

# Particulars of the Experiments Performed

## CONTENTS

Expt No.	Date	Experiment	Marks Obtained	Page No.
1.	25/10/2021	Shell Script to find if the given year is Leap or not		02-03
2.	25/10/2021	Shell Script to find the area of a circle		4-5
3.	25/10/2021	Shell Script to check whether the number is zero/positive/negative		6-7
4.	25/10/2021	Shell Script to find the biggest of three numbers	100/-	8-9
5.	08/11/2021	Shell Script to find the factorial of a number	8/11/2021	10-11
6.	08/11/2021	Shell Script to Compute the gross salary of an employee		12-13
7.	08/11/2021	Shell Script to Convert the temperature Fahrenheit to Celsius		14-15
8.	08/11/2021	Shell Script to perform arithmetic operations on given two numbers	100/-	16-17
9.	15/11/2021	Shell Script to find the Sum of even numbers upto n	15/11/2021	18-19
10.	15/11/2021	Shell Script to print the Combinations of no.s 123		20-21
11.	15/11/2021	Shell Script to find the pow of a number		22-23

# Particulars of the Experiments Performed

## CONTENTS

Expt No.	Date	Experiment	Marks Obtained	Page No.
12.	15/11/2021	Shell Script to find the sum of n natural nos.	92/100 15/11/21	24.25
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 the unix ln cmd		
19.		Write a C/C++ program posix complaint program that prints the POSIX defined Configuration options Supported		
20.		Write a C/C++ program which demo interprocess communication b/w a reader process & a writer process. Use mkfifo, open, read & write, closeapis in your program		

To find if the given year is leap or not

#!/bin/sh

echo "Enter the year"

read year

if [ \$(year % 400) -eq 0 ]

then

echo "leap year"

elif [ \$(year % 100) -eq 0 ]

then

echo "\$year"

echo "Not a leap year"

elif [ \$(year % 4) -eq 0 ]

then

echo "\$year is a leap year"

else

echo "Not a leap year"

fi

↓  
5/10/21

Output :

Enter the year  
2019

2019 is Not a leap year

Enter the year  
2016

2016 is a leap year



## Shell Script to find area of a circle

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

```
echo "Enter the radius of circle"
```

```
read radius
```

```
area='expr 22/7 *$radius * $radius'
```

```
echo "Area of circle is $area"
```

✓  
25/10/21

Teacher's Signature:

OUTPUT:

Enter the radius of circle

12

Area of Circle is 432

Enter the radius of circle

30

Area of Circle is 2826

→ Shell Script to find/check whether the number is positive/negative or zero

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

```
echo "Enter the number"
```

```
read num
```

```
if [ $num -eq 0 ]
```

```
then
```

```
echo "$num is zero"
```

```
elif [ $num -gt 0 ]
```

```
then
```

```
echo "$num is positive"
```

```
else
```

~~echo "\$num is negative"~~

```
fi
```

N  
25/10/21

Output :

— Enter the number

-4

-4 is negative

— Enter the number

0

0 is zero

— Enter the number

9

9 is positive



Shell Script to find the biggest of three numbers

#!/bin/bash

big = 0

echo "Enter 3 numbers"

echo "num1 = "

read a

echo "num2 = "

read b

echo "num3 = "

read c

if [[\$a -gt \$b] -a [\$a -gt \$c]]

then

echo "\$a is greatest"

elif [[\$b -gt \$a] -a [\$b -gt \$c]]

then

echo "\$b is greatest"

else

echo "\$c is greatest"

✓  
25/10/21

Output:

Enter 3 numbers

num1

5

num2

3

num3

1

5 is greatest

Enter 3 numbers

num1

-3

num2

11

num3

0

11 is greatest



Shell Script to find the factorial of a number

```
#!/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"

✓  
B11/11

OUTPUT:

Enter n

5

factorial of 5 is 120

Enter n

10

factorial of 10 is 3628800

Shell Scr

#!/bin

echo "

read

Fact

while

do

Fact

n =

don

echo

fi

done

Q) To Compute Gross salary of an employee

#!/bin/bash

read basic

DA='expr \$basic \\* 20/100'  
echo "DA is \$DA"

HRA='expr \$basic \\* 20/100'  
echo "HRA is \$HRA"

Gross='expr \$basic + \$DA + \$HRA'

echo "GROSS SALARY IS \$GROSS" | bc

N  
8/11/21

OUTPUT:

Enter the basic  
4500

DA is 450  
HRA is 900

GROSS SALARY IS 5850

Enter the basic  
990

DA is 99  
HRA is 198

GROSS SALARY IS 1287

To Compute	#!/bin/ read bc DA='expr echo " HRA='e echo " gross = echo " N split
------------	---

→ Shell Script to Convert Fahrenheit to Celsius

#!/bin/bash

echo "Enter the temperature in Fahrenheit"  
read f

c1='expr \$f - 32'

c2='expr \$c1 \\* 5/9 | bc'

echo "Temperature in degree. celsius \$c"

✓  
S  
8/11/21

OUTPUT:

Enter temperature in fahrenheit  
59

Temperature in degree celsius 15

Enter temperature in fahrenheit  
-40

Temperature in degree celsius -40

## Shell Script for Compute Arithmetic Operations

```
#!/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 "SUBTRACTION"
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:

Enter two numbers

4

6

ADDITION

addition is 10

SUBTRACTION

subtraction is -2

MULTIPLICATION

multiplication is 24

DIVISION

division is 0

---

Enter two numbers

100

1

ADDITION

addition is 101

SUBTRACTION

subtraction is 99

MULTIPLICATION

multiplication is 100

DIVISION

division is 100

N  
still

Q) find Sum of Even Numbers

#!/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/2 Even Numbers is \$sum"

✓  
→ 1/2

OUTPUT:

enter the n:

10

Sum of 5 Even Numbers is 30

enter the n:

5

Sum of 2 Even Numbers is 6

To Print Combinations of 123

#!/bin/sh

echo "The Combinations 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
```

N  
11/21

## OUTPUT :

The Combinations of 123 are

1 1 1

1 1 2

1 1 3

2 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



## To find the Power of Numbers

#!/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

N  
echo "\$num ^ \$pow = \$res"

OUTPUT :

Enter number :

2

input power :

5

$$2^5 = 32$$

Enter number :

5

input power :

3

$$5^3 = 125$$

To find Sum of Natural Numbers

#!/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"

N  
15/11/21

Expt. No.

OUTPUT :

Enter n  
10

SUM OF 10 NATURAL NUMBERS IS 55

Enter n  
30

SUM OF 30 NATURAL NUMBERS IS 465

10

#

ec

hi

i

si

t

w

c

o

u

✓

N

15

TM

Shell Script to display the pass class of a student

#!/bin/bash

```
for (( i=1 ; i<=6 ; i++ ))
do
echo "Enter CIE and SEE marks of subject $i : "
read cie
read see
marks = $((cie + see))
```

case \$marks in

```
100) echo "you got S grade in Subject $i";;
9[0-9]) echo "you got S grade in Subject $i";;
8[0-9]) echo "you got A grade in Subject $i";;
7[0-9]) echo "you got B grade in subject $i";;
6[0-9]) echo "you got C grade in Subject $i";;
5[0-9]) echo "you got D grade in Subject $i";;
4[0-9]) echo "you got E grade in Subject $i";;
3[0-9]) echo "you got F grade in Subject $i";;
2[0-9]) echo "you got F grade in Subject $i";;
1[0-9]) echo "you got F grade in Subject $i";;
[0-9]) echo "you got F grade in Subject $i";;
```

\*) echo "not valid marks";;

esac

```
if [ $marks -gt 40 ] ; then
countp= $((countp+1))
```

else

countf = \$((countf + 1))

fi

done

echo "\$countp subjects passed"

echo "\$countf subjects failed"

OUTPUT :

Enter CIE and SEE marks of Subject 1 :

23

13

You got F grade in Subject 1

Enter CIE and SEE marks of Subject 2 :

11

25

You got F grade in Subject 2

Enter CIE and SEE marks of Subject 3 :

10

10

You got F grade in Subject 3

Enter CIE and SEE marks of Subject 4 :

30

30

You got C grade in Subject 4

Enter CIE and SEE marks of Subject 5

50

49

You got S grade in Subject 5

Enter CIE and SEE marks of Subject 6

50

50

You got S grade in Subject 6

3 subjects passed

3 subjects failed

Q) Fibonacci series upto n

#!/bin/sh

echo "Enter the value of n : "

read n

x=0

y=1

z=2

echo "Fibonacci Series up to \$n terms : "

echo "\$n"

echo "\$y"

while [ \$i -lt \$n ]

do

i=`expr \$i + 1`

z=`expr \$n + \$y`

echo "\$z"

n=\$y

y=\$z

done

OUTPUT:

Enter the value of n:  
5

Fibonacci Series upto to 5 terms  
0 1 1 2 3

→ Shell Script to count the number of vowels of a String

