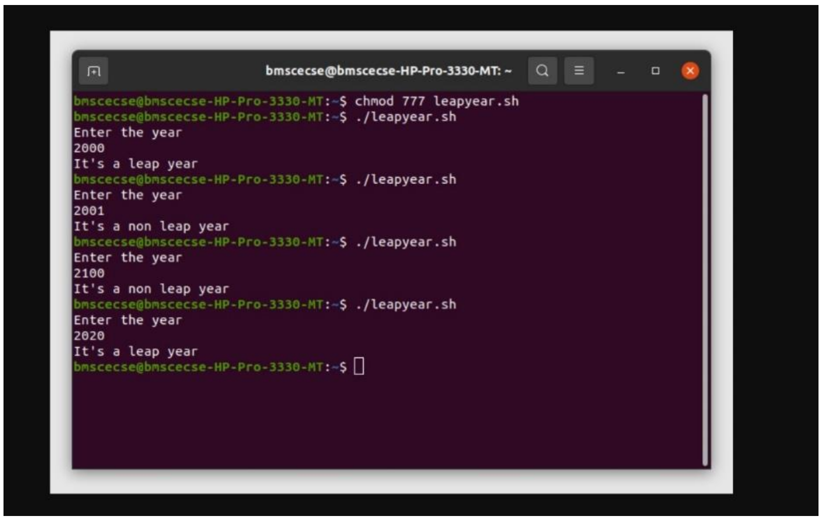
# LAB PROGRAMS:

**PROGRAM1: Shell script to find if the given year is leap or not**

CODE:

```
#!/bin/sh
echo "Enter the year "
read year
if [ $((year%400)) -eq 0 ]
then
        echo "It's a leap year"
elif [ $((year%4)) -eq 0 ]
then
        if [ $((year%100)) -eq 0 ]
        then
                echo "It's a non leap year"
        else
                echo "It's a leap year "
        fi
else
        echo "It's a non leap year "
fi
```

OUTPUT:



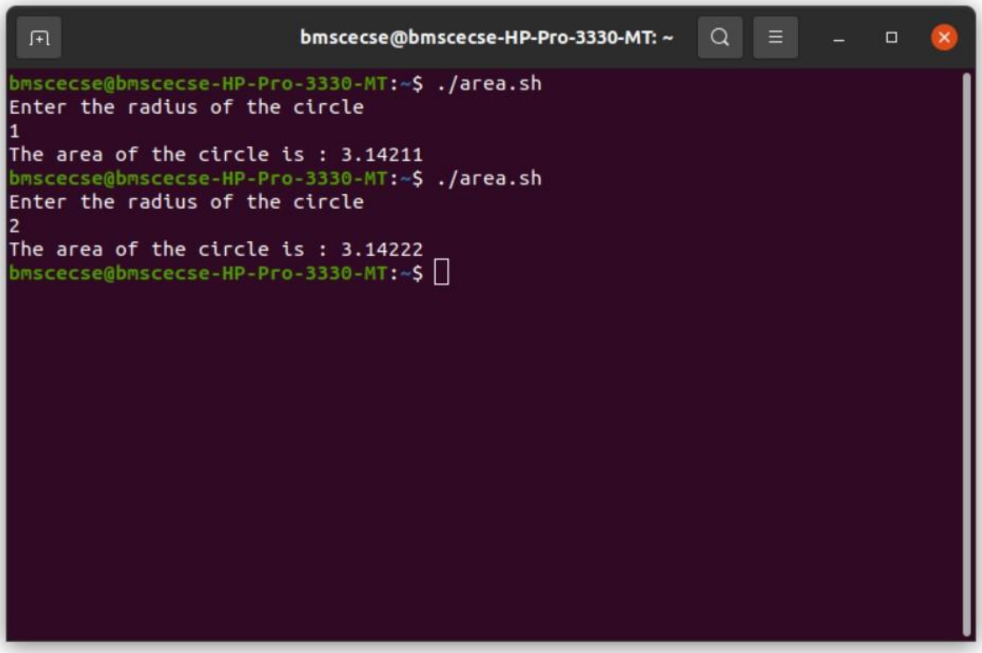
```
bmscecse@bmscecse-HP-Pro-3330-MT:~$ chmod 777 leapyear.sh
bmscecse@bmscecse-HP-Pro-3330-MT:~$ ./leapyear.sh
Enter the year
2000
It's a leap year
bmscecse@bmscecse-HP-Pro-3330-MT:~$ ./leapyear.sh
Enter the year
2001
It's a non leap year
bmscecse@bmscecse-HP-Pro-3330-MT:~$ ./leapyear.sh
Enter the year
2100
It's a non leap year
bmscecse@bmscecse-HP-Pro-3330-MT:~$ ./leapyear.sh
Enter the year
2020
It's a leap year
bmscecse@bmscecse-HP-Pro-3330-MT:~$ 
```

**<u>PROGRAM2:</u> Shell script to find the area of a circle**

CODE:

```
#!/bin/sh

echo "Enter the radius of the circle "

read r

pi=3.142

area=`echo $pi\$r\$r|bc`

echo "The area of the circle is : $area"
```

OUTPUT:



```
bmscecse@bmscecse-HP-Pro-3330-MT:~$ ./area.sh
Enter the radius of the circle
1
The area of the circle is : 3.14211
bmscecse@bmscecse-HP-Pro-3330-MT:~$ ./area.sh
Enter the radius of the circle
2
The area of the circle is : 3.14222
bmscecse@bmscecse-HP-Pro-3330-MT:~$
```

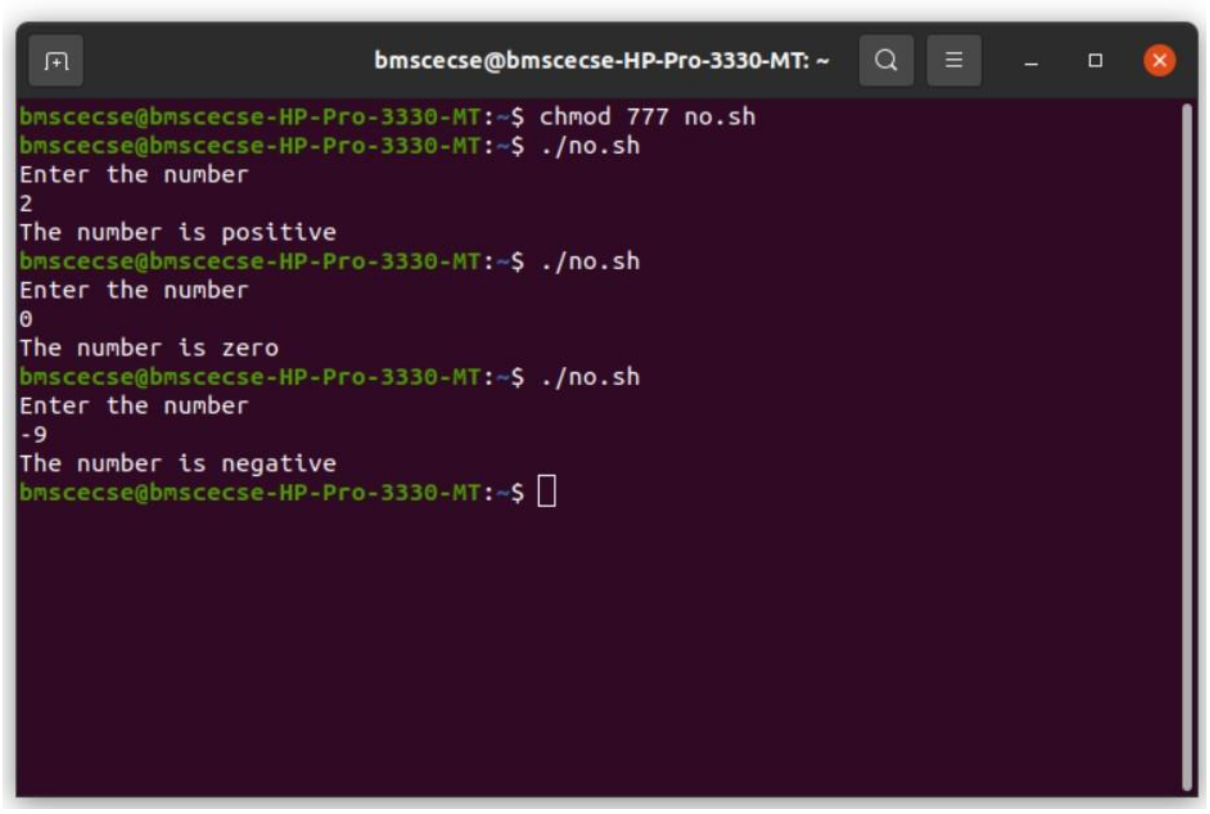
**<u>PROGRAM3:</u> <u>Shell script to check whether the number is zero/ positive/ negative</u>**

