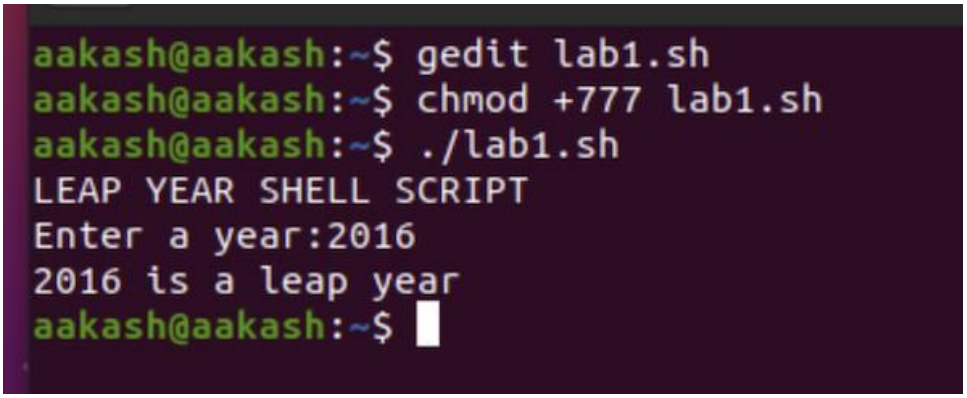
UNIX LAB

Q1. Shell script to find if the given year is leap or not

Code:

```
#!/bin/bash

echo  "Enter the year"
read y

if[$((y%400)) -eq 0]
then
echo "$y is Leap Year"

elif[$((y%100))-eq 0]
then
echo "$y is not a leap year"

elif[$((y%4))-eq 0]
then
echo "$y is a leap year"

else
echo "$y is not a leap year"
Fi
```

OUTPUT:



```
aakash@aakash:~$ gedit lab1.sh
aakash@aakash:~$ chmod +777 lab1.sh
aakash@aakash:~$ ./lab1.sh
LEAP YEAR SHELL SCRIPT
Enter a year:2016
2016 is a leap year
aakash@aakash:~$
```

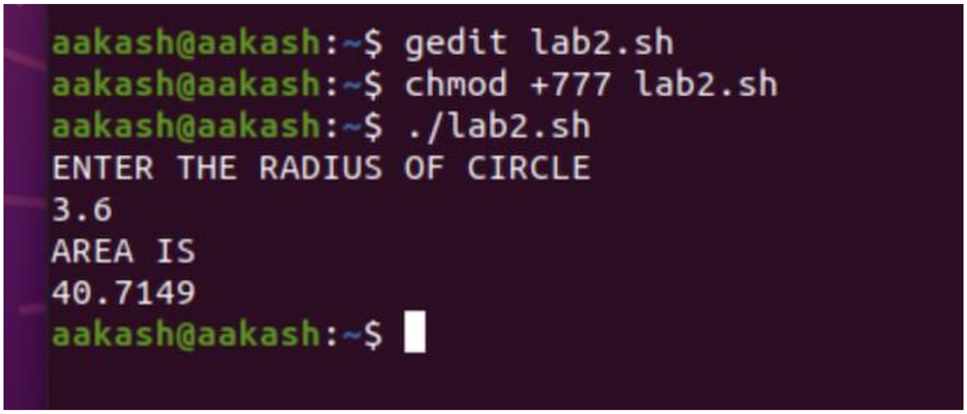
Q2.

Shell script to find the area of a circle

Code:

```
#!/bin/bash

echo "ENTER THE RADIUS OF CIRCLE"
reasd r
echo "AREA IS "
pi=3.1416
ans=`expr $pi \* $r \* $r|bc`
echo "$ans"
```

OUTPUT:



```
aakash@aakash:~$ gedit lab2.sh
aakash@aakash:~$ chmod +777 lab2.sh
aakash@aakash:~$ ./lab2.sh
ENTER THE RADIUS OF CIRCLE
3.6
AREA IS
40.7149
aakash@aakash:~$ 
```

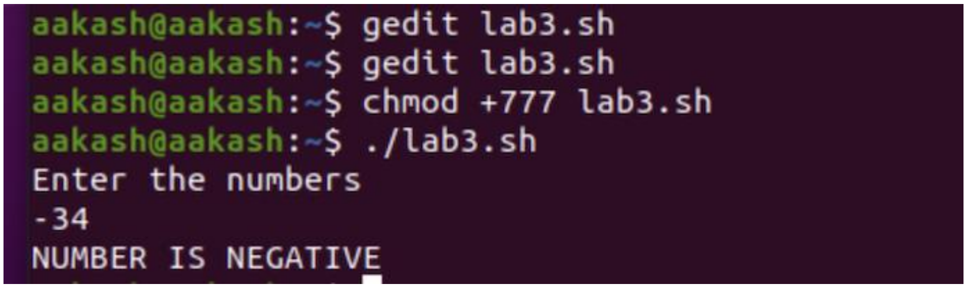
Q3.