```
#!/bin/bash
echo "Enter a string"
read string
Count=0
l='expr "$string": +.*'
for (( i=0; i<$l; i++ ))
do
    c='exprs "$string": '\`(.|\n)\'"
    if [ "$c" = 'a' -o "$c" = 'e' -o "$c" = 'i' -o "$c" = 'o'
        -o "$c" = 'u' ]
    then
        Count=$((Count+1))
    fi
    string='expr "$string": .\`(.|\n)\`"'
done
echo "The numbers of vowels : $Count"
```

OUTPUT:

Enter a string  
travel

The numbers of vowels 2

---

Enter a string  
aeiou

The numbers of vowel is 5



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

#!/bin/sh

echo "Enter 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 the file name

Lab5.sh

No. of character is 126

No. of words is 24

No. of lines is 10

Q) Write a C/C++ program to that outputs the contents of its environment list

```
#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  
HISTSIZE = 1000  
KDE\_NO\_IPV6 = 1  
GTK\_RC\_FILES = /etc/gtk/gtkrc:/root/.gtkrc-1.2-gnome2  
WINDOWID = 44040273  
OLDPWD = /root/tan  
QTDIR = /usr/lib/qt-3.3  
QTINC = /usr/lib/qt-3.3/include  
USER = root  
LS\_COLORS = no=00:fi=00:di=00;34:  
GNOME\_KEYRING\_SOCKET = /tmp/keyring-VSDBVL/socket  
SSH\_AUTH\_SOCK = /tmp/ssh-SENJH3149/agent.3149  
KDEDIR = /usr  
SESSION\_MANAGER = local/localhost.localdomain:/tmp/.ICE-unix/349  
GDM\_XSERVER\_LOCATION = local  
INPUTRC = /etc/inputrc  
PWD = /root/tan/upl  
XMODIFIERS = @im=none  
HOME = /root  
SHLVL = 2  
GNOME\_DESKTOP\_SESSION\_ID = default  
LOGNAME = root  
QTLIB = /usr/lib/qt-3.3/lib  
CVS\_RSH = ssh  
DBUS\_SESSION\_BUS\_ADDRESS = unix:abstract=/tmp/dbus-idOVjtmg  
GBROKEN\_FILERAMES = 1  
COLORTERM = gnome-terminal  
XAUTORITY = /tmp/.gdm5X710W  
- = ./a.out

→ Write a C/C++ program to emulate the unix ls 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
```

```
{// If ( link ( argv [ 1 ] , argv [ 2 ] ) == -1 )
```

```
    printf ( "Symbolic link Created" );
}
```

```
else {
```

```
    if ( link ( argv [ 1 ] , argv [ 2 ] ) == -1 )
```

```
        printf ( "Cannot resolve/create hard links\n" );
    else
```

```
        printf ( "Hard link Created" );
    }
```

```
}
```

```
return 0;
```

```
}
```

## OUTPUT:

Usage : ./a.out [-s] <arg-file> <new-link>  
[root@localhost uspl]# ./a.out 1 2 3 4

Usage: ./a.out [-s] <arg-file> <new-link>  
[root@localhost uspl]# ./a.out 1.c 2  
Hard link created

[root@localhost uspl]# ls -l  
-rw-r--r-- 2 root root 657 Mar 27 16:44 lac  
-rw-r--r-- 2 root root 657 Mar 27 16:44 .2  
[root@localhost uspl]# ./a.out 1.a.c 2  
Cannot create hard link

[root@localhost uspl]# ./a.out -s 1.a.c 2  
Symbolic link created

[root@localhost uspl]# ls -l  
-rw-r--r-- 2 root root 657 Mar 27 16:44 lac  
lrwxrwxrwx 1 root root 854 Apr 1 18:32 2  
1.a.c

[root@localhost uspl]# readlink 2 1.a.c

→ 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 <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 -wide pathname
        trunc option\n");
#endif
#ifndef _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:

System Supports job control  
System Supports saved set-01D and saved set-91D

Chown-restricted option is 1

Pathname trunc option is 1

Disable character for terminal files is 0

→ 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/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_IROTH);
    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);  
    while (Fd, argv[2], strlen (argv[2]));

}

    close (Fd);

}

OUTPUT :

[root@localhost uspl] # ./a.out FIFO1 "This is usp & 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 USP & CD lab