CODE:

```
#!/bin/sh

echo "Enter the number "

read num

if [ $num -eq 0 ]

then

        echo "The number is zero "
```

```
elif [ $num -lt 0 ]
then
        echo "The number is negative "
else
        echo "The number is positive"
fi
```

OUTPUT:



```
bmscecse@bmscecse-HP-Pro-3330-MT:~$ chmod 777 no.sh
bmscecse@bmscecse-HP-Pro-3330-MT:~$ ./no.sh
Enter the number
2
The number is positive
bmscecse@bmscecse-HP-Pro-3330-MT:~$ ./no.sh
Enter the number
0
The number is zero
bmscecse@bmscecse-HP-Pro-3330-MT:~$ ./no.sh
Enter the number
-9
The number is negative
bmscecse@bmscecse-HP-Pro-3330-MT:~$
```

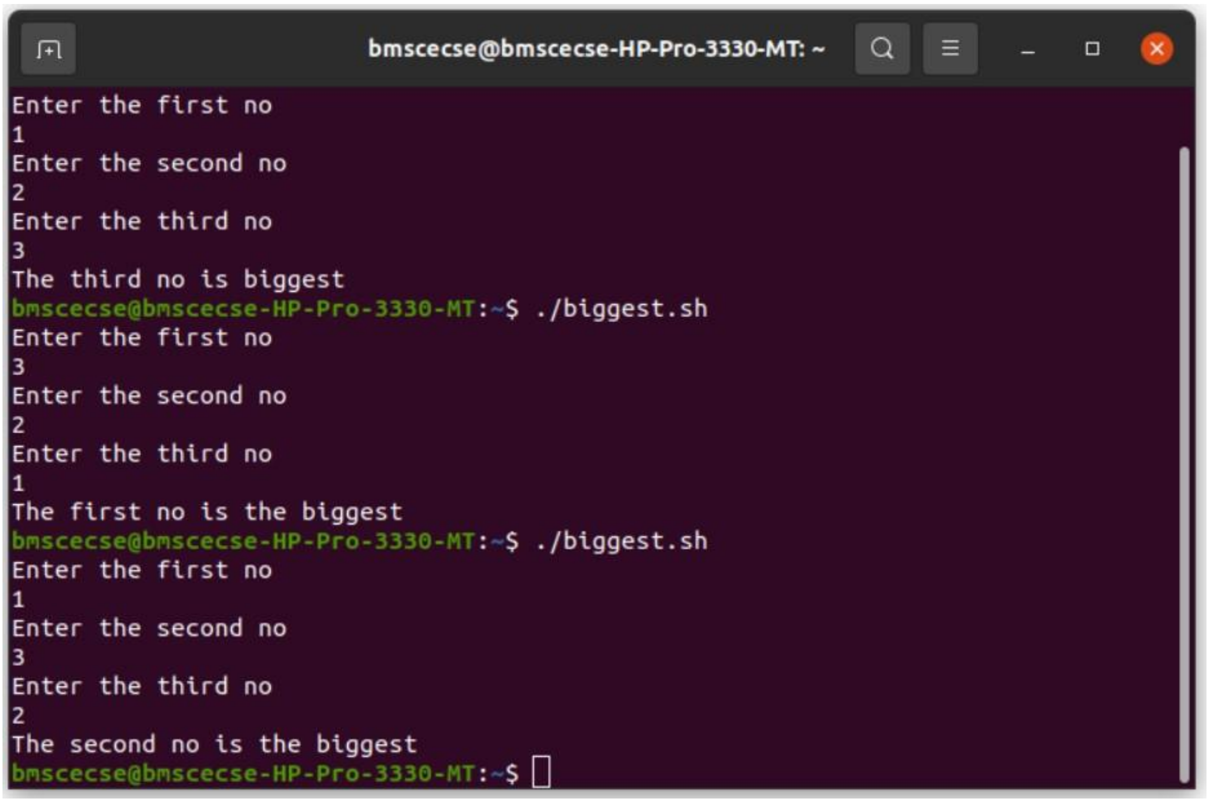
**<u>PROGRAM4:</u> Shell script to find the biggest of three numbers**

CODE:

```
#!/bin/sh
echo "Enter the first no "
read f
echo "Enter the second no"
read s
echo "Enter the third no "
read t
```

```
if [ $f -gt $s -a $f -gt $t ]
then
        echo "The first no is the biggest "
elif [ $s -gt $f -a $s -gt $t ]
then
        echo "The second no is the biggest "
else
        echo "The third no is biggest"
fi
```

OUTPUT:



```
Enter the first no
1
Enter the second no
2
Enter the third no
3
The third no is biggest
bmscecse@bmscecse-HP-Pro-3330-MT:~$ ./biggest.sh
Enter the first no
3
Enter the second no
2
Enter the third no
1
The first no is the biggest
bmscecse@bmscecse-HP-Pro-3330-MT:~$ ./biggest.sh
Enter the first no
1
Enter the second no
3
Enter the third no
2
The second no is the biggest
bmscecse@bmscecse-HP-Pro-3330-MT:~$
```

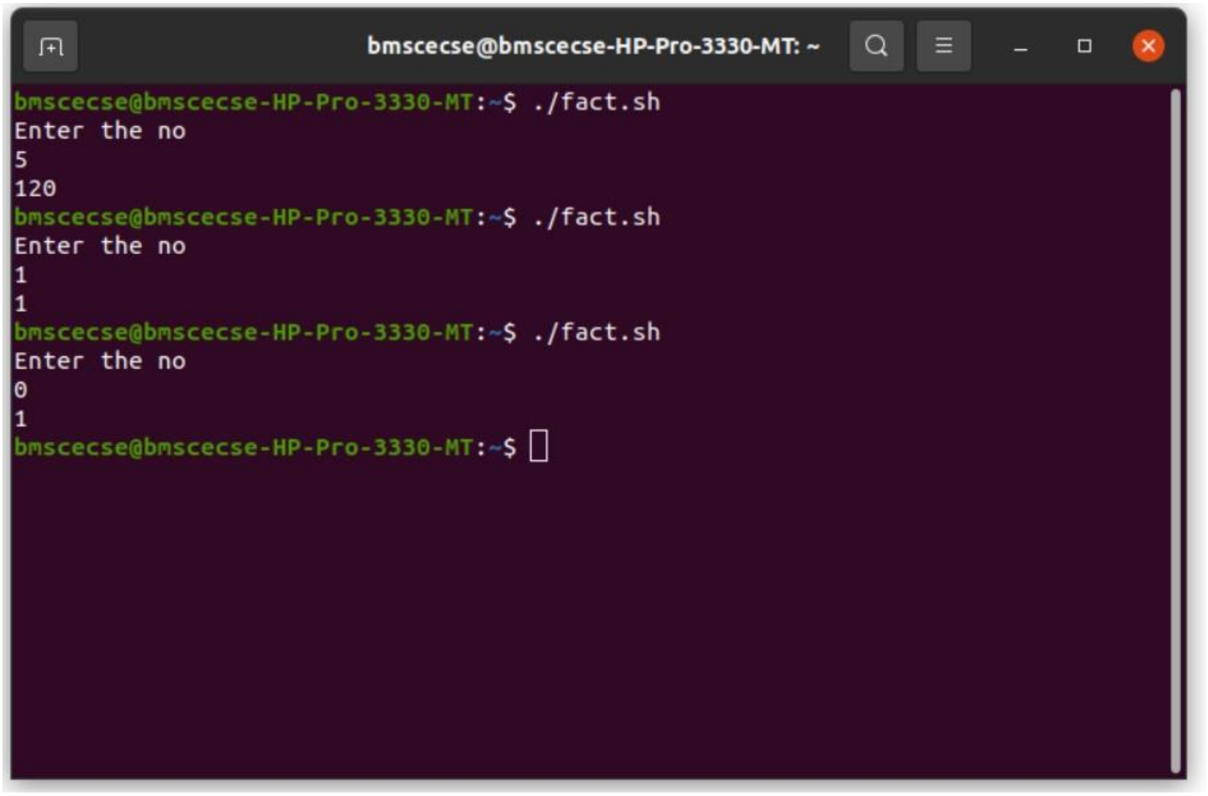
**<u>PROGRAM5:</u> Shell script to find the factorial of a number**

CODE:

```
#!/bin/bash
echo "Enter the no "
read no
st=1
```

```
fact=1
for (( c=$st; c<=$no; c++))
do
        fact=`expr $c\*$fact|bc`
done
echo "factorial is "$fact
```

OUTPUT:



