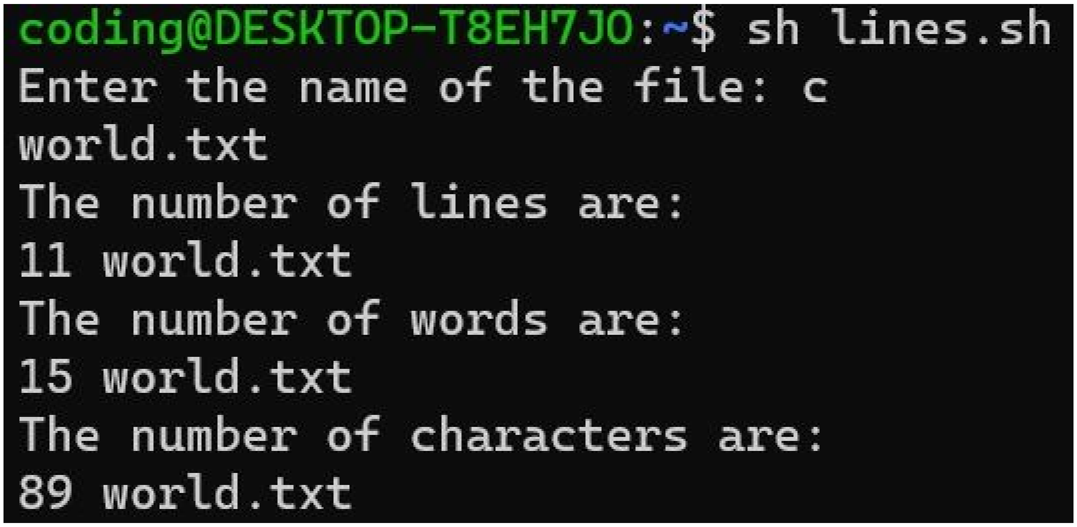
echo "The number of words are:"

wc -w $name

echo "The number of characters are:"

wc -c $name

<u>Output Screenshot</u>



```
coding@DESKTOP-T8EH7JO:~$ sh lines.sh
Enter the name of the file: c
world.txt
The number of lines are:
11 world.txt
The number of words are:
15 world.txt
The number of characters are:
89 world.txt
```

Program 19

<u>Aim of the Program</u> - Shell program to enter N numbers and display it's count, sum of positive and negative numbers separately

<u>Program</u>

#!/bin/bash

echo -n "Enter the value of n : "

