Particulars of the Experiments Performed

CONTENTS

Particulars of the Experiments Performed

CONTENTS

Input 1:
Enter the year:
2000
Output 1:
It is a leap year.

Input 2:
Enter the year:
~~Output~~
1800
Output 2:
It is not a leap year.

Input 3:
Enter the year:
1996
Output 3:
It is a leap year.

Date 25/10/21

Expt. No. 1

Page No. 1

PROGRAM-1

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

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

Teacher's Signature :

Input :

Enter the radius :

10

Output :

Area of the circle :

314.1500

PROGRAM-2

Shell script to find the area of a circle.

```
#!/bin/bash
echo "Enter the radius: "
read radius
echo "Area of the circle: "
pi=3.1415
ans=`echo $pi\*$radius\*$radius|bc`
echo $ans
```

Input 1:
Enter the number:
6
Output 1:
Positive

Input 2:
Enter the number:
-4
Output 2:
Negative

Input 3:
Enter the number:
0
Output 3:
Zero

Date 25/10/21

Expt. No. 3 Page No. 3

PROGRAM-3

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

```
#!/bin/sh
echo "Enter the number: "
read no
if [ $no -eq 0 ]
then
echo "Zero"
elif [ $no -gt 0 ]
then
echo "Positive"
else
echo "Negative"
fi
```

Teacher's Signature : _____

Input :

```
Enter the first number:
23
Enter the second number:
7
Enter the third number:
0
```

Output :

```
23 is the greatest.
```

Date 25/10/21

Expt. No. 4

Page No. 4

PROGRAM-4

Shell script to find the biggest of three numbers.

```
#!/bin/bash
echo "Enter the first number: "
read n1
echo "Enter the second number: "
read n2
echo "Enter the third number: "
read n3
if [ $n1 -gt $n2 ] && [ $n1 -gt $n3 ]
then
echo $n1 " is the greatest."
elif [ $n2 -gt $n1 ] && [ $n2 -gt $n3 ]
then
echo $n2 " is the greatest."
else
echo $n3 " is the greatest."
fi
```

Teacher's Signature : _____

Input:
Enter a number:
3
Output:
The factorial is 6

Date 08/11/21

Expt. No. 5

Page No. 5

PROGRAM-5

Shell script to find the factorial of a number.

```
#!/bin/sh
echo "Enter a number:"
read num
fact=1
while [ $num -gt 1 ]
do
fact=$((fact * num))
num=$((num-1))
done
echo "The factorial is $fact"
```

Teacher's Signature :

Input:
Enter the basic salary:
1000
Output:
Gross salary = 1300

PROGRAM-6

Shell script to compute the gross salary of an employee.

```
#!/bin/sh
echo "Enter the basic salary: "
read basic
da=`expr $basic \* 10/100`
hra=`expr $basic \* 20/100`
gross=$(( $basic + $da + $hra ))
echo "Gross salary = $gross"
```

Teacher's Signature : _____

Input:

Enter the temperature in fahrenheit:

100

Output:

100 F is equal to 37.7777 C

Date 08/11/21

Expt. No. 7 Page No. 7

PROGRAM-7

Shell script to convert the temperature Fahrenheit to Celcius.

```
#!/bin/sh
echo "Enter the temperature in fahrenheit: "
read fahrenheit
celsius=`expr` echo "scale=4; ($fahrenheit-32)/1.8" | bc
echo "$fahrenheit F is equal to $celsius C"
```

Teacher's Signature : _____

Input:
Enter the two numbers:
10
20
Output:
Addition: 30
Subtraction: -10
Multiplication: 200
Division: .50

Date 08/11/21

Expt. No. 8 Page No. 8

PROGRAM-8

Shell script to perform arithmetic operations on given two numbers.

```
#!/bin/sh
echo "Enter the two numbers: "
read n1
read n2
echo "Addition: $(( $n1 + $n2 ))"
echo "Subtraction: $(( $n1 - $n2 ))"
echo "Multiplication: $(( $n1 * $n2 ))"
d=`echo "scale=2; $n1/$n2" | bc`
echo "Division: $d"
```

Teacher's Signature :

Input:

Enter the number:

10

Output:

Sum of the 10 even numbers = 20

Date 15/11/21

Expt. No. 9

Page No. 9

PROGRAM-9

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

```
#!/bin/bash
echo "Enter the number:"
read n
i=2
while [ $i -lt $n ]
do
sum=$((sum+i))
i=$((i+2))
done
echo "Sum of the $n even numbers = $sum"
```

Teacher's Signature : _____

Output:

Вихід

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

Date 15/11/21

Expt. No. 10 Page No. 10

PROGRAM-10

Shell script to print the combinations of numbers 1 2 3.

```
#!/bin/bash
for (( i=1; i<=3; i++ ))
do
for (( j=1; j<=3; j++ ))
do
for (( k=1; k<=3; k++ ))
do
echo $i $j $k
done
done
done
```

