

# Particulars of the Experiments Performed

## CONTENTS

Expt No.	Date	Experiment	Marks Obtained	Page No.
1.	25/10/21	Shell script to find if the given year is Leap or not.	100/100	1
2.	25/10/21	Shell script to find the area of a circle.	100/100	2
3.	25/10/21	Shell script to check whether the number is zero/positive/negative.	100/100	3
4.	25/10/21	Shell script to find the biggest of three numbers.	100/100	4
5.	08/11/21	Shell script to find the factorial of a number.	100/100	5
6.	08/11/21	Shell script to compute the gross salary of an employee.	100/100	6
7.	08/11/21	Shell script to convert the temperature Fahrenheit to Celsius.	100/100	7
8.	08/11/21	Shell script to perform arithmetic operations on given two numbers.	100/100	8
9.	15/11/21	Shell script to find the sum of even numbers upto n.	100/100	9
10.	15/11/21	Shell script to print the combinations of nos 123	100/100	10
11.	15/11/21	Shell script to find the power of a number.	100/100	11

# CONTENTS

Expt No.	Date	Experiment	Marks Obtained	Page No.
12.	15/11/21	Shell script to find the sum of n natural nos		
13.		Shell script to display the pass class of a Student.		
14.		Fibonacci series upto n.		
15.		Shell script to count the number of vowels of a string.		
16.		Shell script to check no. of lines, words, characters in a file.		
17.		Write a C/C++ program to that outputs the contents of its environment list.		
18.		Write a C/C++ program to emulate unix in cmd.		
19.		Write a C/C++ program that posix compliant programs that print posix defined.		
20.		Write a C/C++ program which demo interprocess communication b/w a reader process & writer process. use mknod, open, read & write, close apis in program		

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

```
→#!/bin/sh
echo "Enter a year"
read year
if [ $(year % 4) -eq 0 -a $(year % 100) -ne 0 ]
then
    echo "Leap year"
else
    if [ $(year % 400) -eq 0 ]
    then
        echo "Leap year"
    else
        echo "Non leap year"
    fi
fi
```

Teacher's Signature : \_\_\_\_\_

OBSERVATION:

Enter a year:

300

Non Leap year

Enter a year:

400

Leap year

Enter a year:

1964

Leap year.

2) Shell script to find the area of circle.

→ #!/bin/sh

echo "Enter the area of the circle:"

read r

area='expr \$r\\* \$r\\* 3.14 |bc'

echo "Area of the circle is :"

echo \$area.

Teacher's Signature : \_\_\_\_\_

OBSERVATION:

Enter the radius of the circle:

7

Area of Circle is :

153.86

Enter the radius of the circle:

12

Area of Circle is 452

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

→ #!/bin/bash

echo "Enter a number!"

read n

if [ \$n -gt 0 ]

then

echo "positive"

else

if [ \$n -lt 0 ]

then

echo "Negative"

else

echo "Zero"

fi

fi

Teacher's Signature : \_\_\_\_\_

### OBSERVATION!

Enter a number:

21

Positive

Enter a number:

0

Zero

Enter a number:

- 9

Negative.

4.) Shell script to find the biggest of three numbers.

→ #!/bin/sh

echo "Enter the first number:"

read a

echo "Enter the second number:"

read b

echo "Enter the third number:"

read c

if [ \$a -gt \$b ]

then

if [ \$a -gt \$c ]

then

echo \$a

else

echo \$c

fi

else

if [ \$b -gt \$c ]

then

echo \$c

fi

fi

Teacher's Signature : \_\_\_\_\_

OBSERVATION:

Enter a number:

21

Positive

Enter a number:

10

Enter a number:

5

21

5) Shell script to find the factorial of a number.

```
→ #!/bin/sh
echo "Enter the number"
read foo
n=$foo
fact=1
while [ $foo -gt 1 ]
do
fact=$((fact * foo))
foo=$((foo - 1))
done
echo "Factorial of $n is $fact"
```

Teacher's Signature :

## OBSERVATION

Enter the number:

6

Factorial of 6 is 720

Enter the number:

10

Factorial of 10 is 3628800

6) Shell script to compute the gross salary of an employee.