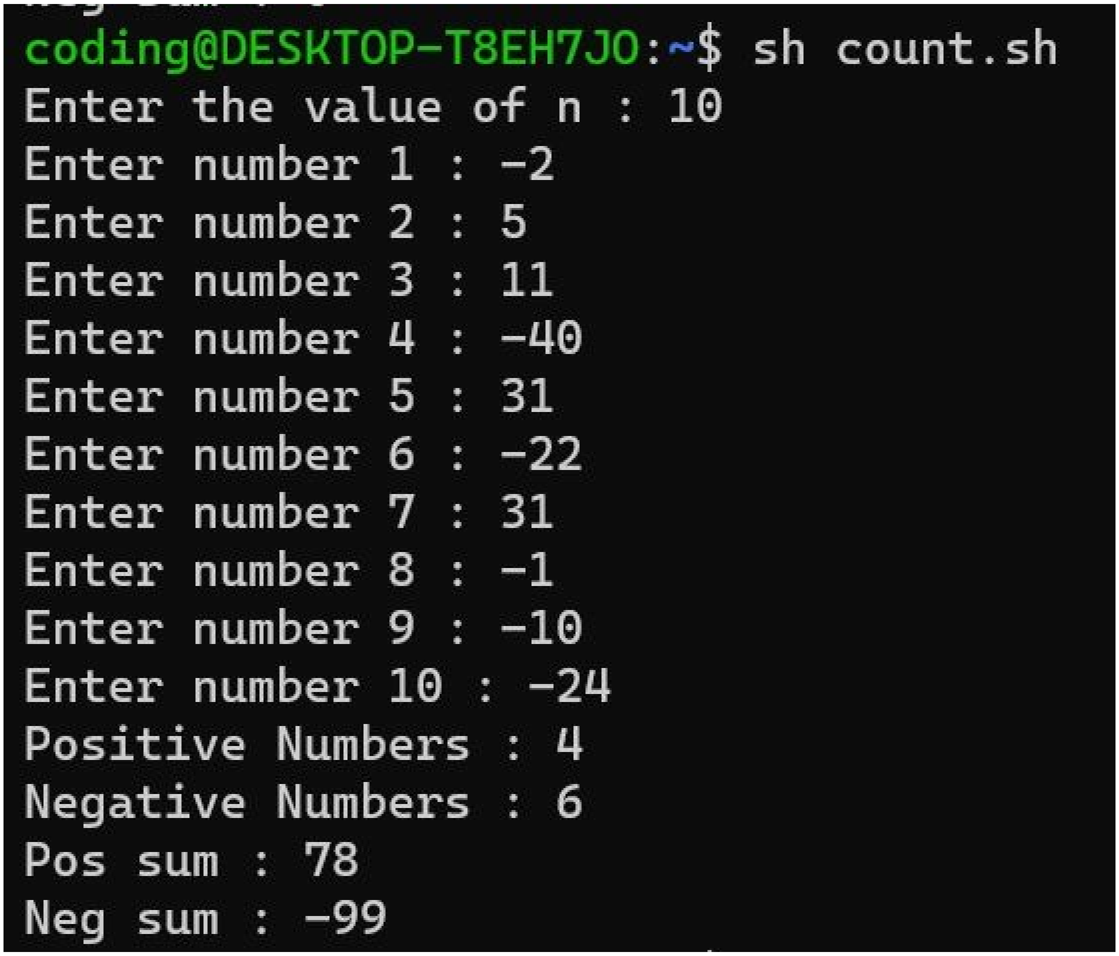
read n

n_cp=$n

pos_count=0

```
neg_count=0
neg_total=0
pos_total=0
while [ $n -gt 0 ]
do
    echo -n "Enter number `expr $n_cp - $n + 1` : "
    read cur
    if [ $cur -gt 0 ]
    then
        pos_count=$(($pos_count+1))
        pos_total=$(($pos_total+$cur))
    elif [ $cur -lt 0 ]
    then
        neg_count=$(($neg_count+1))
        neg_total=$(($neg_total+$cur))
fi
    n=$(($n-1))
done
echo  "Positive Numbers : $pos_count "
echo  "Negative Numbers : $neg_count "
echo  "Pos sum : $pos_total "
echo  "Neg sum : $neg_total "
```

Output Screenshot



```
coding@DESKTOP-T8EH7JO:~$ sh count.sh
Enter the value of n : 10
Enter number 1 : -2
Enter number 2 : 5
Enter number 3 : 11
Enter number 4 : -40
Enter number 5 : 31
Enter number 6 : -22
Enter number 7 : 31
Enter number 8 : -1
Enter number 9 : -10
Enter number 10 : -24
Positive Numbers : 4
Negative Numbers : 6
Pos sum : 78
Neg sum : -99
```

Program 20

Aim of the Program - Shell script to find the sum of last two prime numbers before the entered N number

Program

#!/bin/bash

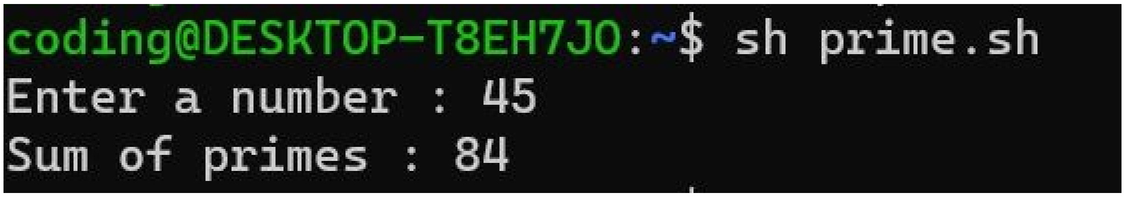
echo -n "Enter a number : "

```
read n
count=0
prime_sum=0
while [ $n -gt 2 ] && [ $count -lt 2 ]
do
    flag=1
    n_copy=$n
    divider=$(($n_copy-1))
    while [ $divider -ge 2 ]
    do
        if [ $(($n_copy % $divider)) -eq 0 ]
        then
#           echo "$n_copy $divider"
            flag=0
            break
        fi
        divider=$(($divider-1))
    done
    if [ $flag -eq 1 ]
    then
#       echo "$n_copy"
        prime_sum=$(($prime_sum+$n_copy))
        count=$(($count+1))
    fi
    n=$(($n-1))
```

done

echo "Sum of primes : $prime_sum"