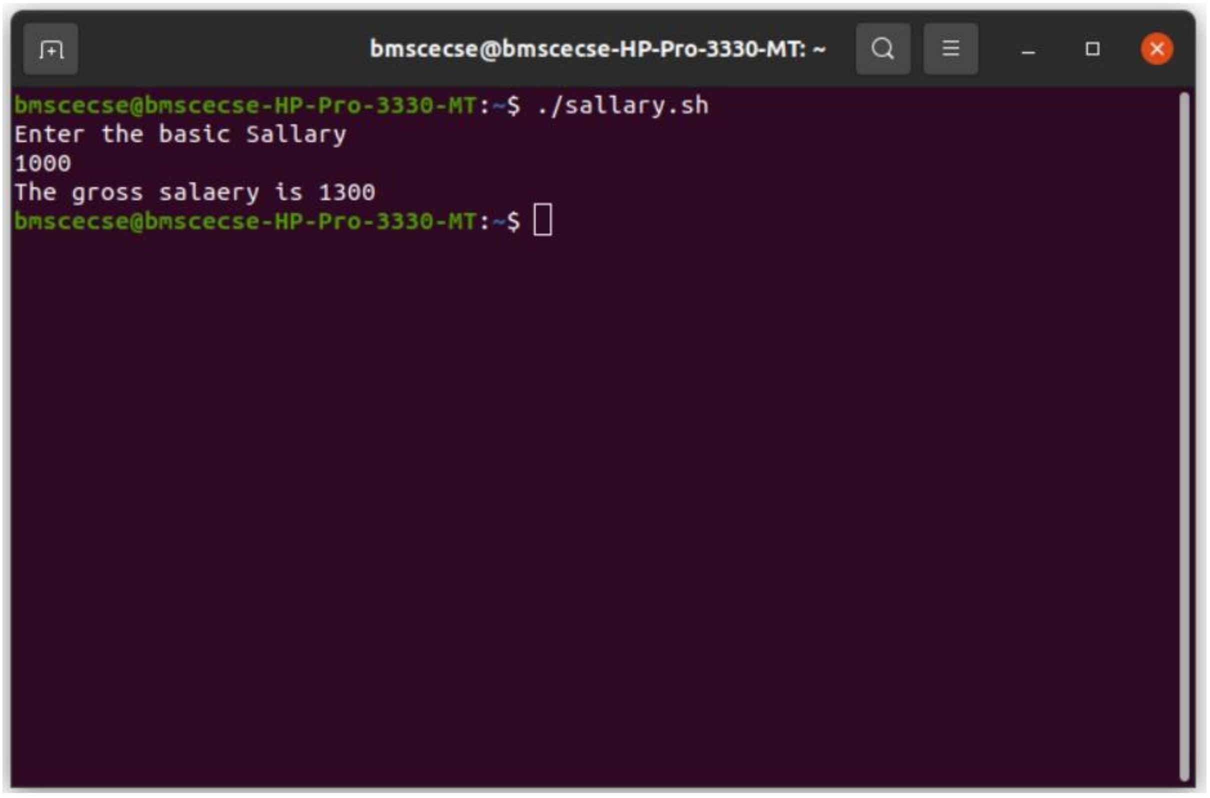
```
bmscecse@bmscecse-HP-Pro-3330-MT:~$ ./fact.sh
Enter the no
5
120
bmscecse@bmscecse-HP-Pro-3330-MT:~$ ./fact.sh
Enter the no
1
1
bmscecse@bmscecse-HP-Pro-3330-MT:~$ ./fact.sh
Enter the no
0
1
bmscecse@bmscecse-HP-Pro-3330-MT:~$
```

**<u>PROGRAM6:</u> Shell script to compute the gross salary of an employee**

CODE:

```
#!/bin/sh
echo "Enter the basic Sallary"
read basic
da=`expr $basic\*10/100|bc`
hra=`expr $basic\*20/100|bc`
gross_sal=`expr $basic+$da+$hra|bc`
echo "The gross salaery is "$gross_sal
```

OUTPUT:



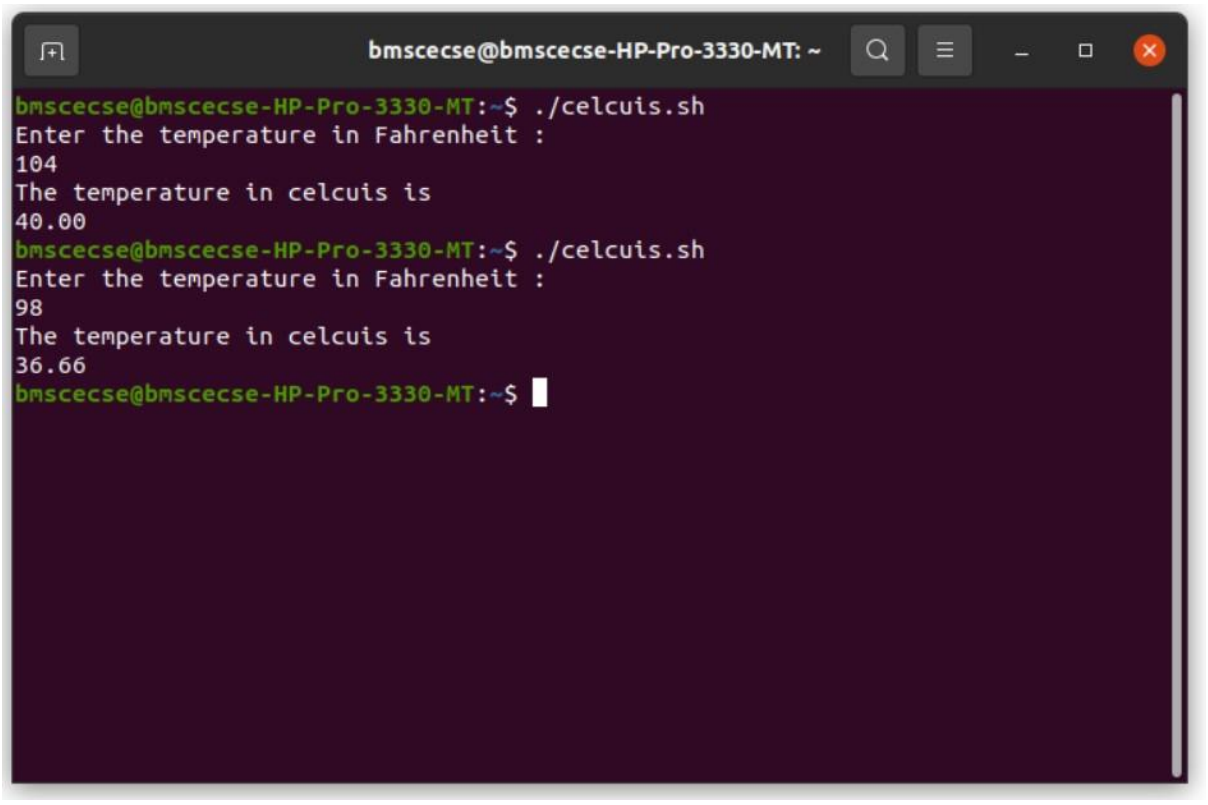
```
bmscecse@bmscecse-HP-Pro-3330-MT:~$ ./sallary.sh
Enter the basic Sallary
1000
The gross salaery is 1300
bmscecse@bmscecse-HP-Pro-3330-MT:~$
```

**<u>PROGRAM7:</u> Shell script to convert the temperature Fahrenheit to Celsius**

CODE:

```
#!/bin/sh

echo "Enter the temperature in Fahrenheit :"

read temp

var=32

f=`expr $temp-$var|bc`

s=`expr $f\*5|bc`

echo "The temperature in celcuis is "

echo "scale=2; $s/9"|bc
```

OUTPUT:



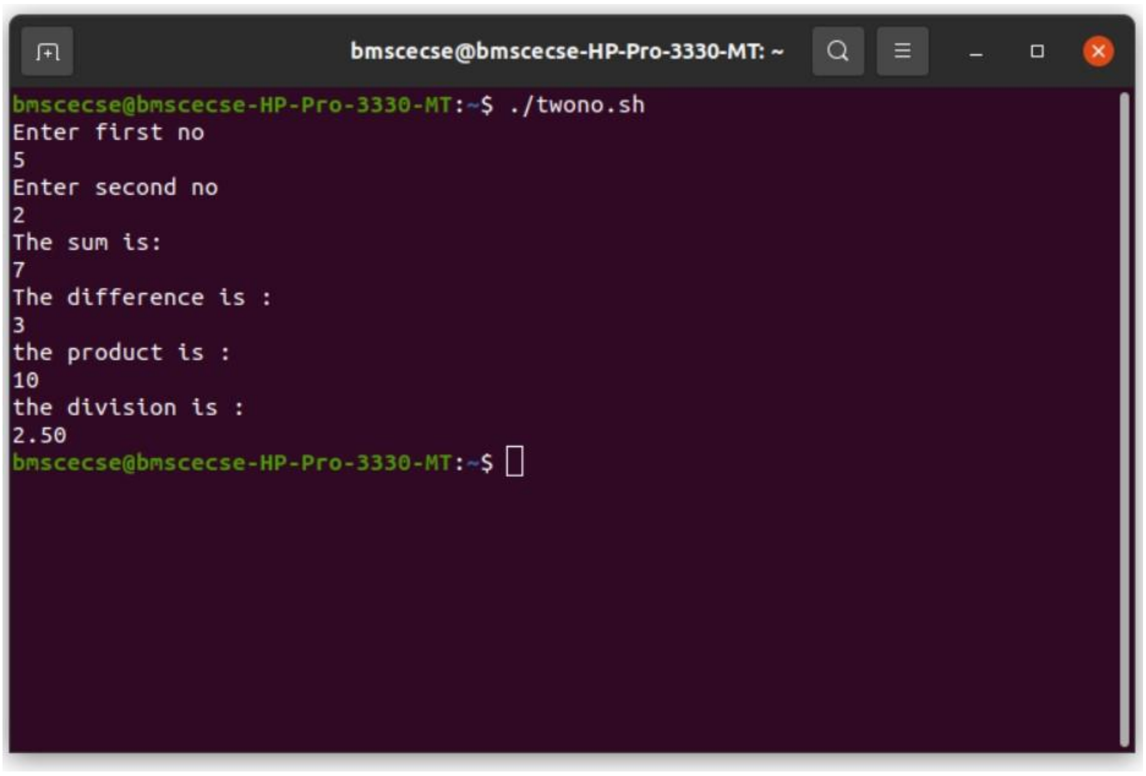
```
bmscecse@bmscecse-HP-Pro-3330-MT:~$ ./celcuis.sh
Enter the temperature in Fahrenheit :
104
The temperature in celcuis is
40.00
bmscecse@bmscecse-HP-Pro-3330-MT:~$ ./celcuis.sh
Enter the temperature in Fahrenheit :
98
The temperature in celcuis is
36.66
bmscecse@bmscecse-HP-Pro-3330-MT:~$
```

**<u>PROGRAM8:</u> Shell script to perform arithmetic operations on given two numbers**

CODE:

```
#!/bin/sh

echo "Enter first no"

read f

echo "Enter second no"

read s

echo "The sum is:"

echo "$f+$s"|bc

echo "The difference is :"

echo "$f-$s"|bc

echo "the product is :"

echo "$f*$s"|bc

echo "the division is :"

echo "scale=2; $f/$s"|bc
```

OUTPUT:



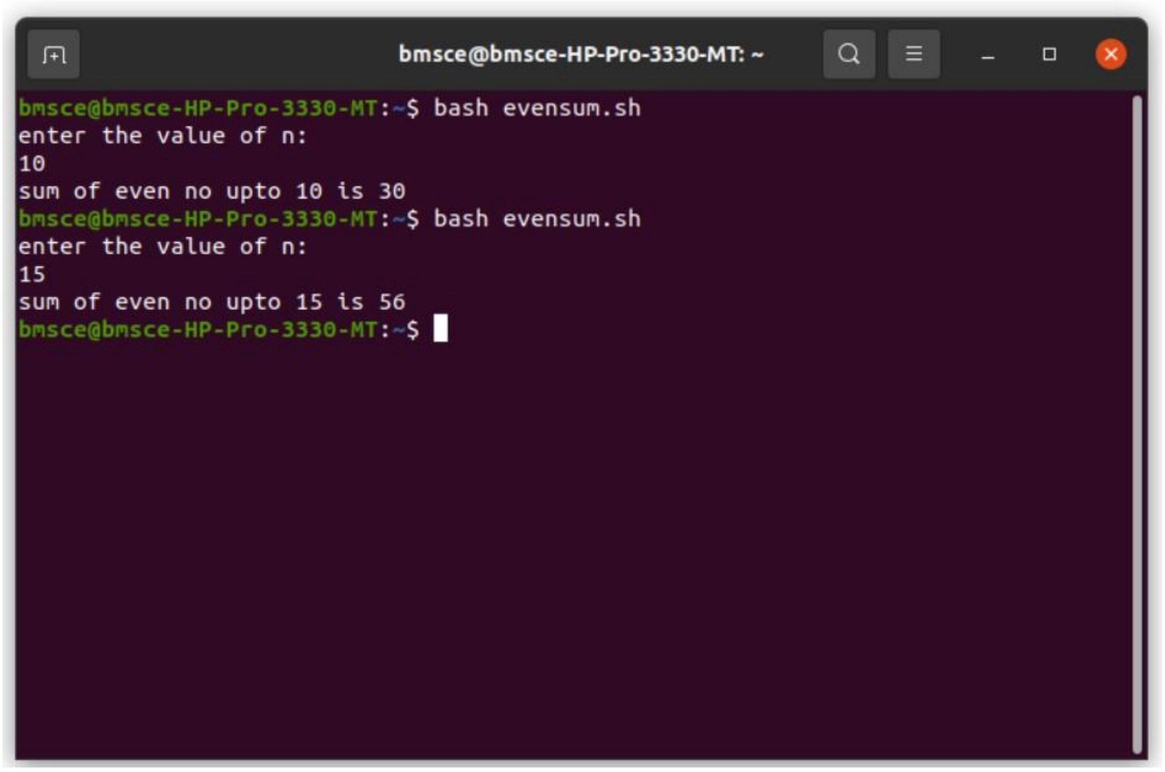
```
bmscecse@bmscecse-HP-Pro-3330-MT:~$ ./twono.sh
Enter first no
5
Enter second no
2
The sum is:
7
The difference is :
3
the product is :
10
the division is :
2.50
bmscecse@bmscecse-HP-Pro-3330-MT:~$
```

**<u>PROGRAM9: </u>Shell script to find the sum of even numbers upto n**

CODE:

```
#!/bin/bash

echo "enter the value of n:"

read n

sum=0

for ((c=0; c<=$n; c=c+2))

do

        sum=`expr $sum+$c|bc`

done

echo "sum of even no upto $n is $sum"
```

OUTPUT:



bmsce@bmsce-HP-Pro-3330-MT: ~

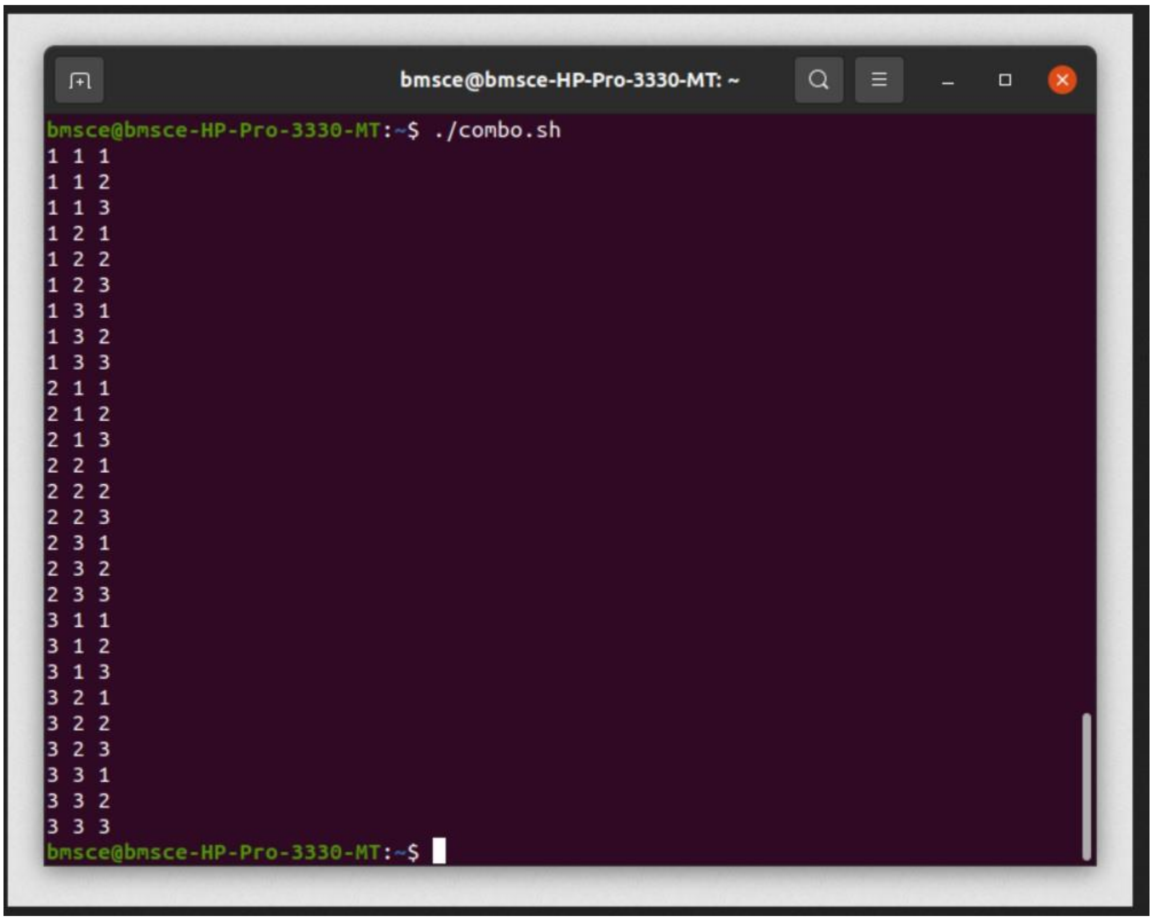
```
bmsce@bmsce-HP-Pro-3330-MT:~$ bash evensum.sh
enter the value of n:
10
sum of even no upto 10 is 30
bmsce@bmsce-HP-Pro-3330-MT:~$ bash evensum.sh
enter the value of n:
15
sum of even no upto 15 is 56
bmsce@bmsce-HP-Pro-3330-MT:~$
```