→ #!/bin/sh

echo "Enter basic salary:"

read bsal

hra='expr \$bsal \* 20/100 | bc'

da='expr \$bsal \* 10/100 | bc'

gsal='expr \$bsal + \$hra + \$da | bc'

echo "Gross salary is: \$gsal"

Teacher's Signature : \_\_\_\_\_

OBSERVATION:

Enter basic salary :

5000

Gross salary is : 6500

Enter basic salary:

990

Gross salary is : 1287

f) Shell script to convert the temperature Fahrenheit to celsius.

→ #!/bin/sh

echo "Enter the number or temperature in Fahrenheit"

read fno

c=\$(fno - 32) \* 5/9 )

Echo "\$f fahrenheit is \$c in celsius".

Teacher's Signature :

OBSERVATION!

Enter temperature in Fahrenheit

-40

-40 Fahrenheit is -40 in celsius.

Enter temperature in Fahrenheit

59

59 Fahrenheit is 15 in celsius.

8) Shell script to perform arithmetic operations on given two numbers.

→ `#!/bin/sh`

`echo "Enter first number!"`

`read fno`

`echo "Enter the second number!"`

`read sno`

`echo "Enter number as choice 1. ADD 2. SUBTRACT  
3. MULTIPLY 4. DIVIDE"`

`read choice`

`case $choice in`

`1) ans=$((fno+sno));;`

`2) ans=$((fno-sno));;`

`3) ans=$((fno*sno));;`

`4) ans='expr $fno / $sno | bc';;`

`* ) echo "invalid option"`

`exit 0; ;`

`esac`

`echo "Result of the operation is : $ans"`

Teacher's Signature :

OBSERVATION

Enter first number:

6

Enter second number:

3

Enter number as choice : 1. ADD    2. SUBTRACT    3. MULTIPLY  
                                4. DIVISION

4

Result of the operation is : 2

q) Shell script to find the sum of even numbers upto n.

→ #!/bin/sh  
if [ \$# -eq 1 ]

then

c=2

sum=0

while [ \$c -le \$1 ]

do

sum=\$((sum+c))

c=\$((c+2))

done

echo "Sum of even number till \$1 is! \$sum"

fi

Teacher's Signature : \_\_\_\_\_

OBSERVATION.

· prog 9 - sumeven.sh 8  
Sum of even number till 8 is 20

· prog 9 - sumeven.sh 5  
Sum of even number till 5 is 6

10) Shell script to print the combinations of numbers  
123

```
→ #!/bin/sh
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 -a $i -ne $k -a $j -ne $k ]
then
echo "$i $j $k"
fi
done
done
done
```

Teacher's Signature : \_\_\_\_\_

OBSERVATION.

111

112

113

121

122

123

131

132

211

212

213

221

222

223

231

232

233

311

312

313

321

322

323

331

332

333

ii) Shell script to find the power of a number.

```
→ #!/bin/sh
if [ $# -eq 2 ]
then
temp=$2
num=1
while [ $temp -gt 0 ]
do
num=`expr $num \* $1 | bc`
temp=$((temp-1))
done
echo "The power of $1 to $2 is: $num"
fi
```

Teacher's Signature : \_\_\_\_\_

OBSERVATION.

\$ . prog II - power.sle 3 6  
The power of 3 to 6 is 729

\$ prog II - power.sle 5 to 3 is 125

12) Shell script to find the sum of n natural numbers.

→ #!/bin/sh  
if [ \$# -eq 1 ]

then

c=1

sum=0

while [ \$c -le \$1 ]

do

sum=\$((sum+c))

c=\$((c+1))

done

echo "Sum of natural numbers: \$sum"

fi

Teacher's Signature : \_\_\_\_\_

OBSERVATION

.1 prog 12. sumN.su 10  
Sum of natural numbers: 55.