Teacher's Signature : _____

Input:
Enter the number:
2
Enter the power:
10
Output:
Result = 1024

Date 15/11/21

Expt. No. 11

Page No. 11

PROGRAM-11

Shell script to find the power of a number.

```
#!/bin/bash
echo "Enter the number:"
read no
echo "Enter the power:"
read power
result=1
count=0
while [ $power -ge 1 ]
do
result=`expr $result\* $no`
power=`expr $power-1`
count=`expr $count+1`
done
echo "Result=$result"
```

Teacher's Signature : _____

Input:
Enter the upper limit:
6
Output:
15

Date 15/11/21

Expt. No. 12 Page No. 12

PROGRAM-12

Shell script to find the sum of n natural numbers

```
#!/bin/bash
echo "Enter the upper limit: "
read max
sum=0
for (( i=0; i<max; i++ ))
do
sum=$(expr $sum + $i)
done
echo $sum
```

Teacher's Signature :

Output:

```
Enter the marks obtained in subject1
70
Enter the marks obtained in subject2
80
Enter the marks obtained in subject3
90
 Grade B
```

Date 29/11/21

Expt. No. 13 Page No. 13

PROGRAM-13

Shell script to display pass class of a student.

```
#!/bin/sh
echo "Enter the marks obtained in subject1"
read m1
echo "Enter the marks obtained in subject2"
read m2
echo "Enter the marks obtained 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"
else
echo "Fail"
fi
```

Teacher's Signature : _____

Input:

Enter a positive value:

5

Output:

Fibonacci Series

0

1

1

2

3

Date 29/11/21

Expt. No. 14 Page No. 14

PROGRAM - 14

Shell script to find the Fibonacci series upto n

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

Teacher's Signature : _____

Input:

Enter the string

hello

Output:

Length of the string = 5

No. of vowels = 2

PROGRAM - 15

Shell script to count the number of vowels in a string.

```
#!/bin/sh
echo "Enter the string"
read str
len=$(expr len $str)
echo "Length of the string = " $len
vowel=0
while [ $len -gt 0 ]
do
ch=$(echo $str | cut -c $len)
case $ch in
   ([aeiouAEIOU]) vowel=$((vowel+1));;
esac
len=$((len-1))
done
echo "No. of vowels = $vowels"
```

Teacher's Signature : _____

Input:

No. of lines, characters and words in a file

Enter the filename:

hellosh

Output:

Lines = 17

Characters = 256

Words = 32

Date 29/11/21

Expt. No. 16 Page No. 16

PROGRAM-16

Shell script to check no. of lines, words, characters in a file.

```
#!/bin/sh
echo "Enter the no. of lines, characters and words in
        a file"
echo "Enter the filename:"
read fname
lines = `wc -l < $fname`
characters = `wc -c < $fname`
words = `wc -w < $fname`
echo "lines = $lines"
echo "Characters = $characters"
echo "Words =$words"
```

Teacher's Signature : _____

Output:

Output for this program is system dependent.

Date 03/01/22

Expt. No. 17 Page No. 17

PROGRAM-17

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;
}
```

Teacher's Signature : _____

Output:

```
./a.out [-s] <org-file> <new_link>
./a.out 1 2 3 4
./a.out [-s] <org-file> <new_link>
./a.out 1.c z
Hard link created
ls -l
./a.out 1a.c.z
Cannot create hard link
./a.out -s 1a.czz
Symbolic link created
readlink zz
1a.c
```

Date 03/01/22

Expt. No. 18 Page No. 18

PROGRAM - 18

Write a program to emulate Unix ln command in C/C++.

```
#include<stdio.h>
#include <sys/types.h>
#include <unistd.h>
#include <string.h>
int main(int argc, char *argv[]){
  if (argc<3|| argv>4 || (argc==4 && strcmp(argv[i],
        "s"))) {
     printf("Usage: ./a.out [-s] <arg_file> <new_file>
                \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)
        }
```

Teacher's Signature :

```
        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 set-GID
chown_restricted option is 1
Pathname trunc option is 1
Disable character for terminal file is 0

PROGRAM-19

Write a program in C/C++ to print POSIX defined configuration options.

```
#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 & set-GID\n");
  #else
      printf("System does not support saved set of
                                    UID & GID\n");
  #endif

  #ifdef _POSIX_CHOWN_RESTRICTED
      printf("chown-restricted option is %d\n", _POSIX_
                                    CHOWN_RESTRICTED);
```

Teacher's Signature : _____

```
#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

#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 : _____

Output:

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

PROGRAM-20

Write a C/C++ program to demonstrate interprocess communication between a reader process and a writer process.

```
#include <sys/types.h>
#include <unistd.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <strlengh>
#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_RPONLY|O_NONBLOCK);
      while(read(fd, buff, sizeof(buff))>0)
```

Teacher's Signature : _____

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

Teacher's Signature : _____