**<u>PROGRAM10:</u> Shell script to print the combinations of numbers 123**

CODE:

```
#!/bin/sh
for i in 1 2 3
do
        for j in 1 2 3
        do
                for k in 1 2 3
                do
                        echo $i $j $k
                done
        done
done
```

OUTPUT:



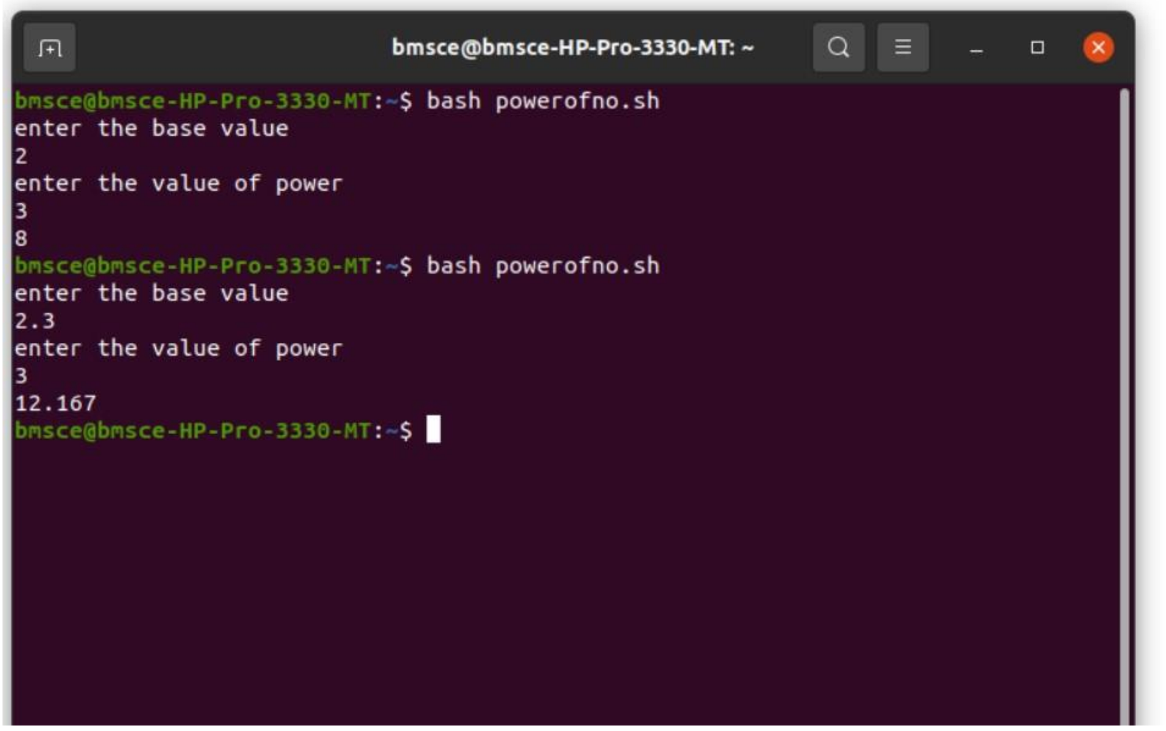
```
bmsce@bmsce-HP-Pro-3330-MT:~$ ./combo.sh
1 1 1
1 1 2
1 1 3
1 2 1
1 2 2
1 2 3
1 3 1
1 3 2
1 3 3
2 1 1
2 1 2
2 1 3
2 2 1
2 2 2
2 2 3
2 3 1
2 3 2
2 3 3
3 1 1
3 1 2
3 1 3
3 2 1
3 2 2
3 2 3
3 3 1
3 3 2
3 3 3
bmsce@bmsce-HP-Pro-3330-MT:~$
```

**<u>PROGRAM11:</u> Shell script to find the power of a number**

CODE:

```
#!/bin/bash
echo "enter the base value"
read b
echo "enter the value of power"
read p
res=1
for ((c=1; c<=$p; c++))
do
        res=`echo "scale=3; $b*$res"|bc`
done
echo $res
```

OUTPUT:



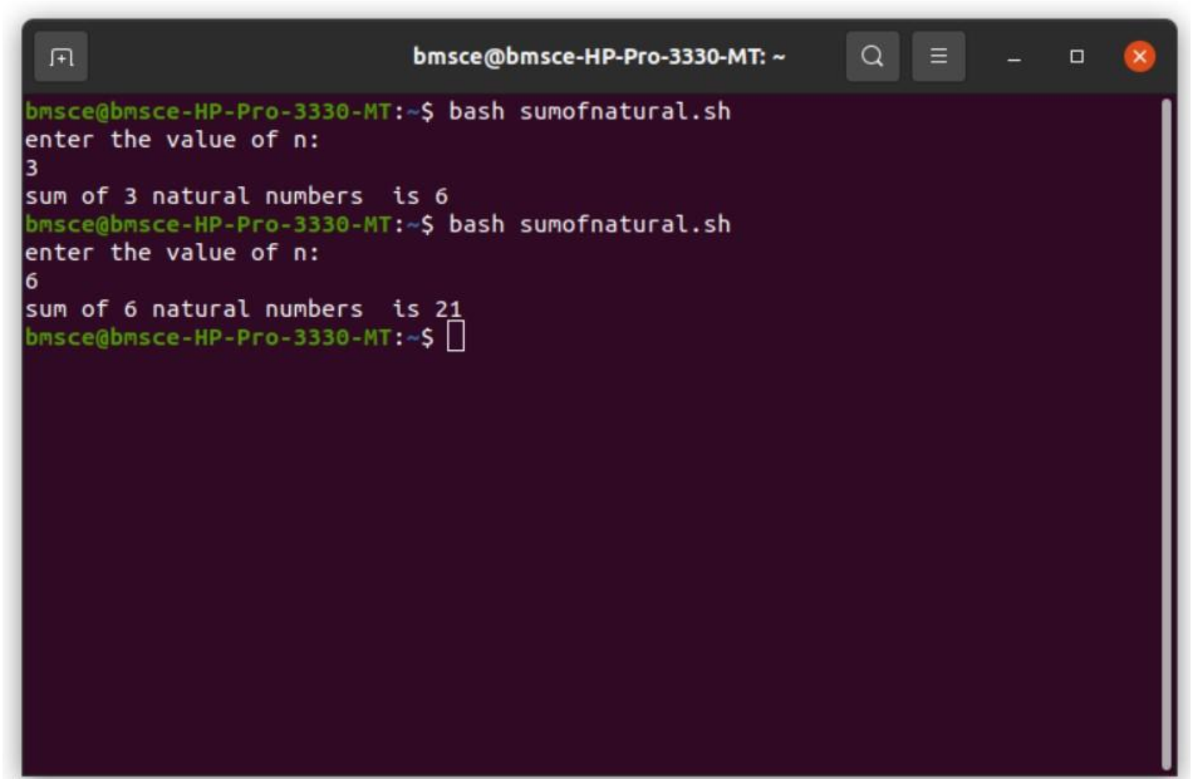
```
bmsce@bmsce-HP-Pro-3330-MT:~$ bash powerofno.sh
enter the base value
2
enter the value of power
3
8
bmsce@bmsce-HP-Pro-3330-MT:~$ bash powerofno.sh
enter the base value
2.3
enter the value of power
3
12.167
bmsce@bmsce-HP-Pro-3330-MT:~$
```

