1. Output-

Year:
2024
Year is leap

Year:
2001
Year is not leap.

2. Output-

enter circles radius
5
Area is :120

Date 25/10/4

Expt. No. 01, 02 Page No. 1

1. Experiment - Shell script to find if given year is leap or not.

```
echo "Year:"
read leap
if [ $(( leap %. 400)) - eq 0]
then echo "Year is leap"
elif [ $(( leap %. 100)) - neo ]
then echo "Year is not leap"
elif [ $(( leap %. 4)) - eq 0 ]
then echo "Year is leap"
else
echo "Year is not leap"
fi
```

2. Experiment - Shell script to find the area of circle

```
echo "Enter circles radius"
read rad
pi = 3.14
area =' echo $pi \* $rad \*$rad 'bc'
echo "Area is: $area"
```

Teacher's Signature : _____

3. Output - Enter a number
3
Positive

Enter a number
0
Zero

Enter a number
-3
Negative

Enter a number
0
Zero
Enter a number
-3
Negative

4. Output- Enter first number:
12
Enter second number:
20
Enter third number
14

20 is greatest

Expt. No. 3,4 Date 25/10/21

Page No. 2

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

```
echo "Enter a number:"
read no
if [ $no -eq 0 ]
then echo "Zero"
elif [ $no -gt 0 ]
then echo "Positive"
else
echo "Negative
fi
```

4. Experiment – Shell script to find the biggest of 3 numbers.

```
echo "Enter first number:"
read n1
echo "Enter second number:"
read n2
echo "Enter third number
read n3
if [ $n1 -gt $n2 -a $n1 -gt $n3 ]
then echo "$n1 is greatest"
elif [ $n2 -gt $n1 -a $n2 -gt $n3 ]
then echo "$n2 is greatest"
else
echo "$n3 is greatest"
fi
```

Teacher's Signature : _____

5. Output – Enter a number
5
120

6. Output – Enter basic salary
10000
Gross salary = 13000

Date 8/11/21

Expt. No. 5,6 Page No. 3

5. Experiment – Shell script to find factorial of a number

```
echo " Enter a number: "
read num
fact = 1
while [ $num -gt 1 ]
do
   fact = $(( fact *num ))
   num = $(( num-1 ))
   done
   echo $ fact
```

6. Experiment – Shell script to calculate the gross salary of an employee

```
echo " Enter basic salary "
read basic
DA = `expr $ basic \* 10/100`
hra = `expr $ basic \* 20/100`
gross-sal = `expr $ basic + $DA + $ hra`
 echo " Gross salary = $gross-sal "
```

Teacher's Signature : _____

7. Output - Enter the temp in F

32

32F in celsius is 0C

8. Output - Enter 2 numbers

10 20

Enter the operation + or - or / or * or %

/

Divide: 10/20 = 0.5

Date 8/11/21

Expt. No. 07, 08 Page No. 4

7. Experiment – Shell script to convert the temperature Fahrenheit to

```
echo "Enter the temp. in F"
read f
c = `echo "scale=2; ($f-32)*(5/9)" | bc`
echo "$f F in celsius is $c C"
```

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

```
echo "Enter 2 numbers:"
read x y
echo "Enter operation + or - or / or * or %."
read ch
case $ch in
+) echo "Add: $x+$y ="; echo $x+$y | bc ;;
-) echo "Subtract: $x-$y ="; echo $x-$y | bc ;;
/) echo "Divide: $x/$y ="; echo "scale=2; $x/$y" | bc ;;
*) echo "Multiply: $x*$y ="; echo $x\*$y | bc ;;
%) echo "Modulus: $x%$y ="; echo $x%$y | bc ;;
*) echo "Error"
esac
```

Teacher's Signature : _____

9) Output — Enter a number
5
6

10. Output —

1 2 3
1 3 2
2 1 3
2 3 1
3 1 2
3 2 1

Date 15/11/21

Expt. No. 9,10 Page No. 5

9) Experiment - Shell script to find sum of even numbers to n

```
echo "Enter a number"
read n
sum=0

for((i=0; i<=$n ; i=i+2))
 do
    sum=$((sum+i))
done
echo $sum
```

10.

```
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 :

11. Output—

Enter base
3
Enter power
3
27

12. Output —

Enter a number
5
15

Date 15/11/21

Expt. No. 11, 12

Page No. 6

11. Experiment – Shell script to find the power of a number

```
echo "Enter base"
read b
echo "Enter power"
read p
sol=1
while[ $p -ge 1 ]
do
   sol= $( echo = "scale=2; $sol \* $b " | bc )
   p = $((p-1))
done
echo $sol
```

12. Experiment – Shell script to find sum of n natural numbers

```
echo "Enter a number"
read n
sum=0
for((i=0; i<= $n; i++)
do
   sum = $((sum+i))
done
echo $sum
```

Teacher's Signature : _____

13. Output:

Enter the marks of student in subject 1
85
Enter the marks of student in subject 2
90
Enter the marks of student in subject 3
95
Grade A

Expt. No. 13 Date 29/11/21

Page No. 7

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

```
#!/bin/bash
echo "Enter the marks of student in subject 1"
read m1
echo "Enter the marks of student in subject 2"
read m2
echo "Enter the marks of student in subject 3"
read m3
sum=`expr $m1 + $m2 + $m3`
avg=`expr $sum/3`
if [ $avg -gt 90 ]
then
    echo "Grade S"
elif [ $avg -gt 80 -a $avg -le 90 ]
then
    echo "Grade A"
elif [ $avg -gt 65 -a $avg -le 80 ]
then
    echo "Grade B"
elif [ $avg -gt 40 -a $avg -le 65 ]
then
    echo "Grade C"
elif [ $avg -lt 40 ]
then
    echo "fail"
fi
```

sh pgm.sh

Teacher's Signature :

14. Output — Enter positive value

5

Fibonacci Series

0

1

1

2

3

5

Date 29/11/21

Expt. No. 14 Page No. 8

14 Experiment - Shell script to find fibonacci series upto n.