Output Screenshot



```
coding@DESKTOP-T8EH7JO:~$ sh prime.sh
Enter a number : 45
Sum of primes : 84
```

Program 21

Aim of the Program - Shell script to form a diamond like pattern for entered N number

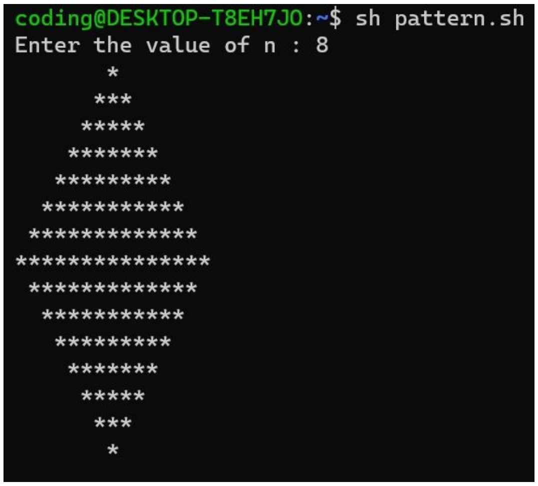
Program

#!/bin/bash

```
echo -n "Enter the value of n : "
read n
stars=1
n_cp=$n
while [ $n -gt 0 ]
do
        spaces=$(($n-1))
        star2=$stars
        while [ $spaces -gt 0 ]
        do
                echo -n " "
                spaces=$(($spaces-1))
        done
        while [ $star2 -gt 0 ]
        do
                echo -n "*"
                star2=$(($star2-1))
        done
        echo " "
        n=$((n-1))
        stars=$((stars+2))
done
stars=$((stars-2))
spaces=1
while [ $n_cp -gt 0 ]
```

```
do
        spaces2=$spaces
        stars=$((stars-2))
        star2=$((stars))
        while [ $spaces2 -gt 0 ]
        do
                echo -n " "
                spaces2=$((spaces2-1))
        done
        while [ $star2 -gt 0 ]
        do
                echo -n "*"
                star2=$((star2-1))
        done
        echo " "
        n_cp=$((n_cp-1))
        spaces=$((spaces+1))
done
```

Output Screenshot



```
coding@DESKTOP-T8EH7JO:~$ sh pattern.sh
Enter the value of n : 8
       *
      ***
     *****
    *******
   *********
  ***********
 *************
***************
 *************
  ***********
   *********
    *******
     *****
      ***
       *
```

SYSTEM PROGRAMS

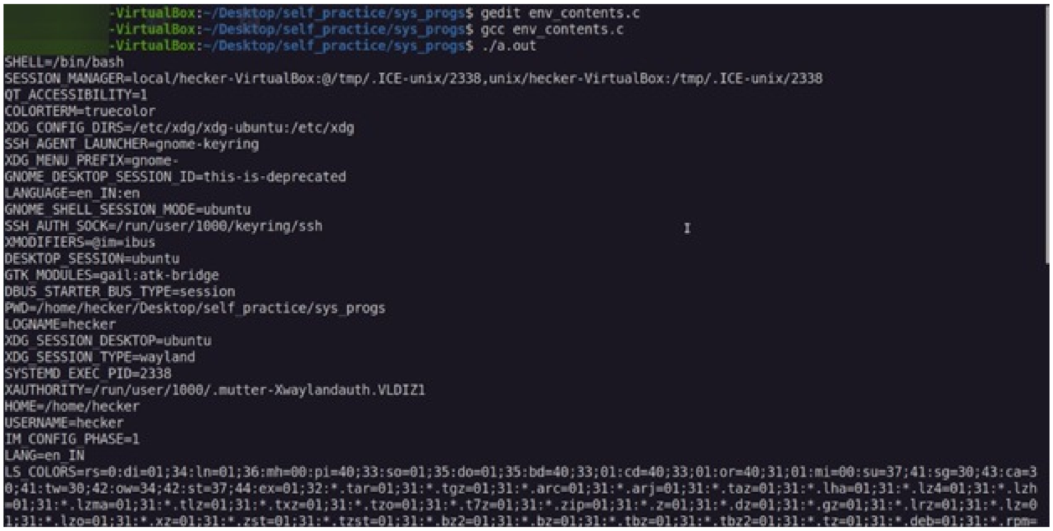
Program 1

<u>Aim of the Program</u> - Write a C/C++ program to that outputs the contents of its Environment list

