

# **Particulars of the Experiments Performed CONTENTS**

Output :-

chmod 777 lab1.sh

-Enter year:

2010

2010 is not a leap year

Enter year:

2000

2000 is a leap year



Expt Lab 1 :- Shell script to find if the given year is leap year or not.

```
#!/bin/sh
```

```
echo "Enter year: "
```

```
read year
```

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

```
then echo "$year is a leap year."
```

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

```
then echo "$year is a leap year"
```

```
elif [ $(($year % 100)) -ne 0 ]
```

```
then echo "$year is not a leap year"
```

```
else
```

~~```
echo "$year is not a leap year"
```~~~~```
fi
```~~~~✓  
25/10/21~~

Output :-

chmod 777 lab2.sh

Enter radius of circle:

4

Area of circle is 50.24



Date 25/10/21

Expt. No. C2

Page No. 2

Lab 2: Shell script to find area of a circle.

`#!/bin/sh`

`echo "Enter radius of circle."`

`read r`

`pi=3.14`

`area=`echo $pi*$r*${r} |bc``

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

✓  
25/10/21

Output :-

chmod 777 lab3.sh

Enter a number:

0

0 is zero.

Enter a number:

1

1 is positive

Enter a number:

-6

-6 is negative

Lab 3 :- Shell script to check whether a number is positive, negative or zero.

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

✓ 25/10/21

Output :- chmod 777 lab4.sh

Enter 3 numbers:

4

-1

6

6 is the largest.

Enter 3 numbers:

10

5

-15

10 is the largest.

Lab 4 :- Shell script to find biggest of 3 numbers

#!/bin/sh

echo "Enter 3 numbers: "

read a

read b

read c

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

then echo "\$a is the largest."

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

then echo "\$b is the largest."

else

echo "\$c is the largest."

fi

✓  
25/10/21

Output ~ chmod 777 lab5.sh

Enter a number:

7

Factorial of 7 is 5040.

Lab 5:- Shell script to find factorial of a number

#!/bin/sh

echo "Enter a number : "

read num

factorial = 1

while [ \$num -gt 1 ]

do

factorial = \$((\$factorial \* num))

num = \$((\$num - 1))

echo "Factorial of \$num is \$factorial."

M/1/21  
8/11/21

Output:- chmod 777 lab6.sh

Enter basic salary of employee :

10000

The gross salary is 13000.

Lab 6:- Shell script to compute gross salary of an employee.

#!/bin/sh

echo "Enter basic salary of employee : "

read bsal

da=`echo 0.1 \* \$bsal | bc`

hra=`echo 0.2 \* \$bsal | bc`

grosssal=`echo \$bsal + \$da + \$hra | bc`

echo "The gross salary is \$grosssal."

N  
8/11/21

Output :- chmod 777 lab7.sh

Converting Fahrenheit to Celsius :

Enter temperature in Fahrenheit :

100

Temperature in Celsius is 37.77.

Lab 7:- Shell script to convert temperature Fahrenheit to Celsius.

```
#!/bin/sh
echo "Converting Fahrenheit to Celsius."
echo "Enter temperature in Fahrenheit :"
read fahr.
cel=`echo $(fahr-32)`
t=`echo "scale=2; 5/9;" | bc`
cel=`echo $(fahr-32) \* $t | bc`
echo "Temperature in Celsius is $cel."
```

✓  
M/21  
8/11/21

Output:- chmod 777 lab8.sh

-Enter 2 numbers:

10

3

Addition:  $10 + 3 = 13$

Subtraction:  $10 - 3 = 7$

Multiplication :  $10 * 3 = 30$

Division :  $10 / 3 = 3.3$

Modulus :  $10 \% 3 = 1$

Lab 8: - Shell script to perform arithmetic operation on 2 numbers.

```
#!/bin/sh
echo "Enter 2 numbers : "
read a
read b
val=`expr $a + $b 1bc`
echo "Addition : $a + $b = $val"
val=`expr $a - $b 1bc`
echo "Subtraction : $a - $b = $val"
val=`expr $a * $b 1bc`
echo "Multiplication : $a * $b = $val"
val=`expr $a / $b 1bc`
echo "Division : $a / $b = $val"
val=`expr $a % $b 1bc`
echo "Modulus : $a % $b = $val"
```

8/11/21

Output :- chmod 777 lab9.sh

Enter value of n :

10

Sum of even numbers upto 10 = 30.



Date 15/11/21

Expt. No. 9

Page No. 9

Lab 9 :- Shell script to find sum of even numbers upto n.

```
#!/bin/sh
echo "Enter value of n"
read n
i = 2
while [ $i -le $n ]
do
    sum = $(($sum+i))
    i=$((i+2))
done
echo "Sum of even numbers upto $n = $sum"
```

15/11/21

Output:- chmod 777 lab10.sh

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 15/11/21

Page No. 10

Lab 10 :- Shell script to print combinations of numbers 1, 2, 3.

```
#!/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
```

15/11/21

Output:- chmod 777 lab11.sh

- Enter the number and exponent:

2

3

2 to the power 3 is 8.



