OUTPUT:-

Enter the year:
2016
It is a leap year

Enter the year:
200
Not a leap year

Enter the year
800
The given year is leap year

Enter the year
2021
It is not a leap year

Expt. No. 1 Date 25/10/2021

Page No. 1

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

Ans:

```
#!/bin/sh
echo "Enter the year"
read year
if [ echo "year %.400" | bc -eq 0 ]
then
  echo "the given year is leap year"
elif [ echo "year %.4" | bc -eq 0 ]
then
   echo "Not a leap year"
else
  echo "It is a leap year"
fi
else
  echo "It is not a leap year"
fi
```

Teacher's Signature :

OUTPUT :-

Enter the reading
5

78.53

Expt. No. 2 Date 23.10.2021

Page No. 8

2) Shell script to find area of a circle.

Ans:

```
#!/bin/bash
echo "Enter the radius :"
read rad
pi = 3.142
area = $(echo "$pi * $rad * $rad" | bc)
echo $area
```

Teacher's Signature :

OUTPUT:-

Enter any number:
-1
Number is negative

Enter any number:
0
Number is zero

Enter any number:
7
Number is positive

Expt. No. 3 Date 24/10/2021

Page No. 3

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

```
#!/bin/sh
echo "Enter any number:"
read num
if [ $num -eq 0 ]
then
echo "Number is zero"
elif [ $num -lt 0 ]
then
  echo "Number is negative"
else
 echo "Number is positive"
fi
```

Teacher's Signature :

OUTPUT :-

Enter first number :
3
Enter second number :
1
Enter third number :
7
7 is the greatest

Expt. No. 4 Date 25/01/2021

Page No. 4

4) Shell script to find biggest of 3 numbers.

Ans:

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

if [ $a -ge $b -a $a -ge $c ]
then
    echo "$a is the greatest"
elif [ $b -ge $a -a $b -ge $c ]
then
    echo "$b is the greatest"
else
    echo "$c is the greatest"
fi
```

Teacher's Signature :

OUTPUT :-

Enter a number :

5

Factorial of a number 5 : 120

Expt. No. 5 Date 8/11/2021

Page No. 5

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

Ans:

```
#!/bin/sh
echo "Enter a number:"
read num
n=1
while [ $num -ge 1 ]
do
n=$((n*num))
num=$((num-1))
done
echo "Factorial of a number $num: $n"
```

Teacher's Signature :

OUTPUT:-

Enter basic salary :

25000

Gross salary = 43750

Expt. No. 6

Date 8/11/2023

Page No. 6

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

```
#!/bin/bash
echo "Enter basic salary:"
read bs
hra = $(echo "0.5 * $bs" | bc)
da = $(echo "scale=2; 0.25 * $bs" | bc)
gs = $(echo "scale=2; $hra + $da + $bs" | bc)
echo "Gross salary = $gs"
```

Teacher's Signature :

OUTPUT :-

Enter temperature in fahrenheit :

112

Temperature in celsius : 44.44

Expt. No. 7 Date 21/11/2023

Page No. 7

7) Shell script to convert the temperature Fahrenheit to Celsius.

Ans: #!/bin/sh

echo "Enter temperature in fahrenheit:"

read fh

temp=$(echo "scale=2; (5/9)*($fh-32)" | bc)

echo "Temperature in celsius: $temp"

Teacher's Signature :

OUTPUT:

Enter first number:

5

Enter second number:

3

ADD: 5+3 = 8

SUBTRACT: 5-3 = 2

MULTIPLICATION: 5 * 3 = 15

DIVISION: 5/3 = 1.33

REMAINDER: 2

Expt. No. 8 Date 8/11/2021

Page No. 8

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