<u>Program</u>

```
#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;

}
```

<u>Output Screenshot</u>



```
[redacted]-VirtualBox:~/Desktop/self_practice/sys_progs$ gedit env_contents.c
          -VirtualBox:~/Desktop/self_practice/sys_progs$ gcc env_contents.c
          -VirtualBox:~/Desktop/self_practice/sys_progs$ ./a.out
SHELL=/bin/bash
SESSION_MANAGER=local/hecker-VirtualBox:@/tmp/.ICE-unix/2338,unix/hecker-VirtualBox:/tmp/.ICE-unix/2338
QT_ACCESSIBILITY=1
COLORTERM=truecolor
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/etc/xdg
SSH_AGENT_LAUNCHER=gnome-keyring
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
DBUS_STARTER_BUS_TYPE=session
PWD=/home/hecker/Desktop/self_practice/sys_progs
LOGNAME=hecker
XDG_SESSION_DESKTOP=ubuntu
XDG_SESSION_TYPE=wayland
SYSTEMD_EXEC_PID=2338
XAUTHORITY=/run/user/1000/.mutter-Xwaylandauth.VLDIZ1
HOME=/home/hecker
USERNAME=hecker
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=37;41:sg=30;43:ca=3
0;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:*.lha=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=0
1;31:*.lzo=01;31:*.xz=01;31:*.zst=01;31:*.tzst=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.tz=01;31:*.deb=01;31:*.rpm=
```

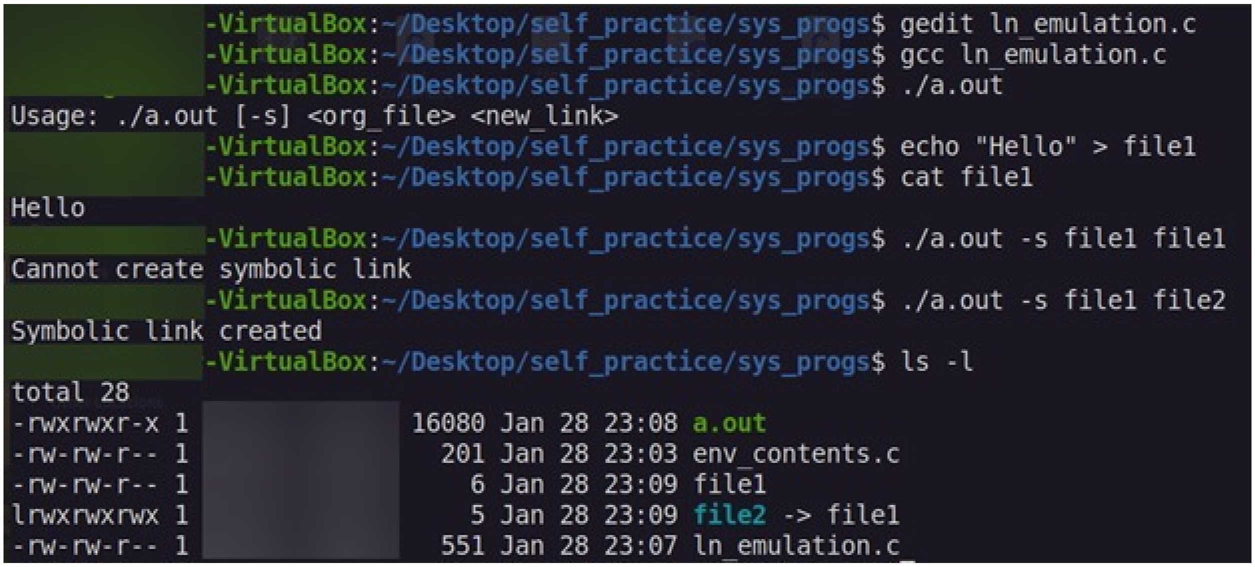
Program 2

Aim of the program - To emulate the unix line 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;
}
```

Output Screenshot



