Particulars of the Experiments Performed

CONTENTS

Expt No.	Date	Experiment	Marks Obtained	Page No
1	25-10-21	Shell script to find if year is leap or not		
2	25-10-21	Shell script to find area of circle		
3	25-10-21	Shell script to find if number is 0, +ve or -ve		
4	25-10-21	Shell script to find greatest of three number	2/11/21	
5	2-11-21	Find factorial of a given number		
6	2-11-21	Compute Gross salary		
7	2-11-21	Convert temperature Fahrenheit to celsius		
8	2-11-21	Perform arithmetic operations on two given numbers	16/11/21	
9	16-11-21	Sum of even numbers upto n		
10	16-11-21	Show all combinations of 123		
11	16-11-21	Find power of a number		
12	16-11-21	Sum of n natural numbers		

Particulars of the Experiments Performed

CONTENTS

Expt No.	Date	Experiment	Marks Obtained	Page No.
13	29/11/21	Shell script to find pass class of a student.		
14	29/11/21	Shell script to find fibonacci series upto n.		
15	29/11/21	Shell script to find no. of vowels in a string		
16	29/11/21	Shell script to find no. of line, characters and words in a file		
17	03/1/22	Write a program that outputs contents of environment list.		
18	3/1/22	Write a program to emulate ls command		
19	3/1/22	Write a program to print POSIX defined configuration using test matches		
20	3/1/22	Write a program to demonstrate interprocess communication between a reader process and a writer process		

Output

i) Enter the year: 400
It is a leap year

ii) Enter the year: 100
It is not leap year

iii) Enter the year: 16
It is leap year

iv) Enter the year: 17
It is not leap year

Date 25-10-21

Expt. No. 1 Page No.

Aim: Shell script to find out if a year is leap or not

```
#!/bin/sh
echo "Enter the year:"
read y
if [ $((y%100)) -eq 0]
then
    if [ $((y%400)) -eq 0 ]
    then
        echo "It is leap year"
    else
        echo "It is not leap year"
    fi
else
    rem = $((y%4))
    if [ $rem -eq 0 ]
    then
        echo "It is leap year"
    else
        echo "It is not leap year"
    fi
fi
```

Teacher's Signature :

Output

Enter radius: 40.4
5124.74

Date 25-10-21

Expt. No. 2

Page No.

Aim → Shell script to find area of circle

```
#!/bin/sh
echo "Enter radius: "
read rad
pi=3.14
echo $pi \* $rad \* $rad /bc
```

25/10/21

Teacher's Signature :

Output

Enter the number: 3
The number is positive

Enter the number: 0
Number is zero

Enter the number: -1
Number is negative

Date 25-10-21

Expt. No. 3

Page No.

Aim → Shell script to find if a number is zero, positive or negative

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

25/10/21

Teacher's Signature :

Output

Enter the numbers : 1 2 3
3 is greatest

Enter the numbers : 1 2 1
2 is greatest

Enter the numbers : 1 1 1
1 is greatest

Date 25-10-21

Expt. No. 4

Page No.

Aim → Shell script to find greatest of three numbers

```
#! /bin /sh
echo "Enter the numbers: "
read a b c
if [ $a -ge $b ]
then
    if [ $a -ge $c ]
    then
        echo "$a is largest"
    fi
elif [ $b -ge $c ]
then
    if [ $b -ge $a ]
    then
        echo "$b is greatest"
    fi
else
    echo "$c is largest"
fi
```

25/10/21

Teacher's Signature :

Output

Enter the number : 5
Factorial is 120

Date 2-11-21

Expt. No. 5 Page No.

Aim → Find factorial of a number.

```
echo "Enter the number: "
read n
result =1
for (( i=1 ; i<= $n ; i++))
do
    result= $(( result * i))
done
echo " factorial is $result"
```

2/11/21

Teacher's Signature :

Output

Enter the basic salary: 1000
Gross salary is 1300

Date 2-11-21

Expt. No. 6 Page No.