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

CODE:

```
#!/bin/sh
echo "Enter the numbers"
read num

if[$num -eq 0]
then
echo "NUMBER IS ZERO"
elif[$num -gt 0]
then
echo "NUMBER IS POSITIVE"
else
echo "NUMBER IS NEGATIVE"
Fi
```

OUTPUT:



```
aakash@aakash:~$ gedit lab3.sh
aakash@aakash:~$ gedit lab3.sh
aakash@aakash:~$ chmod +777 lab3.sh
aakash@aakash:~$ ./lab3.sh
Enter the numbers
-34
NUMBER IS NEGATIVE
```

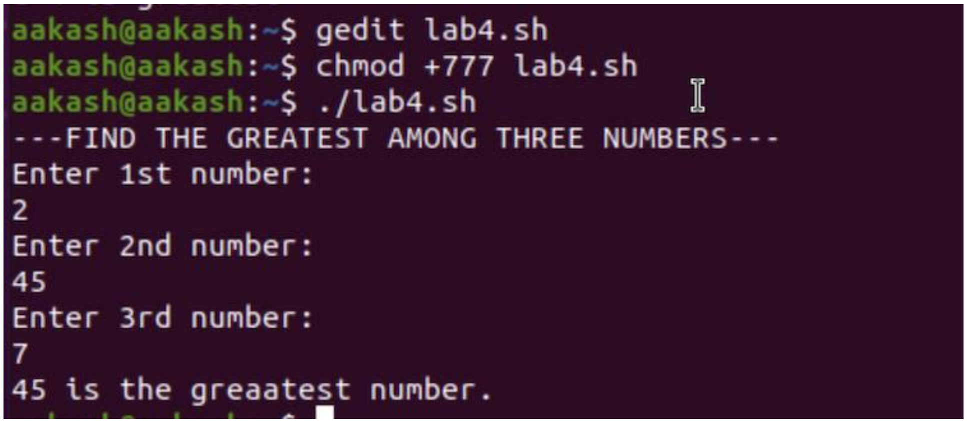
Q4.

Shell script to find the biggest of three numbers

CODE:

```
# gratest of three number

#!/bin/sh
echo "Enter first number"
read a
echo "Enter the second number"
read b
echo "Enter the third number"
read c

if[($a -gt $b) -a ($a -gt $c)]
then
echo "A  $a is greatest"

elif[($b -gt $a) -a ($b -gt $c)]
then
echo "B  $b is greatest"

else
echo "C $c is greatest"
```

Fi



```
aakash@aakash:~$ gedit lab4.sh
aakash@aakash:~$ chmod +777 lab4.sh
aakash@aakash:~$ ./lab4.sh
---FIND THE GREATEST AMONG THREE NUMBERS---
Enter 1st number:
2
Enter 2nd number:
45
Enter 3rd number:
7
45 is the greaatest number.
```

SET 2

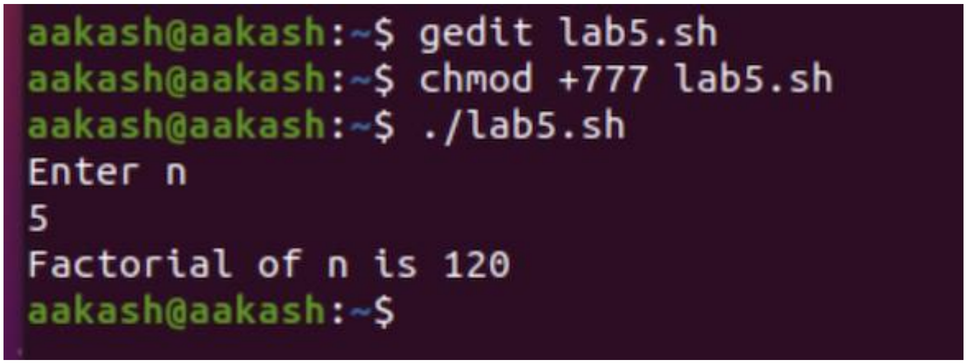
Q5.

Shell script to find the factorial of a number

CODE:

```
#!/bin/bash

echo "Enter n"
read n
fact=1

while [ $n -gt 1 ]
do
fact=$((fact*n))
n=$((n-1))
done
echo "Factorial of n is $fact"
```

OUTPUT:



```
aakash@aakash:~$ gedit lab5.sh
aakash@aakash:~$ chmod +777 lab5.sh
aakash@aakash:~$ ./lab5.sh
Enter n
5
Factorial of n is 120
aakash@aakash:~$
```

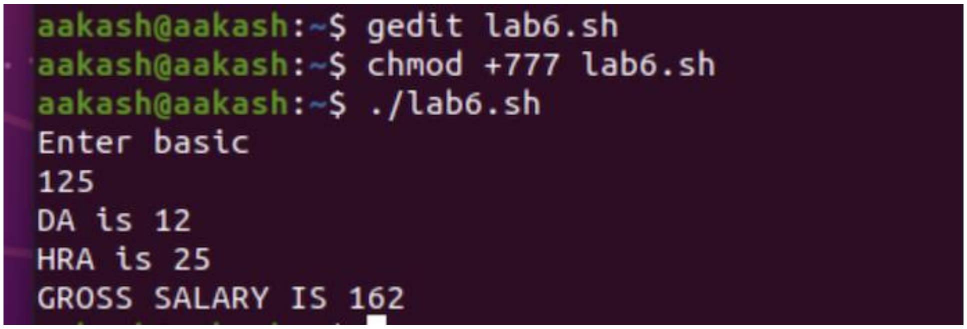
Q6.