```
[redacted]-VirtualBox:~/Desktop/self_practice/sys_progs$ gedit ln_emulation.c
[redacted]-VirtualBox:~/Desktop/self_practice/sys_progs$ gcc ln_emulation.c
[redacted]-VirtualBox:~/Desktop/self_practice/sys_progs$ ./a.out
Usage: ./a.out [-s] <org_file> <new_link>
[redacted]-VirtualBox:~/Desktop/self_practice/sys_progs$ echo "Hello" > file1
[redacted]-VirtualBox:~/Desktop/self_practice/sys_progs$ cat file1
Hello
[redacted]-VirtualBox:~/Desktop/self_practice/sys_progs$ ./a.out -s file1 file1
Cannot create symbolic link
[redacted]-VirtualBox:~/Desktop/self_practice/sys_progs$ ./a.out -s file1 file2
Symbolic link created
[redacted]-VirtualBox:~/Desktop/self_practice/sys_progs$ ls -l
total 28
-rwxrwxr-x 1 [redacted] 16080 Jan 28 23:08 a.out
-rw-rw-r-- 1 [redacted]   201 Jan 28 23:03 env_contents.c
-rw-rw-r-- 1 [redacted]     6 Jan 28 23:09 file1
lrwxrwxrwx 1 [redacted]     5 Jan 28 23:09 file2 -> file1
-rw-rw-r-- 1 [redacted]   551 Jan 28 23:07 ln_emulation.c
```

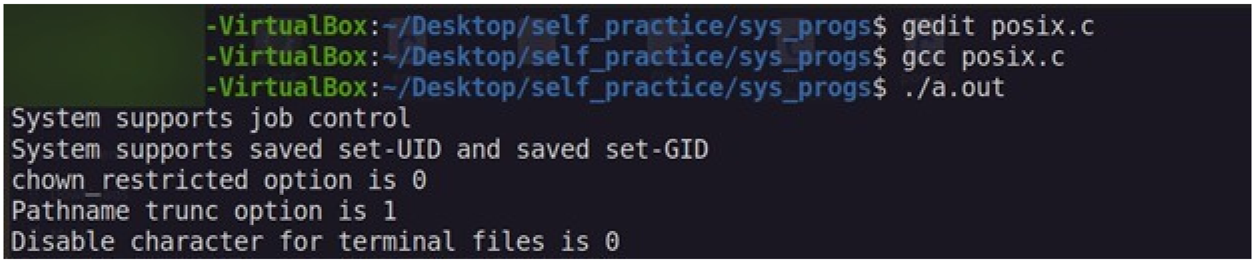
Program 3

Aim of the Program - A POSIX compliant program that prints POSIX defined configurations option supported in any given system using feature test macro.

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 Screenshot



```
[redacted]-VirtualBox:~/Desktop/self_practice/sys_progs$ gedit posix.c
-VirtualBox:~/Desktop/self_practice/sys_progs$ gcc posix.c
-VirtualBox:~/Desktop/self_practice/sys_progs$ ./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
```

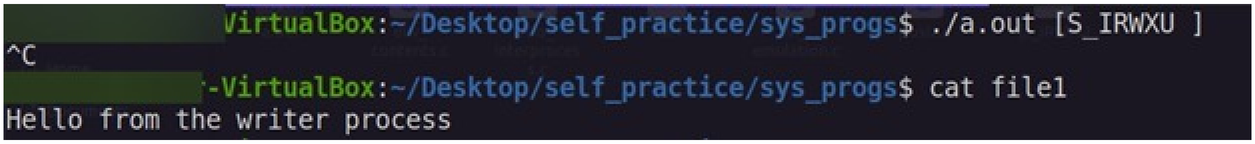
Program 4

Aim of the Program - Write a program which demonstrates interprocess communication between a reader and a write process

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)) > 0)
                printf("%s",buf);
        }
        else
        {
                fd = open(argv[1], O_WRONLY);
                write(fd,argv[2],strlen(argv[2]));
        }
        close(fd);
}
```

<u>Output Screenshot</u>



```
[redacted]VirtualBox:~/Desktop/self_practice/sys_progs$ ./a.out [S_IRWXU ]
^C
[redacted]-VirtualBox:~/Desktop/self_practice/sys_progs$ cat file1
Hello from the writer process
```