```
#!/bin/bash
echo "Enter first number:"
read num1
echo "Enter second number:"
read num2
echo "ADD $num1 + $num2"
add=$(echo "$num1+$num2" |bc)
sub=$(echo "$num1-$num2" |bc)
mul=$(echo "$num1*$num2" |bc)
div=$(echo "scale=2; $num1/$num2" |bc)
rem=$(echo "$num1%$num2" |bc)
echo "ADD: $num1+$num2=$add"
echo "SUBTRACT: $num1-$num2=$sub"
echo "MULTIPLICATION: $num1*$num2=$mul"
echo "DIVISION: $num1/$num2=$div"
echo "REMAINDER: $num1%$num2=$rem"
```

Teacher's Signature :

OUTPUT :-

Enter the number :

10

Sum of even numbers upto 10 : 30

Expt. No. 9 Date 15/11/2021

Page No. 9

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

Ans:

```
#!/bin/sh
echo "Enter the number:"
read n
i=0
sum=0
while [ $i -le $n ]
do
    sum = $((sum + i ))
    i = $(( i +1))
done
echo "Sum of even numbers upto $n : $sum"
```

Teacher's Signature :

OUTPUT:

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

Expt. No. 10 Date 15/11/2021

Page No. 19

1a) Shell script to print the combinations of 123

```
#!/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
```

Teacher's Signature :

OUTPUT:-

Enter the number:

5

Enter the power:

3

5 ^ 3 : 125

Expt. No. 11 Date 15/11/2023

Page No. 11

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

Ans:

```
#!/bin/sh
echo "Enter the number:"
read num
echo "Enter the power:"
read pow
i=1
res=1
while [ $i -le $pow ]
do
    res=$(( res*num ))
    i=$((i+1))
done
echo "$num ^ $pow is $res"
```

Teacher's Signature :

OUTPUT:-

Enter the number:

10

Sum of first 10 natural numbers : 55

Expt. No. 12 Date 15/10/2021

Page No. 12

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

Ans:
```
#!/bin/sh
echo "Enter the number:"
read n
i=1
sum=0
while [ $i -le $n ]
do
    sum=$((sum+i))
    i=$((i+1))
done
echo "Sum of first $n natural numbers: $sum"
```

15/12/2021

OUTPUT:-

Enter cie marks of subject1:- 45

Enter see marks of subject1:- 80
Grade: A, Passed

Enter cie marks of subject2:- 35
Enter see marks of subject2:- 65
Grade: C, Passed

Enter cie marks of subject3:- 40
Enter see marks of subject3:- 60
Grade: B, Passed

Enter cie marks of subject4:- 50
Enter see marks of subject4:- 95
Grade: S, Passed

Expt. No. 13 Date 29-11-2021

Page No. 13

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

Ans:-

```
#!/bin/bash
i=1
n=5
while [ $i -le $n ]
do
    echo "Enter cie marks of subject $i"
    read cie
    echo "Enter see marks of subject $i"
    read see
    marks = $((cie+see/2))
    i= $((i+1))
    if [ $marks -ge 90 ]
    then
        echo "Grade: S, Passed"
    elif [ $marks -ge 80 ]
    then
        echo "Grade : A, Passed"
    elif [ $marks -ge 60 ]
    then
        echo "Grade: C, Passed"
    elif [ $marks -ge 40 ]
    then
        echo "Grade: E, Passed"
    else
        echo "Grade: F, Failed"
    fi
done
```

Teacher's Signature :

OUTPUT :

Enter the value of n : 5

0

1

1

2

3

Expt. No. 14 Date 29-11-2021

Page No. 14

14 Shell script to find the Fibonacci series upto n.

Ans:

```
#!/bin/bash
echo "Enter the value of n:"
read n
f=1
s=0
echo "0"
echo "1"
i=2
while [ $i -le $n ]
do
    echo "$f+$s" |bc
    t=$((f))
    t=$((s+f))
    s=$((t))
    i=$((i+1))
done
```

Teacher's Signature :

OUTPUT:

Enter the string

Boston

Count of vowels: 2

Expt. No. 15 Date 29-11-2021

Page No. 15

15 Shell script to count the number of vowels of string

Ans:-