```
#!/bin/sh
a1=0
a2=1
echo "Enter positive value"
read n
echo "Fibonacci series:"
if [ $n -eq 1 ]
then
    echo "$a1"
elif [ $n -eq 2 ]
then
    echo "$a1 \n $a2"
while [ $n -gt 2 ]
do
    a3=`expr $a1 + $a2`
    echo "$a3"
    a1=$a2
    a2=$a3
    n=`expr $n-1`
done
```

15. Output - Enter a line of text
There are 10,000 number of lines of a code in different variable.

The given string has 21 vowels, 22 consonants and
5 numbers in 1.

16. Output of the program -

Enter the file name
lab 626.sh
Number of characters in lab 626.sh is 51
Number of words in lab 616.sh is 13
Number of lines in lab 616.sh is 3

Expt. No. 15, 16 Date 29/11/21

Page No. 9

15. Experiment - Shell script to count number of vowel of a string.

```
echo -n "Enter line of text"
read string
num_count= $(echo $String |grp -O "[0-9]"|wc --liner)
vow_count= $( echo $ string |grp -O* -i "[aeiou]"|wc --liner)
cons_count= $(echo $string |grp -O -i["dfghjklmnpqrstvwxyz"]
echo "The given string has $vow_count vowels, $cons_count consonats
        and $num_count numbers in it.
```

16. Experiment - Shell script to check number of lines, words, character in a file.

```
echo " Enter the file name"
read lab 616. sh
c =\ cat $file |wc - c`
w = ` cat $ file |wc - w`
l = ` grip - c ". " $ file `
echo "Number of characters in $file in $c "
echo " Number of words in $file in $w"
echo " Number of lines in $file is $ l"
```

lab 616. sh

```
this is line 1
line 2 is given here
this is line 3
```

Teacher's Signature : _____

Expt. No. 17 Date 3/1/22

Page No. 10

17. Experiment - Write a C/C++ program to that outputs the content 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: Usage: ./a.out [-s] <org_file> <new_link>
[root@localhost usp1]# ./a.out 1234
Usage: ./a.out [-s] <org_file> <new_link>
[root@localhost usp1]# ./a.out 1.cg
Hard link created

[root@localhost usp1]# ls -l
-rw-r--r--2 root root 657 Mar 27 16:44 la.c
-rw-r--r--2 root root 657 Mar 27 16:442 (Bolded columns are hardlink count & inode number respectively)

[root@localhost usp1]# ./a.out la.c2
Cant create hard link (Because 3 already exists)

[root@localhost usp1]# ./a.out -s la.c22
Symbolic link created

[root@localhost usp1]# ls -l
-rw-r--r--2 root root 657 Mar 27 16:44 la.c
lrwxrwxrwx 1 root root 4 Apr 1 18:32 22 ->
la.c

[root@localhost usp1]# readlink 22
la.c

Date 3/1/22

Expt. No. 18 Page No. 11

18. Experiment – Write 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("Cant create symbolic link\n");
  else
  printf("Symbolic link created\n");
  }
  else
  {
  if((link(argv[1],argv[2])==-1)
  printf("Cant create hard link\n");
  else
  printf("Hard link created\n");
  }
  return 0;
}
```

Teacher's Signature :

19. Output : System supports Job control
System supports saved set-UID and saved set-GID
Ehown- restricted option is 1
Pathname trunc option is 1
Disable character for terminal file is 0.

Date 3/1/22

Expt. No. 19 Page No. 12

19. Experiment - Write a C/C++ POSIX complaint program that print 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 doesn't 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 doesn't support chown-restricted option\n");
#endif
#if def _POSIX_NO_TRUNC
```

Teacher's Signature : _____

Date 3/1/21

Expt. No. 19 Page No. 13

```
printf ("Pathname trunc option is %d\n", _POSIX_NO_TRUNC);
#else
printf ("System doesnt support system-wide pathname trunc option \n");
#endif
#if def _POSIX_VDISABLE
printf ("Disable character for terminal file is %d \n", _POSIX_Vdisable);
#else
printf ("System doesnt support _POSIX_VDISABLE \n");
#endif
 return 0;
}
```

20. Output - ~~/* Terminal 1~~ [root@localhost usp1]#./a.out FIFO1 "Hi is USP and CP lab"

After this, open new terminal by pressing shift+ctrl+N or go to File → Open Terminal

[root@localhost /]#./a.out FIFO1
Hi is USP & CP Lab

Date 3/1/21

Expt. No. 20

Page No. 14

20. Experiment - Write a C/C++ program to demonstrate interprocess communication between a reader process and writer process. Use mkfifo, open, read, write and close API's

```
#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 buff[250];
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)
{
fd = open(argv[1], O_RDONLY|O_NONBLOCK);
while (read(fd, buf, sizeof(buff))>0)
printf("%s", buf);
}
```

Teacher's Signature : _____