Aim → Compute gross salary

Formula → da = 10% of basic salary
hra = 20% of basic salary
gross salary = da + hra + basic salary

```
echo "Enter the basic salary :"
read basic_salary
da = `expr $basic_salary \* 10/100`
hra = `expr $basic_salary \* 20/100`
gross_salary = `expr $basic_salary + $da + $hra`
echo "Gross salary is $gross_salary"
```

2/11/21

Teacher's Signature :

Output

Enter temperature in Fahrenheit to celsius: 98
36.03°C

Date 2-11-21

Expt. No. 7 Page No.

Aim → Convert temperature Fahrenheit to Celsius

```
echo "Enter temperature in Fahrenheit to Celsius: "
read f
c=`echo "scale=2 ; (5/9) * ($f -32)" | bc`
echo $c
```

Teacher's Signature :

Output

Enter the numbers : 3 2
Addition - 5
Subtraction - 1
Multiplication - 6
Division - 1.5

Expt. No. 8 Date 2-11-21

Page No.

Aim → Perform arithmetic operations on 2 given numbers

```
echo "Enter two numbers : "
read n1 n2
add = $((n1 + n2))
sub = $((n1 - n2))
mul = $((n1 * n2))
div = $(echo "scale=2; $n1/$n2" | bc)
echo "Addition - $add"
echo "Subtraction - $sub"
echo "Multiplication - $mul"
echo "Division - $div"
```

2/11/21

Teacher's Signature :

Output

Enter a number : 10
The sum : 30

Expt. No. 9 Date 12-11-21

Page No.

Aim → Sum of even numbers upto a number

```
echo "Enter a number - "
read n
sum =0
for (( i=0 ; i<=$n ; i=i+2))
do
    sum = $(( sum +i ))
done
echo $ sum
```

9/10 12/1

Teacher's Signature :

Output

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

Date 16-11-21

Expt. No. 10 Page No.

Aim → Find all combinations of 1 2 3

```
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
```

16/11/21

Teacher's Signature :

Output

```
Enter base : 8
Enter power : 2
64
```

Date 16-11-21

Expt. No. 11 Page No.

Aim → Find power of a number

```
echo "Enter base: "
read b
echo "Enter power: "
read p
ans =1
while [ $p -ge 1 ]
do
    ans = $(echo "$ans * $b" | bc)
    p = $((p -1))
done
echo $ans
```

16/11/21

Teacher's Signature :

Output

```
Enter a number: 30
465
```

Date 16-11-21

Expt. No. 12 Page No.

Aim → 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
```

N
16/11/21

Teacher's Signature :

O/P

Enter subject: AA
Enter CIE marks: 30
Enter SEE marks: 60
C Grade
Enter subject: ADS
Enter CIE/see marks: 40/50
C Grade
Enter subject: USP
Enter CIE marks: 48
Enter SEE marks: 95
S Grade
Enter subject: AI
Enter CIE marks: 43
Enter SEE marks 90
A Grade
Enter subject: CN
Enter CIE marks: 20
Enter SEE marks: 30
Fail in CN
Enter subject: SPHF
Enter CIE marks: 50
Enter SEE marks: 50
B Grade

Total pass is 5
Total fail is 1

Date 29/11/21

Expt. No. 13 Page No.

Aim – Shell script to find pass class of a student.