Date 15/11/21

Expt. No. 11

Page No. 11

Lab 11 :- Shell script to find power of a number

```
#!/bin/sh
echo "Enter the number and exponent : "
read a
read b
p=$b
res=1
while [ $b -gt 0 ]
do
    res=$(echo "$res * $a" | bc)
    b=$(echo "$b - 1" | bc)
done
echo "$a to the power $p is $res"
```

15/11/21

Output:- chmod 777 lab12.sh

Enter n:

10

Sum of first 10 natural numbers is : 55

Lab 12:- Shell script to find sum of n natural numbers.

```
#!/bin/sh
echo "Enter n: "
read n
sum=0
for (( i=1; i<=n; i++ ))
do
```

    sum=\$((sum+i))

done

echo "Sum of first \$n natural numbers : \$sum"

15/11/21

Output:- Enter cie and see(50) marks for subject 1 :

45

44

A grade

Enter cie and see(50) marks for subject 2:

46

40

A grade

Enter cie and see(50) marks for subject 3:

20

10

Fail

Enter cie and see(50) marks for subject 4:

15

20

Fail

Enter cie and see(50) marks for subject 5:

30

22

D grade

Enter cie and see(50) marks for subject 6:

18

20

Fail

Number of subjects passed : 3

Number of subjects failed : 3

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

```

#!/bin/bash
pass=0
fail=0
for((i=1; i<=6; i++))
do
echo "Enter cie and sce(50) marks for subject $i : "
read cie
read sce
total=$((cie+sce))

if [ $total -gt 90 ]
then echo "S grade"
pass=$((pass+1))
elif [ $total -gt 80 ]
then echo "A grade"
pass=$((pass+1))
elif [ $total -gt 70 ]
then echo "B grade"
pass=$((pass+1))
elif [ $total -gt 60 ]
then echo "C grade"
pass=$((pass+1))
elif [ $total -gt 50 ]
then echo "D grade"
pass=$((pass+1))
fi
done

```

Teacher's Signature :

```
elif [ $total -gt 10 ]
```

```
then echo "E grade"
```

```
pass=$((pass+1))
```

```
else
```

```
echo "Fail"
```

```
fail=$((fail+1))
```

```
fi
```

```
done
```

```
echo "Number of subjects passed : $pass"
```

```
echo "Number of subjects failed : $fail"
```

Output - chmod 777 lab14.sh

Enter n:

3

0

1

1

Enter n:

5

0

1

1

2

3

Lab 14:- Shell script to find Fibonacci series upto n.

```
#!/bin/sh
echo "Enter n : "
read n
f1=0
f2=0
for (( i=0; i<n; i++ ))
do
echo "f1"
f3=$((f1+f2))
f1=$f2
f2=$f3
done
```

Output:- chmod 777 lab15.sh

Enter string

good

The given string has 2 vowels.

Enter string

zzz

The given string has 0 vowels.

Lab 15:- Shell script to count number of vowels of a string

```
#!/bin/sh
```

```
echo "Enter string: "
read str
```

```
vowcount=0
```

```
v=`expr "$str": '.*'`
```

```
for (( i=0; i<$v; i++ ))
do
```

```
s=`expr "$str": '\([.\) ]'`
```

```
if [ "$s" = 'a' -o "$s" = 'e' -o "$s" = 'i' -o "$s" = 'o'
-o "$s" = 'u' ]
```

then

```
vowcount=$((vowcount+1))
```

fi

```
str=`expr "$str": '\([.\)*\)'`
```

done

```
echo "The number of vowels: $vowcount."
```

Output :- chmod 777 lab16.sh

Enter filename  
usp.txt

Number of characters in usp.txt : 37

Number of words in usp.txt : 8

Number of lines in usp.txt : 3

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

```
#!/bin/bash
echo "Enter filename"
read file
c=`cat $file | wc -c`
w=`cat $file | wc -w`
l=`grep -c "\." $file`
echo "Number of characters in $file : $c"
echo "Number of words in $file : $w"
echo "Number of lines in $file : $l"
```

Lab 17:- 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;
}
```

Lab 18:- Write C/C++ program to emulate unix ls command

#

#include &lt;stdio.h&gt;

#include &lt;sys/types.h&gt;

#include &lt;unistd.h&gt;

#include &lt;string.h&gt;

int main(int argc, char \*argv[])

{

if(argc &lt; 3 || argc &gt; 4 || (argc == 4 &amp;&amp; strcmp(argv[1], "-S")))

printf("USAGE: ./a.out [-S] &lt;arg\_file&gt; &lt;new\_link&gt;\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("Hard link created\n");

}

return 0;

}

Lab 19:- Write 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()
{
#ifndef _POSIX_JOB_CONTROL
    printf("System supports job control\n");
#else
    printf("System does not support job control\n");
#endif
#ifndef _POSIX_SAVED_IDS
    printf("System supports saved set-UID and saved set-GID\n");
#endif
#ifndef _POSIX_CHOWN_RESTRICTED
    printf("chown restricted option is %d\n", _POSIX_CHOWN_RESTRICTED);
#else
    printf("System doesn't support chown restricted option\n");
#endif
#ifndef _POSIX_NO_TRUNC
    printf("Pathname trunc option is option is %d\n", _POSIX_NO_TRUNC);
#else
    printf("System doesn't support system-wide pathname trunc
option.\n");

```

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

```
#endif  
#ifdef _POSIX_VDISABLE  
printf ("Disable character for terminal files is %d\n",  
    _POSIX_VDISABLE);  
#else  
printf ("System doesn't support _POSIX_VDISABLE\n");  
#endif  
return 0;  
}
```

Lab 20:- 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 <font.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);
    }
}
```

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

Date \_\_\_\_\_

Page No. 23

else  
{

    fd = open (argv[1], O\_WRONLY);  
    write (fd, argv[2], strlen(argv[2]));  
}

    close (fd);  
}