**<u>PROGRAM12:</u> Shell script to find the sum of n natural numbers**

CODE:

```
#!/bin/bash
echo "enter the value of n:"
read n
sum=0
for ((c=0; c<=$n; c++))
do
        sum=`expr $sum+$c|bc`
done
echo "sum of $n natural numbers  is $sum"
```

OUTPUT:



```
bmsce@bmsce-HP-Pro-3330-MT:~$ bash sumofnatural.sh
enter the value of n:
3
sum of 3 natural numbers  is 6
bmsce@bmsce-HP-Pro-3330-MT:~$ bash sumofnatural.sh
enter the value of n:
6
sum of 6 natural numbers  is 21
bmsce@bmsce-HP-Pro-3330-MT:~$ 
```

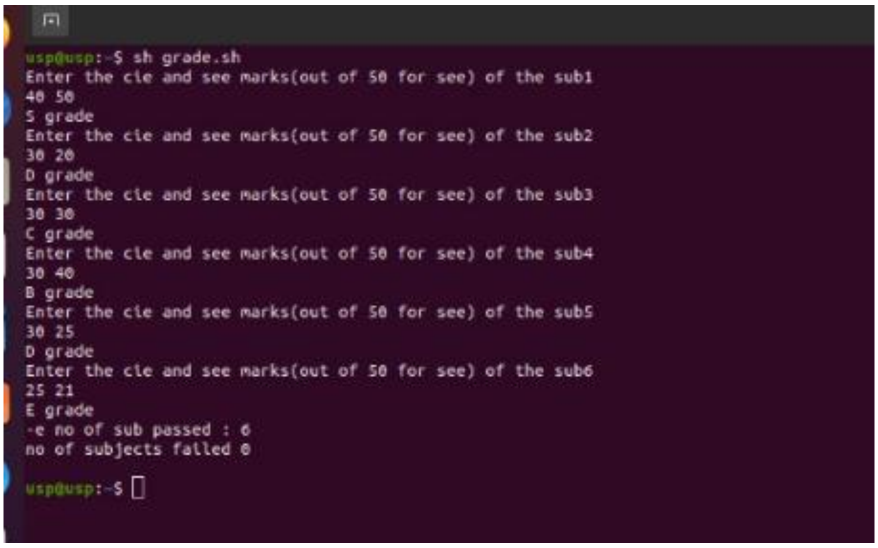
**PROGRAM13: Shell script to display the pass class of a student**

CODE:

```
#!/bin/sh

pass=0

fail=0

i=1

while [ $i -le 6 ]

do

        echo "Enter the cie and see marks(out of 50 for see) of the sub$i "

        read cie see

        total=`expr $cie+$see|bc`

        case $total in

                100) echo "S grade "

                        pass=$((pass+1)) ;;

                9[0-9]) echo "S grade "

                        pass=$((pass+1)) ;;
```

```
                    8[0-9]) echo "A grade "
                              pass=$((pass+1)) ;;
                    7[0-9]) echo "B grade "
                              pass=$((pass+1)) ;;
                    6[0-9]) echo "C grade "
                              pass=$((pass+1)) ;;
                    5[0-9]) echo "D grade "
                              pass=$((pass+1)) ;;
                    4[0-9]) echo "E grade "
                              pass=$((pass+1)) ;;
                    [0123][0-9]) echo "F grade "
                              fail=$((fail+1)) ;;
                    *)echo "error in input"
          esac
          i=$((i+1))
done
echo -e "no of sub passed : $pass\nno of subjects failed $fail\n"
```

OUTPUT:



```
usp@usp:~$ sh grade.sh
Enter the cie and see marks(out of 50 for see) of the sub1
40 50
S grade
Enter the cie and see marks(out of 50 for see) of the sub2
30 20
D grade
Enter the cie and see marks(out of 50 for see) of the sub3
30 30
C grade
Enter the cie and see marks(out of 50 for see) of the sub4
30 40
B grade
Enter the cie and see marks(out of 50 for see) of the sub5
30 25
D grade
Enter the cie and see marks(out of 50 for see) of the sub6
25 21
E grade
-e no of sub passed : 6
no of subjects failed 0

usp@usp:~$ 
```

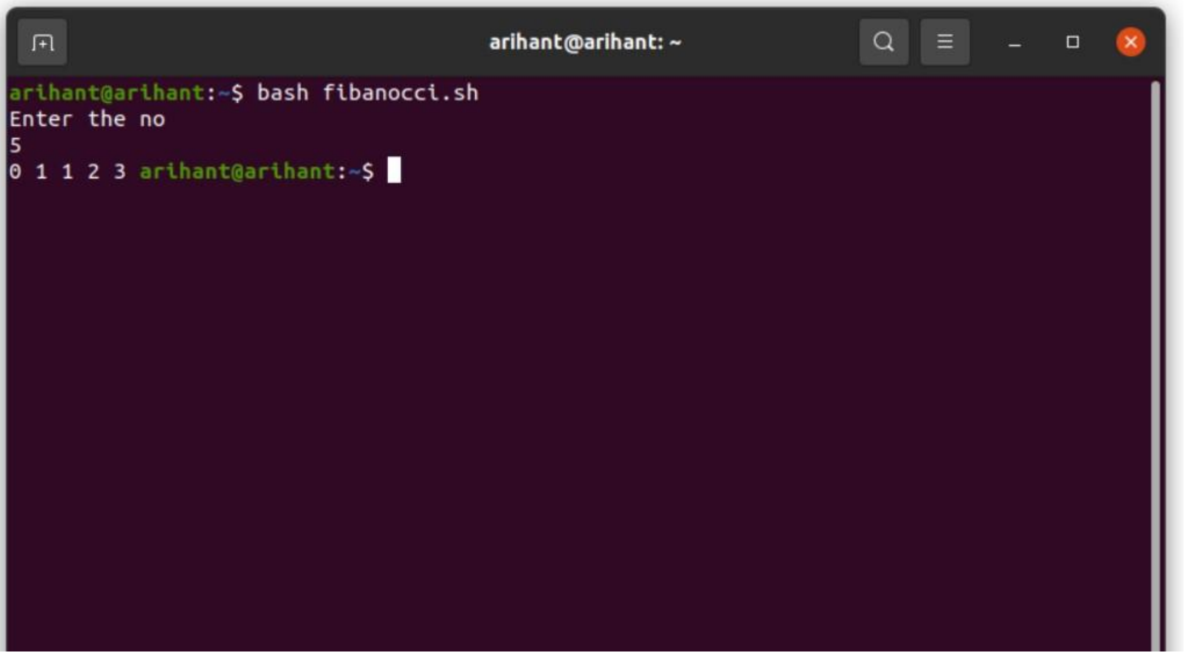
**PROGRAM14: Shell script to find the Fibonacci series up to n**

CODE:

```
#!/bin/sh
echo "Enter the no"
```

```
read no
m=0
n=1
while [ $no -gt 0 ]
do
        echo -e "$m \c"
        temp=$m
        m=$((m+$n))
        n=$temp
        no=$((no-1))
done
```

OUTPUT:



```
arihant@arihant:~$ bash fibanocci.sh
Enter the no
5
0 1 1 2 3 arihant@arihant:~$
```

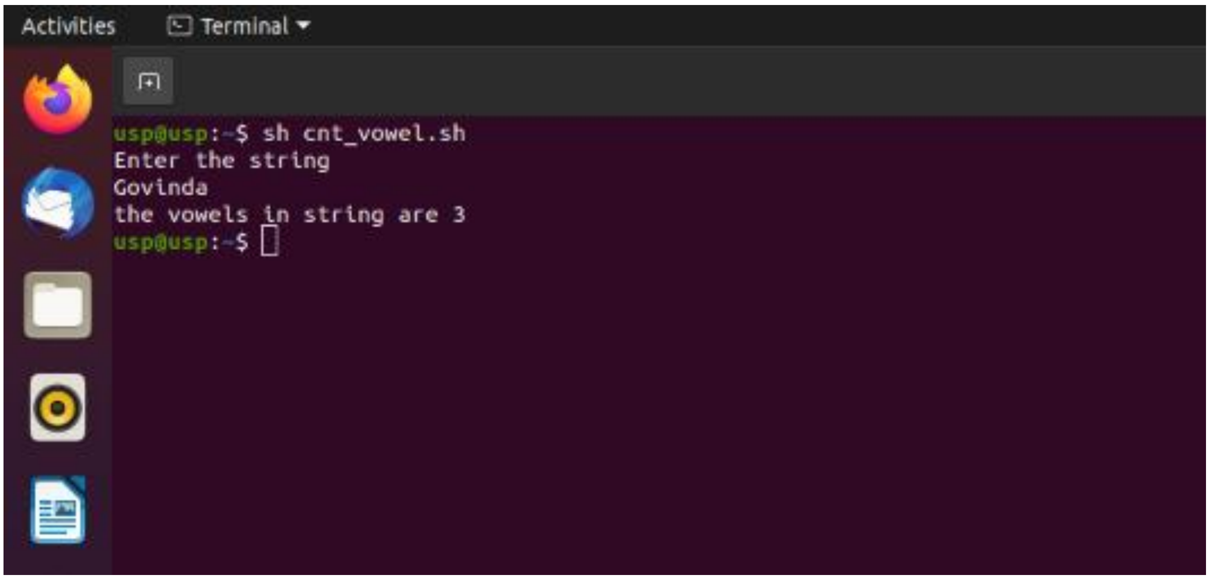
**<u>PROGRAM15: </u>Shell script to count the number of vowels of a string**