.1 prog 12. sumN.su 30  
Sum of natural numbers: 465

13) Shell script to display the pass class of a student.

→#!/bin/bash

for ((i=0; i<6; i=i+1))

do

echo "Enter the CIE marks of the student"  
read cie

echo "Enter the SEE marks of the student"

read see

total=\$((cie+see))

case \$total in

100) echo "The Student has scored S grade in subject \$i";;

95[0-9]) echo "The Student has scored A grade in subject \$i";;

8[0-9]) "The Student has scored B grade in subject \$i";;

7[0-9]) "The Student has scored C grade in subject \$i";;

6[0-9]) "The Student has scored D grade in subject \$i";;

5[0-9]) "The Student has scored E grade in subject \$i";;

4[0-9]) "The Student has scored F grade in subject \$i";;

3[0-9]) "The Student has scored F grade in subject \$i";;

2[0-9]) "The Student has scored F grade in subject \$i";;

1[0-9]) "The Student has scored F grade in subject \$i";;

[0-9]) "The Student has scored F grade in subject \$i";;

\*) echo "Invalid number"

esac

if [ \$total -gt 40 ]

then

countpass=\$((countpass+1))

Teacher's Signature : \_\_\_\_\_

else

countfail = \$ ((countfail + 1))

fi.

done

echo " \$countpass subjects passed"

echo " \$ countfail subjects failed"

## OUTPUT

Enter CIE and SEE

marks of subject 1.

23

13

You get F grade

in subject 1

Enter CIE and SEE

marks of subject 2.

11

25

13

You get F grade in subject 2.

Enter CIE

and SEE marks of subject 3.

30

30

You got C grade in subject 3.

Enter CIE and SEE marks of subject 4.

50

49

You got S grade in subject 4

Enter CIE and SEE marks of subject 5.

50

50

You got S grade in subject 5.

Enter CIE and SEE marks of subject 6.

40

42

You got A grade in subject 6.

4 subject passed

2 subject failed

14) Shell script to find the fibonacci series upto n.

→ echo "Enter the numbers to be generated"  
read n

function fib

x=0

y=1

i=2

echo "Fibonacci series upto \$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

}

r='fib \$n'

echo "\$r"

OUTPUT:

Enter the value of n: 5

Ribonacci series upto 5 terms:

0 1 1 2 3

15) Shell script to count the number of vowels of a string

→ #!/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".

Teacher's Signature : \_\_\_\_\_

OUTPUT -

Enter a string  
travel

The number of vowels ?

Enter a string:

aeiou

The number of vowels is 5

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

→ #!/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

Enter file name

Lab5.sh

No. of character is 26.

No. of words is 24.

No. of lines is 10.

17) Write a C/C++ program to that outputs the environment list contents.

```
#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

SSH\_AGENT\_PID = 3207  
HOSTNAME = localhost.localdomain  
DESKTOP\_STARTUP\_ID =  
SHELL = /bin/bash  
TERM = xterm  
HSTSIZE = 1000  
KDE\_NO\_XPM = 1  
XTK\_RC\_FILES = /etc/gtk/gtkrc:/root/.gtkrc-1.2-gnome2,  
WINDOWID = 44040273  
OLDPWD = /root/tan  
USER = user/lib/qt-3.3  
QTINC = user/lib/qt-3.3/include  
USER = root  
LS\_COLORS = no=00;Fi=00;di=00; 34:1  
GNOME\_KEYRING\_SOCKET = /tmp/keyring - vsBNL/socket  
SSH\_AUTH\_SOCK = /tmp/ssh-SEWJHJ3149/agent.3149  
KDEDIR = user  
SESSION\_MANAGER = local/localhost.localdomain: /tmp/.ICE-unix/3149  
MON\_XSERVER\_LOCATION = local  
INPUTRC = /etc/inputrc  
PWD = /root/tan/user  
XMODIFIERS = @im=none  
HOME = /root  
SHLVL = 2  
UNOME\_DESKTOP\_SESSION\_ID = Default  
LOGNAME = root  
QTLIB = user/lib/qt-3.3/lib  
CUS\_RSH = ssh  
DBUS\_SESSION\_ADDRESS = unix:abstract=/tmp/dbusid.Ovjtim  
UBROKEN = gnome-terminal  
COLORTERM = gnome-terminal  
XAUTHORITY = /tmp/.gdm5XU71Uw  
- = /a.out

18) Write a C/C++ program to emulate the Unix ln command.