Shell script to compute the gross salary of an employee

CODE:

```
#!/bin/bash


echo "Enter basic"

read basic


DA=`expr $basic \* 10 / 100`

echo "DA is $DA"


HRA=`expr $basic \* 20 / 100`

echo "HRA is $HRA"


GROSS_SAL=`expr $basic \+ $DA \+ $HRA`


echo "GROSS SALARY IS $GROSS_SAL"
```

OUTPUT:



```
aakash@aakash:~$ gedit lab6.sh
aakash@aakash:~$ chmod +777 lab6.sh
aakash@aakash:~$ ./lab6.sh
Enter basic
125
DA is 12
HRA is 25
GROSS SALARY IS 162
```

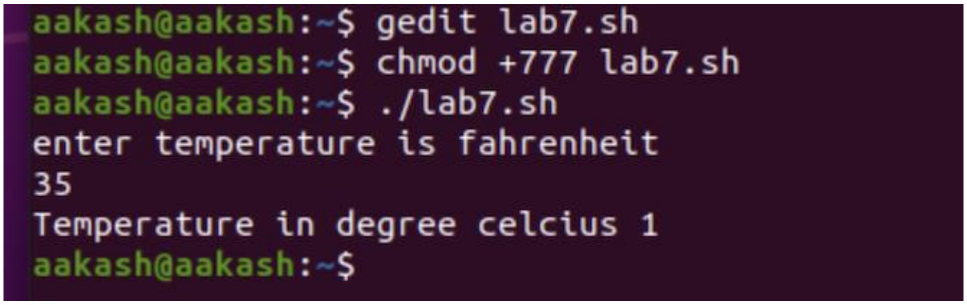
Q7.

Shell script to convert the temperature Fahrenheit to Celsius

CODE:

```
#!/bin/bash


echo "enter temperature is fahrenheit"

read f


c1=`expr $f - 32 `

c=`expr $c1 \* 5 / 9 |bc`



echo "Temperature in degree celcius $c"
```

OUTPUT:



```
aakash@aakash:~$ gedit lab7.sh
aakash@aakash:~$ chmod +777 lab7.sh
aakash@aakash:~$ ./lab7.sh
enter temperature is fahrenheit
35
Temperature in degree celcius 1
aakash@aakash:~$
```

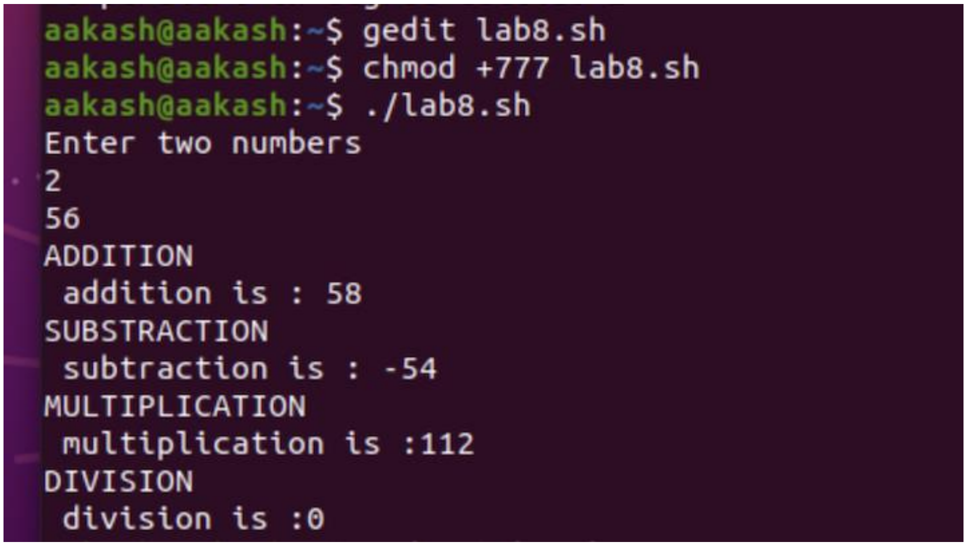
Q8.

Shell script to perform arithmetic operations on given two numbers

CODE:

```
#!/bin/bash

echo "Enter two numbers"
read a
read b

add=`expr $a \+ $b|bc`
echo "ADDITION"
echo " addition is : $add "

sub=`expr $a \- $b|bc`
echo "SUBSTRACTION"
echo " subtraction is : $sub "

mul=`expr $a \* $b|bc`
echo "MULTIPLICATION"
echo " multiplication is :$mul "

div=`expr $a \/ $b|bc`
echo "DIVISION"
echo " division is :$div "
```

OUTPUT:



```
aakash@aakash:~$ gedit lab8.sh
aakash@aakash:~$ chmod +777 lab8.sh
aakash@aakash:~$ ./lab8.sh
Enter two numbers
2
56
ADDITION
 addition is : 58
SUBSTRACTION
 subtraction is : -54
MULTIPLICATION
 multiplication is :112
DIVISION
 division is :0
```

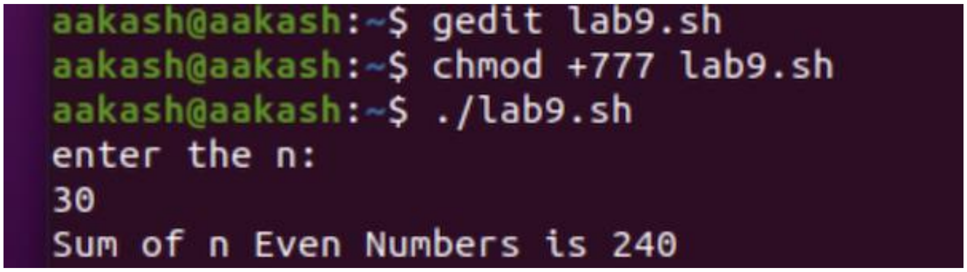
Q9.