CODE:

```
#!/bin/sh
echo "Enter the string "
```

```
read str
count=0
len=`expr length $str`
while [ $len -gt 0 ]
do
        ch=`expr $str | cut -c $len`
        case $ch in
        [aeiouAEIOU]) count=$((count+1)) ;;
        esac
        len=$((len-1))
done
echo "the vowels in string are $count "
```

OUTPUT:



```
usp@usp:~$ sh cnt_vowel.sh
Enter the string
Govinda
the vowels in string are 3
usp@usp:~$
```

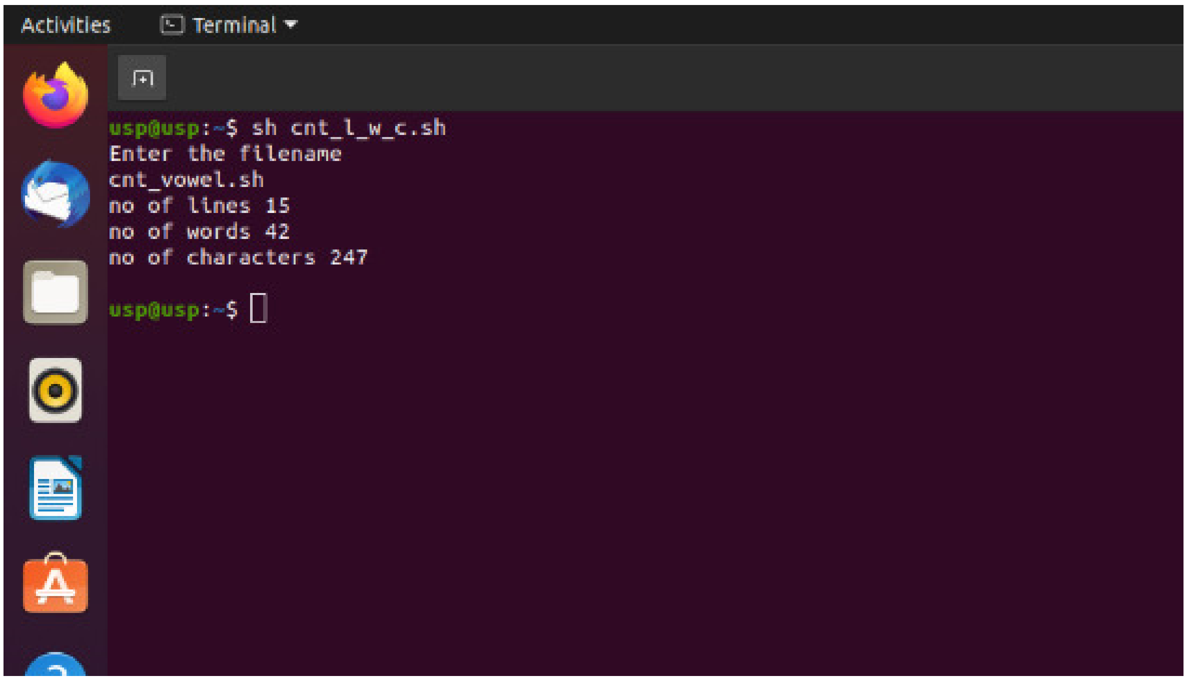
**<u>PROGRAM16:</u> Shell script to check number of lines, words, characters in a file**

CODE:

```
#!/bin/sh
echo "Enter the filename "
read fname
l=`wc -l < $fname`
```

```
w=`wc -w < $fname`
c=`wc -m < $fname`
echo -e "no of lines $l\nno of words $w\nno of characters $c\n"
```

OUTPUT:



```
usp@usp:~$ sh cnt_l_w_c.sh
Enter the filename
cnt_vowel.sh
no of lines 15
no of words 42
no of characters 247

usp@usp:~$
```

**<u>PROGRAM17:</u> Write a C/C++ program to that outputs the contents of its Environment list**

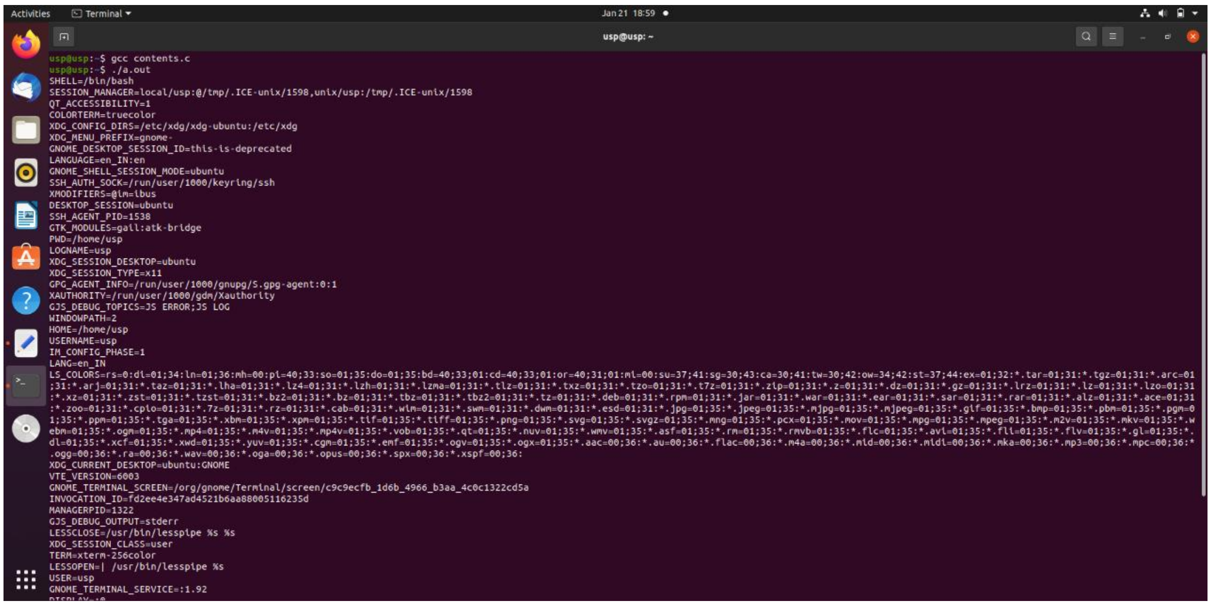
CODE:

```
#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:



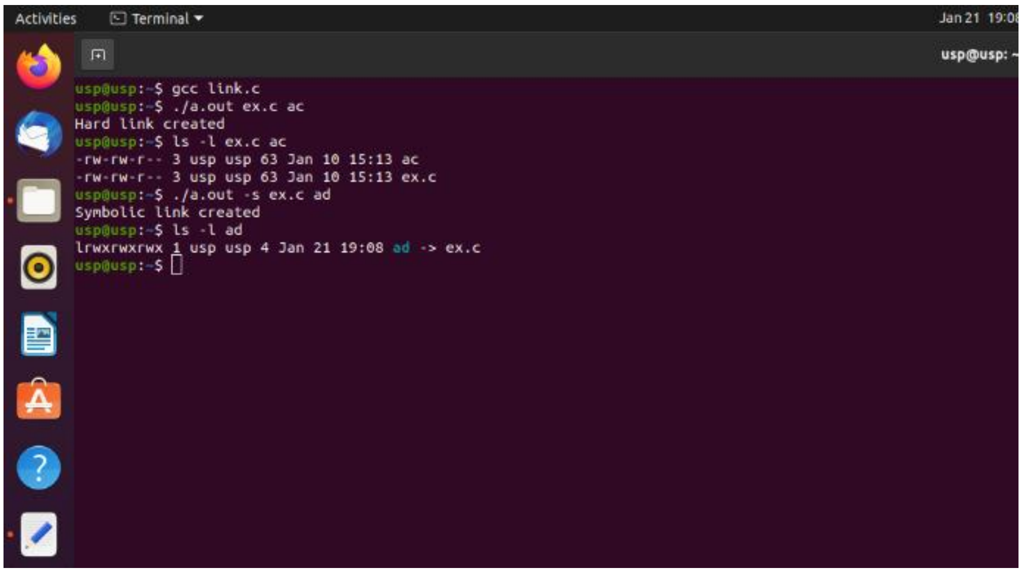
```
usp@usp:~$ gcc contents.c
usp@usp:~$ ./a.out
SHELL=/bin/bash
SESSION_MANAGER=local/usp:@/tmp/.ICE-unix/1598,unix/usp:/tmp/.ICE-unix/1598
QT_ACCESSIBILITY=1
COLORTERM=truecolor
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/etc/xdg
XDG_MENU_PREFIX=gnome-
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
LANGUAGE=en_IN:en
GNOME_SHELL_SESSION_MODE=ubuntu
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
XMODIFIERS=@im=ibus
DESKTOP_SESSION=ubuntu
SSH_AGENT_PID=1538
GTK_MODULES=gail:atk-bridge
PWD=/home/usp
LOGNAME=usp
XDG_SESSION_DESKTOP=ubuntu
XDG_SESSION_TYPE=x11
GPG_AGENT_INFO=/run/user/1000/gnupg/S.gpg-agent:0:1
XAUTHORITY=/run/user/1000/gdm/Xauthority
GJS_DEBUG_TOPICS=JS ERROR;JS LOG
WINDOWPATH=2
HOME=/home/usp
USERNAME=usp
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=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:*.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=01;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=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=0
1;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:*.w
ebm=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:
XDG_CURRENT_DESKTOP=ubuntu:GNOME
VTE_VERSION=6003
GNOME_TERMINAL_SCREEN=/org/gnome/Terminal/screen/c9c9ecfb_1d6b_4966_b3aa_4c0c1322cd5a
INVOCATION_ID=fd2ee4e347ad4521b6aa88005116235d
MANAGERPID=1322
GJS_DEBUG_OUTPUT=stderr
LESSCLOSE=/usr/bin/lesspipe %s %s
XDG_SESSION_CLASS=user
TERM=xterm-256color
LESSOPEN=| /usr/bin/lesspipe %s
USER=usp
GNOME_TERMINAL_SERVICE=:1.92
DISPLAY=:0
```

**<u>PROGRAM18:</u> Write a C/C++ program to emulate the unix ln command**

CODE:

```
#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:



```
usp@usp:~$ gcc link.c
usp@usp:~$ ./a.out ex.c ac
Hard link created
usp@usp:~$ ls -l ex.c ac
-rw-rw-r-- 3 usp usp 63 Jan 10 15:13 ac
-rw-rw-r-- 3 usp usp 63 Jan 10 15:13 ex.c
usp@usp:~$ ./a.out -s ex.c ad
Symbolic link created
usp@usp:~$ ls -l ad
lrwxrwxrwx 1 usp usp 4 Jan 21 19:08 ad -> ex.c
usp@usp:~$ 
```

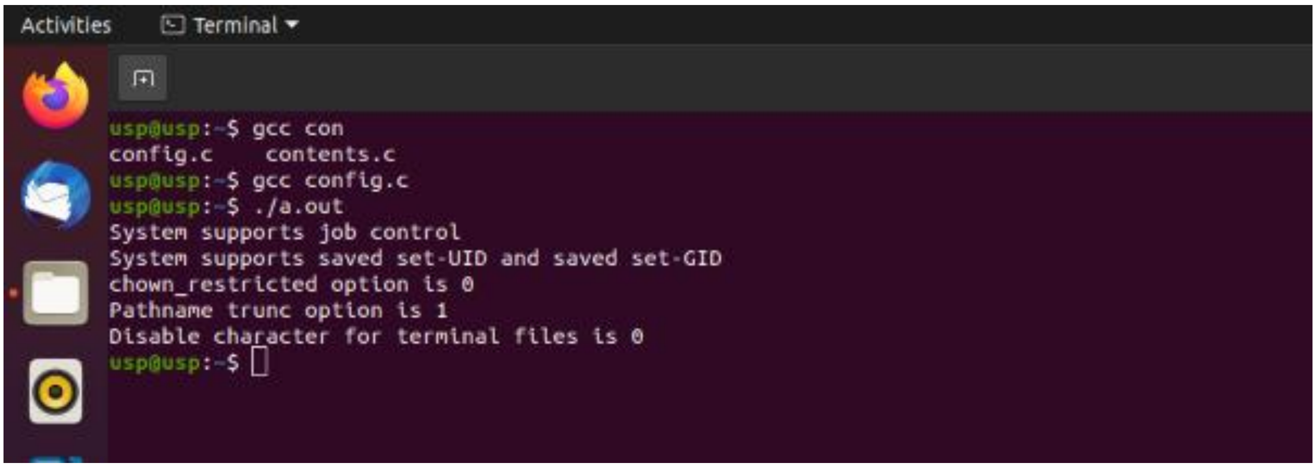
**<u>PROGRAM19:</u> Write a C/C++ POSIX compliant program that prints the POSIX defined configuration options supported on any given system using feature test macros.**

CODE:

```
#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:



```
usp@usp:~$ gcc con
config.c    contents.c
usp@usp:~$ gcc config.c
usp@usp:~$ ./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
usp@usp:~$ 
```

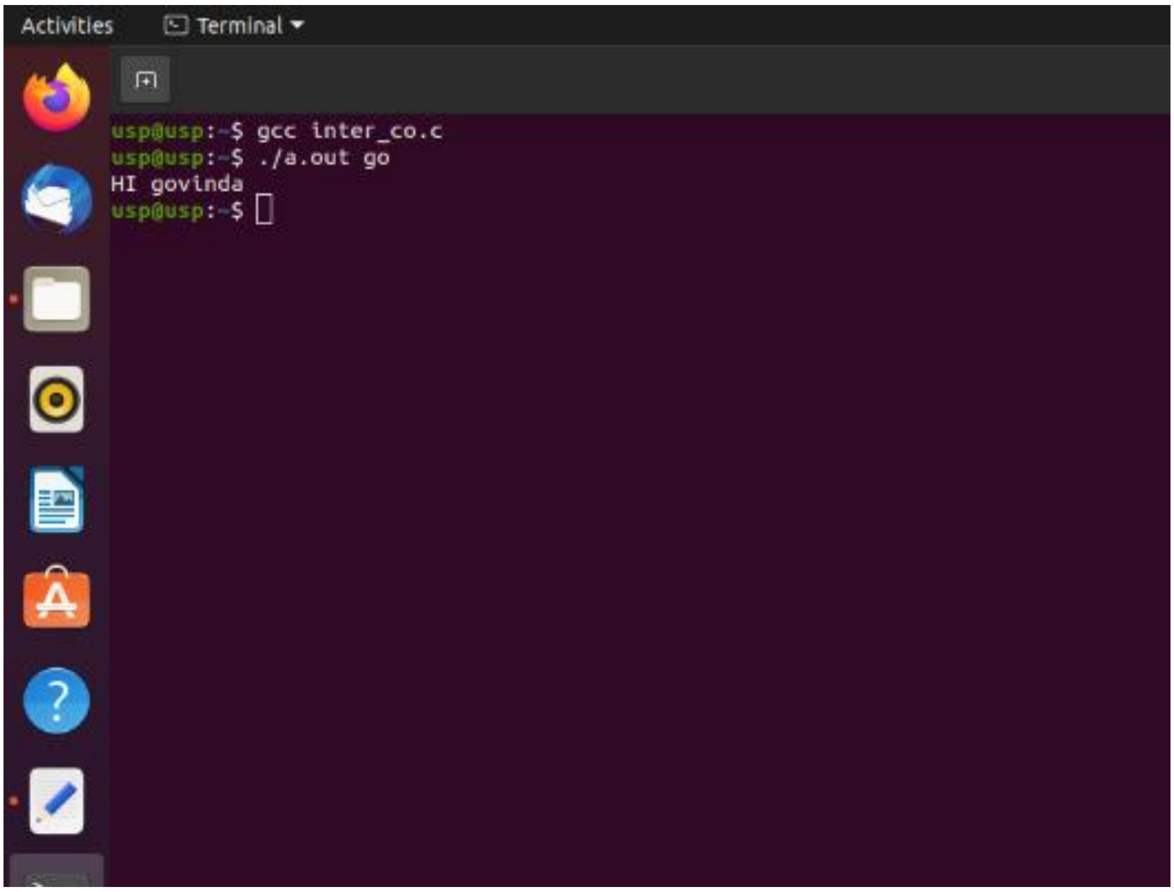
**<u>PROGRAM20:</u> Write a C/C++ program which demonstrates interprocess communication between a reader process and a writer process. Use mkfifo, open, read, write and close APIs in**

**your program.**

CODE:

```
#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)   //reader process
{
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);
}
```

OUTPUT:



```
usp@usp:~$ gcc inter_co.c
usp@usp:~$ ./a.out go
HI govinda
usp@usp:~$
```