```
#!/bin/sh
echo "Enter the string :"
read str
len=$(expr length $str)
count=0
while [ $len -gt 0 ]
do
ch=$(echo $str | cut -c $len)
case $ch in
[aeiouAEIOU])
count=$((count+1))
;;
esac
len=$((len-1))
done
echo "Count of vowels: $count"
```

Teacher's Signature :

Lab 43.sh:

Hi Hello

Hello World

OUTPUT:

Number of lines: 2

Number of words: 4

Number of character: 17

Expt. No. 16 Date 29-11-2021

Page No. 16

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

Ans:

```
#!/bin/sh
file_path = "/home/bmsce/lab93.sh"
number_of_lines = `wc -lines < $file_path`
number_of_words = `wc -word < $file_path`
number_of_characters = `wc --char < $file_path`

echo "Number of lines : $number_of_lines"
echo "Number of words: $number_of_words"
echo "Number of characters : $number_of_character"
```

Teacher's Signature :

OUTPUT:

SSH_AGENT_PID = 3207

HOSTNAME = Local host . Local domain

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 = 44044273

OLDPWD = /root/tom

GTDIR = /usr/lib/gt-3.3

GTINC = /usr/lib/gt-3.3/include

USER = root

LS_COLORS = no=00: fi=00: di=00;34: l

GNOME_KEYRING_SOCKET = /tmp/keyring-vcDBVL/socket

SSH_AUTH_SOCK = /tmp/ssh-SEWJHJ3149/agent.3149

KDEDIR = /usr

SESSION_MANAGER = local/localhost.localdomain:/tmp/.ICE-unix/31

MAIL = /var/spool/mail/root

DESKTOP_SESSION = default

PATH = /usr/lib/gt-

GDM_XSERVER_LOCATION = local

INPUTRC = /etc/inputrc

.

.

.

)

Expt. No. 17 Date 3-1-2022

Page No. 17

17. Write a C/C++ program to that outputs the contents of its Environment list

Ans:

```
#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;
}
```

Teacher's Signature :

OUTPUT:-

```
[root@localhost usp1] # ./a.out 1 2 3 4
Usage : ./a.out [-s] <org file> <new - link>
[root@localhost usp1] # ./a.out 1.c 2
Hard link created

[root@localhost usp1] # ls -l
-rw-r--r-- 2 root root 657 Mar 27 16:44 1.c
-rw-r--r-- 2 root root 657 Mar 27 16:44 2
```

Expt. No. 18 Date 3-1-2023

Page No. 13

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

Ans:

```
#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; }
```

Teacher's Signature :

OUTPUT :-

System supports job control
System supports saved set-UID and saved set-GID
Chown-restricted option is 1
Pathname trun option is 1
Disable character for terminal files is 0

Date 3-1-2024

Expt. No. 19

Page No. 19

19. Write a C/C++ POSIX compliant program that prints the POSIX defined configuration options supported on any given system using feature test macros.

Ans
```
#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
#if def_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
#if def_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);
```

Teacher's Signature :

```
#else
printf ("System does not support wide pathnames traverse option \n");
#endif
#ifdef _POSIX_VDISABLE
printf ("Disable character for terminal files is %d \n", _POSIX_
                                                    VDISABLE);
#endif
return 0;
}
```

Teacher's Signature :

OUTPUT :-

[root@localhost usp1)# ./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

Date 3-1-2022

Expt. No. 20

Page No. 21

20. Write a C/C++ program which demonstrates interprocess communication between a reader process and a writes process. Use mkfifo, open, read, write and close API's in your program.

Ans:-

```
#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 <pipe> [<msg>]\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);
```

Teacher's Signature :

Date 3-1-2020

Expt. No. 20 Page No. 22

```
else
{
fd = open (argv[1], O_WRONLY);
write (fd, argv[2], strlen (argv[2]));
}
close (fd);
}
```

Teacher's Signature :