```
pass = 0
for ((i=0; i<6; i++))
do
    echo "Enter Subject:"
    read sub
    echo "Enter CIE marks:"
    read cie
    echo "Enter SEE marks:"
    read see
    see = $(( see /2))
    tot = $(( cie + see ))
    case $tot in
    100) echo "S grade";;
    9[0-9]) echo "S grade";;
    8[0-9]) echo "A grade";;
    7[0-9]) echo "B grade";;
    6[0-9]) echo "C grade";;
    5[0-9]) echo "D grade";;
    4[0-9]) echo "E grade";;
    [0-3][0-9]) echo "Fail in $sub"
                pass = $(( pass - 1 ));;
    *) echo "Enter valid marks!";;
    esac
done
echo "Total pass is $pass"
fail = $(( 6 - pass ))
echo "Total fail is $fail"
```

Teacher's Signature :

O/p

Enter a numbe - 5

0

1

1

2

3

Date 29/11/21

Expt. No. 14 Page No.

Aim - Shell script to find fibonacci series upto n.

```
echo "Enter a number:"
read n
a=0
b=1
c=2
d=0
echo -e "$a $b \c"
while [ $c -lt $n ]
do
    c=`expr $c +1`
    d=`expr $a + $b`
    echo -e "$d \c"
    a=$b
    b=$d
done
```

Teacher's Signature :

O/P

Enter the string : Hello
Number of vowels is 2

Date 29/11/21

Expt. No. 15 Page No.

Aim - Shell script to find number of vowels in a string

```
echo "Enter the string"
read s
count=0
len=`expr "$s" : '.*'`
for (( i=1; i<=len; i++))
do
    c=`echo $s | cut -c $i`
    case $c in
        [aeiouAEIOU]) count=$((count+1))
    esac
done
echo "Number of vowels is $count"
```

Teacher's Signature :

O/P

Enter filename : program16.sh
Lines = 15
Words = 34
Characters = 145

Date 29/11/21

Expt. No. 16 Page No.

Aim → Shell script to find no. of lines, characters and words in a file.

```
echo "Enter filename: "
read f
lines = `wc -l < $f`
words = `wc -w < $f`
characters = `wc -m < $f`
echo "Lines = $lines \n Words = $words \n Characters = $characters"
```

Teacher's Signature :

Q/k

Output of this program is system dependent

Date 3/1/22

Expt. No. 17

Page No.

Aim → Write a program that outputs contents of environment list.

```
#include <stdio.h>
int main(int argc, char *argv[])
{
    int i;
    char **ptr;
    extern char **environment;
    for (ptr = environment; *ptr != 0; ptr++)
        printf("%s\n", ptr);
    return 0;
}
```

Teacher's Signature :

O/P

./a.out [-o] <org-file> <new-link>
./a.out 1 2 3 4
./a.out [-o] <org-file> <new-link>
./a.out 1.c 2
Hard link created
ls -l
./a.out -s 1a.c.2.2
Cannot create link
./a.out -s 1a.c 22
Symbolic link created
readlink 22
1a.c

Date 3/1/22

Expt. No. 18 Page No.

Aim - Write a program to emulate 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"))){
        print ("Usage: ./a.out [-s]<org_file><new_links>\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 :

6/6

System supports job control
System supports saved set-UID and saved set-GID
Chown-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.

Aim → Write a program to print POSIX defined configuration using test matches

```
#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
```

Teacher's Signature :

Date

Expt. No.

Page No.

```
#ifdef _POSIX_VDISABLE
  printf("Disable character for terminal file is %d\n", _POSIX_VDISABLE);
#else
   printf("System does not support _POSIX_VDISABLE\n");
#endif
return 0;
}
```

Teacher's Signature :

O/P

```
./a.out FIFO1 " This is USP lab " //Terminal 1
./a.out FIFO1 //Terminal 2
This is USP lab.
```

Date 3/1/22

Expt. No. 20

Page No.

Aim: Write program to demonstrate interprocess communication between a reader process and a writer process.

```
#include <sys/types.h>
#include <unistd.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){
    fd = open (argv[1], O_RDONLY|O_NONBLOCK);
    while (read(fd, buf, sizeof(buf))>0)
        printf("%s", buf);
  }
  else {
    fd=open(argv[1], O_RWRONLY);
    write (fd, argv[2], strlen(argv[2]));
  }
  close(fd);
}
```

Teacher's Signature :