Shell script to find the sum of even numbers upto n

CODE:

```
#!/bin/sh

echo "enter the n:"
read n
i=0
sum=0
t=`expr $n \+ 1`
while [ $i -lt $t ]
do
sum=`expr $sum \+ $i`
i=`expr $i \+ 2`
done

echo "Sum of n Even Numbers is $sum"
```

OUTPUT:



```
aakash@aakash:~$ gedit lab9.sh
aakash@aakash:~$ chmod +777 lab9.sh
aakash@aakash:~$ ./lab9.sh
enter the n:
30
Sum of n Even Numbers is 240
```

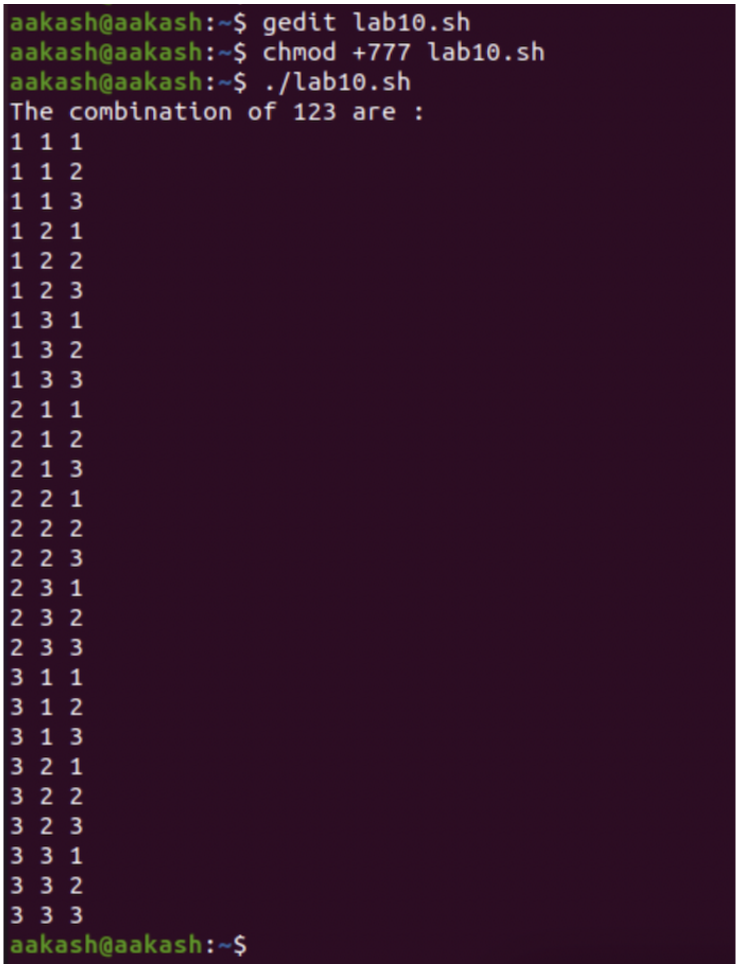
Q10.

Shell script to print the combinations of numbers 123

CODE:

```
#!/bin/sh


echo "The combination of 123 are :"


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:



```
aakash@aakash:~$ gedit lab10.sh
aakash@aakash:~$ chmod +777 lab10.sh
aakash@aakash:~$ ./lab10.sh
The combination of 123 are :
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
aakash@aakash:~$
```

Q11.

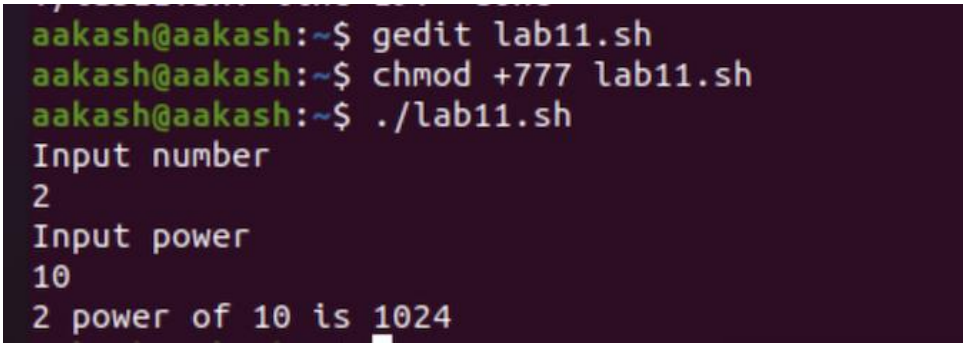
Shell script to find the power of a number

CODE:

```
#!/bin/sh

echo "Enter number:"
read num

echo "input power:"
read pow
```

```
i=num

res=1

count=1

t=`expr $pow \+ 1`


while [ $count -lt $t ]

do

res='expr $res \* $i'

count=`expr $count \+ 1`

done


echo "$num^$pow=$res"
```

OUTPUT:



```
aakash@aakash:~$ gedit lab11.sh
aakash@aakash:~$ chmod +777 lab11.sh
aakash@aakash:~$ ./lab11.sh
Input number
2
Input power
10
2 power of 10 is 1024
```

Q12.

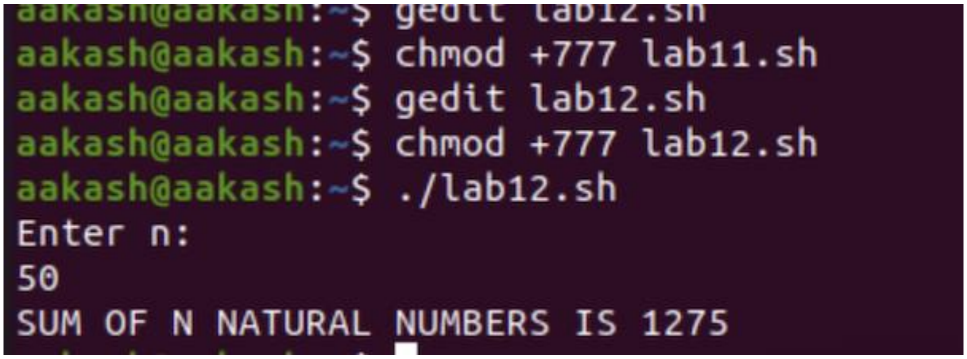
Shell script to find the sum of n natural numbers

CODE:

```
#!/bin/sh


echo "Enter n:"

read n
```

```
i=0

sum=0

t=`expr $n \+ 1`

while [ $i -lt $t ]

do

sum=`expr $sum \+ $i`

i=`expr $i \+ 1`

done


echo "SUM OF N NATURAL NUMBERS IS $sum"
```

OUTPUT:



```
aakash@aakash:~$ gedit lab12.sh
aakash@aakash:~$ chmod +777 lab11.sh
aakash@aakash:~$ gedit lab12.sh
aakash@aakash:~$ chmod +777 lab12.sh
aakash@aakash:~$ ./lab12.sh
Enter n:
50
SUM OF N NATURAL NUMBERS IS 1275
aakash@aakash:~$
```

Q13.

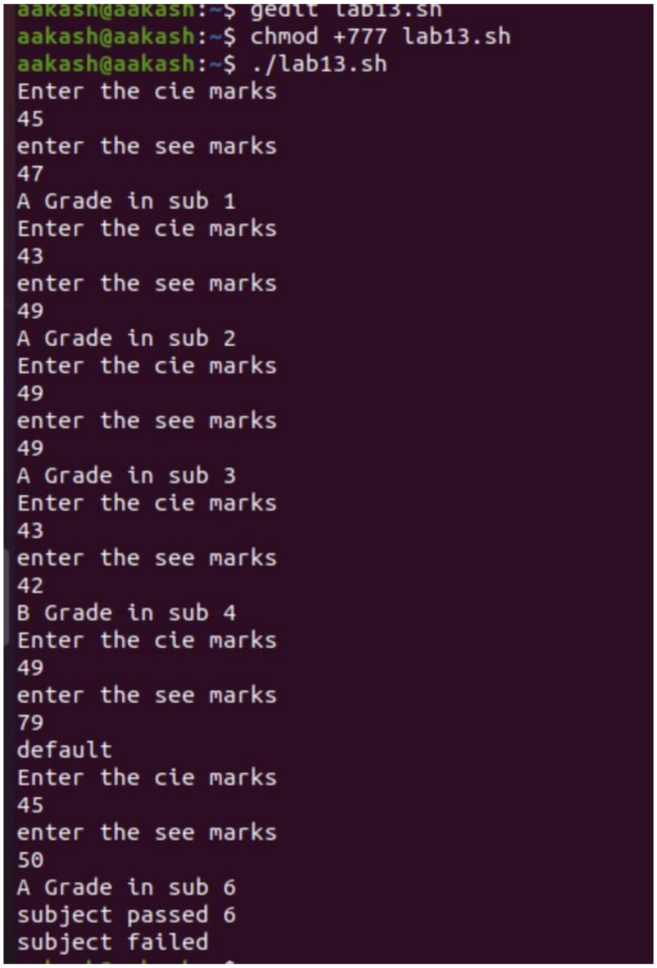
Shell script to display the pass class of a student

CODE:

```
#!/bin/bash


#total=0

for((i=1;i<=6;i++))

do

echo "Enter the cie marks "
```

```
read cie
echo "enter the see marks "
read see
total=$((cie + see))
#if [ $total -lt 40 ]
#temp=t
case $total in
100) echo "S Grade in sub $i";;
9[0-9]) echo "A Grade in sub $i";;
8[0-9]) echo "B Grade in sub $i";;
7[0-9]) echo "C Grade in sub $i";;
6[0-9]) echo "D Grade in sub $i";;
5[0-9]) echo "E Grade in sub $i";;
4[0-9]) echo "F Grade in sub $i";;
3[0-9]) echo "F Grade in sub $i";;
2[0-9]) echo "F Grade in sub $i" ;;
1[0-9]) echo "F Grade in sub $i";;
*) echo "default";;
esac
if [ $total -gt 40 ]
then
countpass=$((countpass+1))
else
countfail=$((countfail+1))
fi
done
echo "subject passed $countpass"
echo "subject failed $countfailed"
```

OUTPUT:



```
aakash@aakash:~$ gedit lab13.sh
aakash@aakash:~$ chmod +777 lab13.sh
aakash@aakash:~$ ./lab13.sh
Enter the cie marks
45
enter the see marks
47
A Grade in sub 1
Enter the cie marks
43
enter the see marks
49
A Grade in sub 2
Enter the cie marks
49
enter the see marks
49
A Grade in sub 3
Enter the cie marks
43
enter the see marks
42
B Grade in sub 4
Enter the cie marks
49
enter the see marks
79
default
Enter the cie marks
45
enter the see marks
50
A Grade in sub 6
subject passed 6
subject failed
```

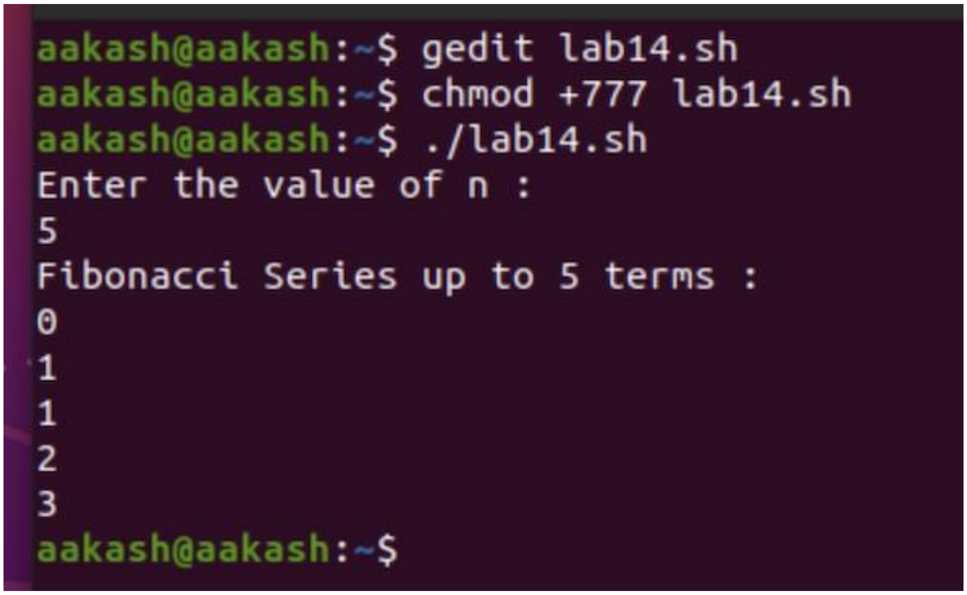
Q14.

Shell script to find the Fibonacci series up to n

CODE:

```
#!/bin/sh

echo "Enter the value of n :"
read n
x=0
y=1
i=2
echo "Fibonacci Series up to $n terms :"
echo "$x"

echo "$y"

while [ $i -lt $n ]

do

i=`expr $i + 1 `

z=`expr $x + $y `

echo "$z"

x=$y

y=$z

Done
```

OUTPUT:



```
aakash@aakash:~$ gedit lab14.sh
aakash@aakash:~$ chmod +777 lab14.sh
aakash@aakash:~$ ./lab14.sh
Enter the value of n :
5
Fibonacci Series up to 5 terms :
0
1
1
2
3
aakash@aakash:~$
```

Q15

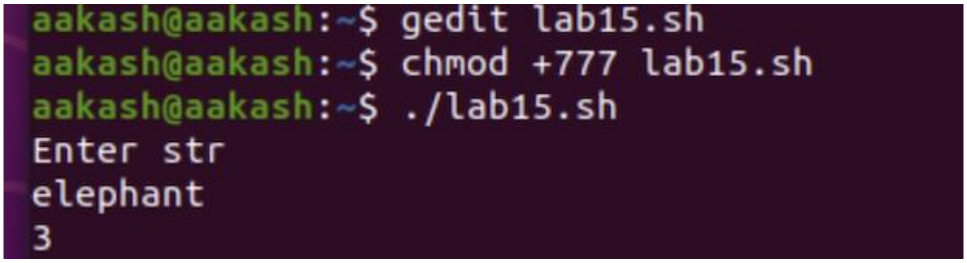
Shell script to count the number of vowels of a string

CODE:

```
#!/bin/bash
```

```
echo "enter the string"
read str
count=0

for $i in str
do
if [ $i=='a' -o $i=='e' -o $i=='i' -o $i=='o' -o $i=='u' ]
then
count++
done

echo "The number of vowels in the string is $count"
```

OUTPUT:



```
aakash@aakash:~$ gedit lab15.sh
aakash@aakash:~$ chmod +777 lab15.sh
aakash@aakash:~$ ./lab15.sh
Enter str
elephant
3
```

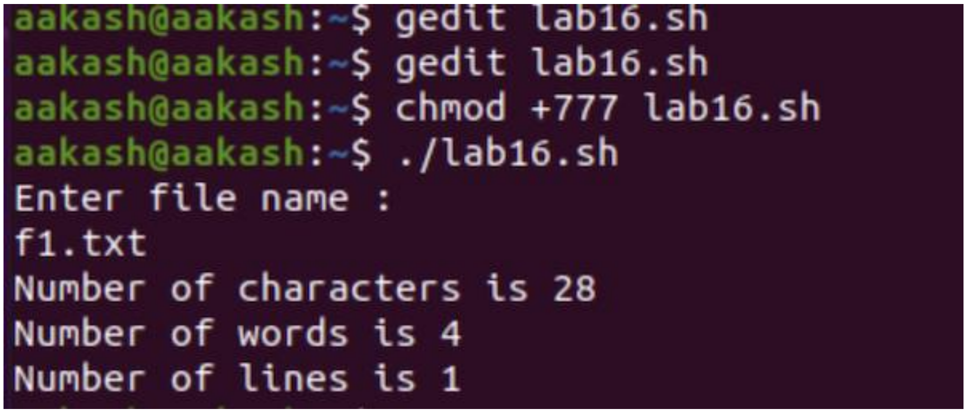
Q16.

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

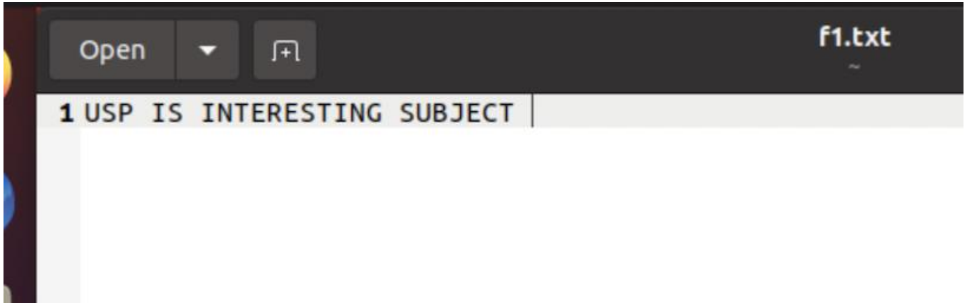
CODE:

```
#!/bin/bash
echo "Enter the file name:"
read file_name
c=`cat $file_name | wc -c`
w=`cat $file_name | wc -w`
l=`grep -c "." $file_name`
echo "No. of characters is $c"
echo "No. of words is $w"

echo "No. of lines is $l"
```

OUTPUT:



```
aakash@aakash:~$ gedit lab16.sh
aakash@aakash:~$ gedit lab16.sh
aakash@aakash:~$ chmod +777 lab16.sh
aakash@aakash:~$ ./lab16.sh
Enter file name :
f1.txt
Number of characters is 28
Number of words is 4
Number of lines is 1
```



```
USP IS INTERESTING SUBJECT
```

Q17.

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;

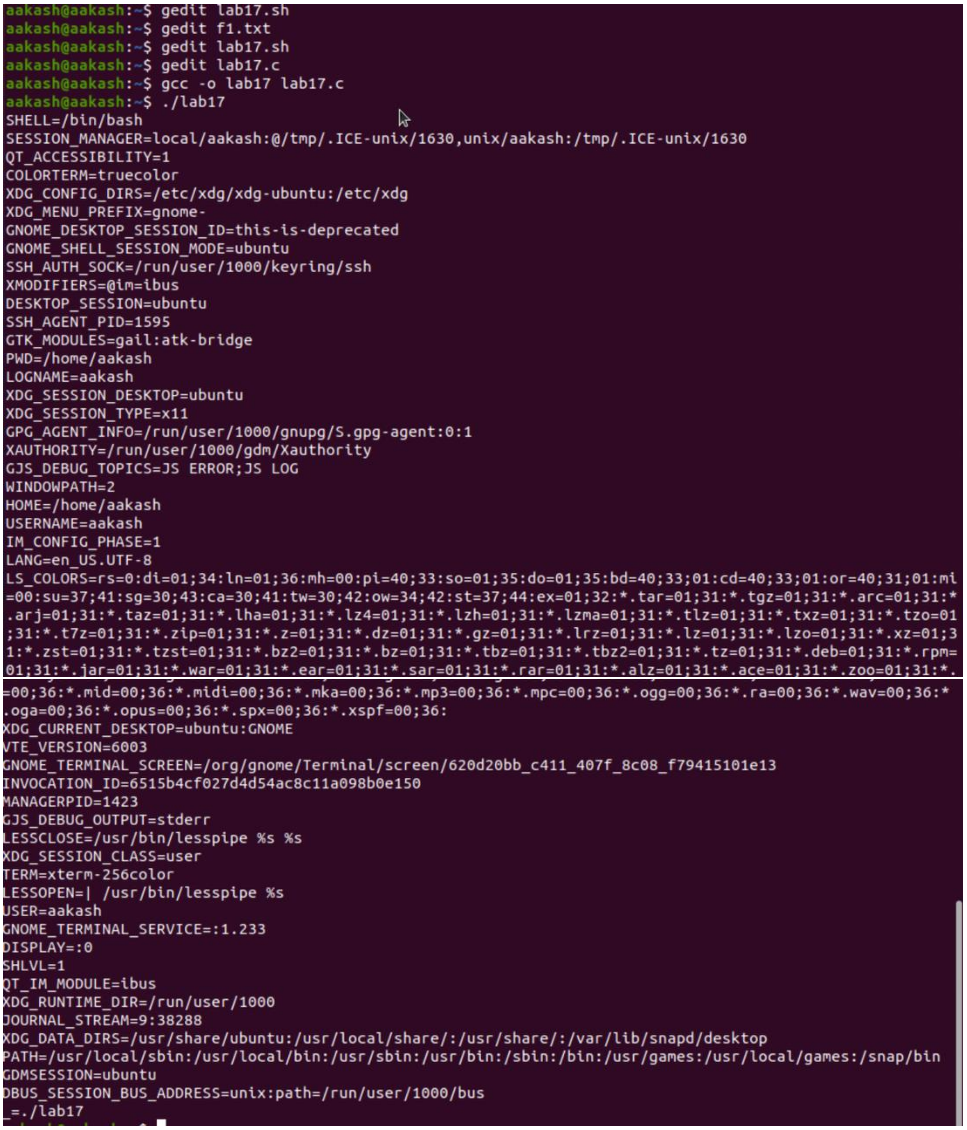
for(ptr = environ; *ptr != 0; ptr++)

printf("%s\n",*ptr);

return 0;

}

OUTPUT:



```
aakash@aakash:~$ gedit lab17.sh
aakash@aakash:~$ gedit f1.txt
aakash@aakash:~$ gedit lab17.sh
aakash@aakash:~$ gedit lab17.c
aakash@aakash:~$ gcc -o lab17 lab17.c
aakash@aakash:~$ ./lab17
SHELL=/bin/bash
SESSION_MANAGER=local/aakash:@/tmp/.ICE-unix/1630,unix/aakash:/tmp/.ICE-unix/1630
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
GNOME_SHELL_SESSION_MODE=ubuntu
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
XMODIFIERS=@im=ibus
DESKTOP_SESSION=ubuntu
SSH_AGENT_PID=1595
GTK_MODULES=gail:atk-bridge
PWD=/home/aakash
LOGNAME=aakash
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/aakash
USERNAME=aakash
IM_CONFIG_PHASE=1
LANG=en_US.UTF-8
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;3
1:*.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:*.
=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/620d20bb_c411_407f_8c08_f79415101e13
INVOCATION_ID=6515b4cf027d4d54ac8c11a098b0e150
MANAGERPID=1423
GJS_DEBUG_OUTPUT=stderr
LESSCLOSE=/usr/bin/lesspipe %s %s
XDG_SESSION_CLASS=user
TERM=xterm-256color
LESSOPEN=| /usr/bin/lesspipe %s
USER=aakash
GNOME_TERMINAL_SERVICE=:1.233
DISPLAY=:0
SHLVL=1
QT_IM_MODULE=ibus
XDG_RUNTIME_DIR=/run/user/1000
JOURNAL_STREAM=9:38288
XDG_DATA_DIRS=/usr/share/ubuntu:/usr/local/share/:/usr/share/:/var/lib/snapd/desktop
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
GDMSESSION=ubuntu
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus
_=./lab17
```

Q18.

Write a C/C++ program to emulate the unix ln command

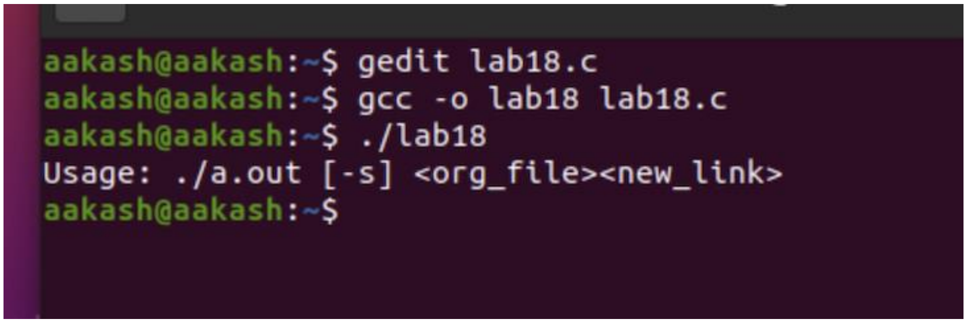
CODE:

```
#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:



```
aakash@aakash:~$ gedit lab18.c
aakash@aakash:~$ gcc -o lab18 lab18.c
aakash@aakash:~$ ./lab18
Usage: ./a.out [-s] <org_file><new_link>
aakash@aakash:~$
```

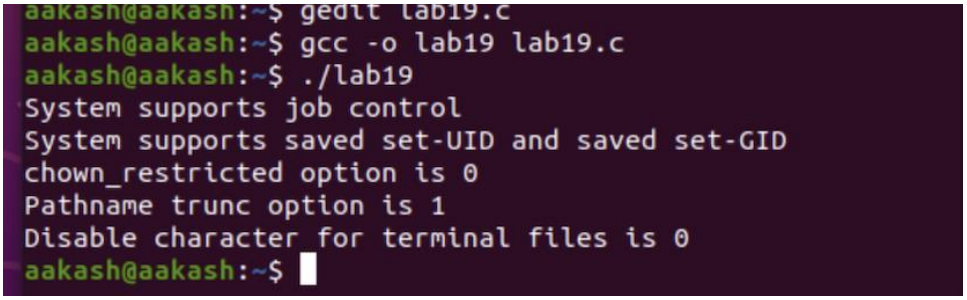
Q19.

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:



```
aakash@aakash:~$ gedit lab19.c
aakash@aakash:~$ gcc -o lab19 lab19.c
aakash@aakash:~$ ./lab19
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
aakash@aakash:~$
```

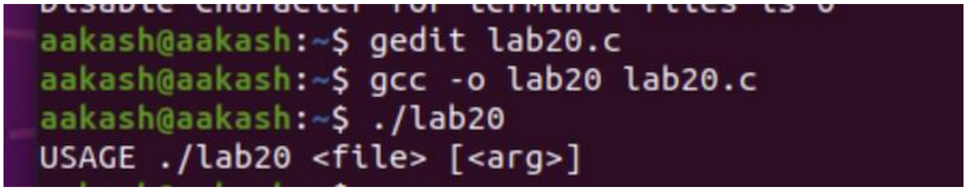
Q20.

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.

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:



```
aakash@aakash:~$ gedit lab20.c
aakash@aakash:~$ gcc -o lab20 lab20.c
aakash@aakash:~$ ./lab20
USAGE ./lab20 <file> [<arg>]
```