```
#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");
    }

    if ((link(argv[1], argv[2])) == -1)
        printf("Cannot create hard link\n");
    else
        printf("Hard link created\n");
}

return 0;
}
```

## OUTPUT:

Usage: ./a.out [-s] <log-file> <new-link>  
[root @ localhost uspl] # ./a.out 1234

usage: ./a.out [-s] <log-file> <new-link>  
[root @ localhost uspl] # ./a.out l.c z

Hard Link Created

[root @ localhost uspl] # ls -l  
-rw-r--r-- 2 root root 657 May 27 16:44 lac  
-rw-r--r-- 2 root root 657 May 27 16:44. z

[root @ localhost uspl] # ./a.out l.a.c z

Cannot create hard link

[root @ localhost uspl] # ./a.out -s. la.c zz

symbolic link created

[root @ localhost uspl] # ls -l

-rw-r--r-- 2 root root 657 May 27 16:44 lac  
l.a.c 2 2 root root '4 Apr 1 18:32 -z  
l.a.c

[root @ localhost uspl] # sendlink zz l.a.c.

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 <sys/types.h>
# include <stropts.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 .1.\n",
           _POSIX_CHOWN_RESTRICTED);
    #else
    printf ("System does not support chown_restricted option\n");
    #endif
```

Teacher's Signature: \_\_\_\_\_

```
#endif
```

```
#ifdef _POSIX_NO_TRUNC
```

printf("pathname trunc option is 'l.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 file is 'l.d\\n', \_POSIX\_VDISABLE);

```
#else
```

printf("System does not support \_POSIX\_VDISABLE\\n");

```
#endif
```

```
return 0;
```

```
}
```

Teacher's Signature : \_\_\_\_\_

## OUTPUT.

System supports job control  
System supports shared set-UID and  
shared set-GID

chown-restricted option is 1  
Pathname trunc option is 1

Disable character for terminal files is 0.

20) Write a C/C++ program which demonstrates Interprocess communication between a reader process and a writer process. Use mktfio, open, read, write and close APIs in your program.

```

→ #include <sys/types.h>
# include <unistd.h>
# include <fcntl.h>
# include <sys/stat.h>
# include <string.h>
# include <errno.h>
# include <stropts.h>

int main (int argc, char* argv[])
{
    int fd;
    char buf[256];
    if (argc != 2 && argc != 3)
    {
        printf ("USAGE : $ <file> [<arg>]\n", argv[0]);
        return 0;
    }

    mktfio (argv[1], S_IFIFO | S_IRWXU | S_IRWXG | S_IWXO);
    if (argc == 2)

        fd = open (argv[1], O_RDONLY | O_NONBLOCK);
        while (read (fd, buf, sizeof (buf)) > 0)
            printf ("%.*s", buf);
    }
}

```

Teacher's Signature : \_\_\_\_\_

Expt. No. \_\_\_\_\_

Date \_\_\_\_\_

Page No. \_\_\_\_\_

}

else

L

fd = open (argv[1], O\_WRONLY);

write (fd, argv[2], strlen (argv[2]));

}

close (fd);

}

Teacher's Signature : \_\_\_\_\_

## Output.

[root@localhost uspl]# ./a.out FIFO1 "This is  
VSP & CD Lab"

After this open new Terminal by pressing  
Shift + Ctrl + N or Go to file → open  
Terminal.

[root@localhost ~]# ./a.out FIFO1  
This is VSP & CD Lab.