B M S. COLLEGE OF ENGINEERING

(Autonomous Institution)

RECORD OF PRACTICAL WORK

NAME :...SARASWATHI.B.....

SUBJECT :...DSP.....

SEMESTER :...V.....BRANCH :..CSE.....

ROLL NO :.....USN :1BM19CS032

Particulars of the Experiments Performed

CONTENTS

Particulars of the Experiments Performed

CONTENTS

Expt. No. 1

Date 25/10/21

Page No. 1

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

```
#!/bin/sh
echo "Enter year"
read year
if [ $((year%4)) -eq 0 ]
then
if [ $((year%100)) -eq 0 ]
then
if [ $((year%400)) -eq 0 ]
then
echo "Leap year"
else
echo "Not a leap year"
Fi
else
echo "Leap year"
Fi
else
echo "Not a leap year"
fi
```

Teacher's Signature :

Output:

Enter year
2016
Leap year

Enter year
2013
Not a leap year

Date 25/10/24

Expt. No. 2

Page No. 2

* Shell script to find area of circle

```
#!/bin/sh
echo "Enter radius"
read r
pi=3.1415
ans=`echo "$pi*$r*$r" | bc`
echo $ans
```

Teacher's Signature :

Output

Enter radius

4

50.2624

Expt. No. 3

Date 25/10/21

Page No. 3

* Shell script to find whether the number is zero or positive or negative

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

Teacher's Signature :

Output

Enter number
0
zero

Enter number
4
positive numbe

Ente number
-8
Negative numbe

Expt. No. 4 Date 25/10/21

Page No. 4

* shell script to find biggest of three numbers

```
#!/bin/sh
echo "Enter a"
read a
echo "Enter b"
read b
echo "Enter c"
read c
if [ $a -gt $b -a $a -gt $c ]
then
echo "a is big"
elif [ $b -gt $c ]
then
echo "b is big"
else
echo "c is big"
Fi
```

Teacher's Signature :

Output:

```
Enter a
-3
Enter b
 8
Enter c
 1
b is big
```

Expt. No. 5

Date 8/11/21

Page No. 5

* Shell script to find factorial of a number

```
#!/bin/bash
Echo "Enter number"
read n
fact=1
for (( i=1 ; i<=$n ; i++ ))
do
fact=$((fact*i))
done
echo "Factorial is : $fact"
```

output

Enter number
4
Factorial is : 24

Expt. No. 6

Date 1/11/21

Page No. 6

1. Shell script to compute gross salary of an employee

```
#!/bin/sh
echo "Enter basic salary:"
read basic
da=`echo "scale=4; $basic * 0.1" | bc`
hra=`echo "scale=4; $basic * 0.2" | bc`
gross=`echo "$basic + $da + $hra" | bc`
echo "Gross salary : $gross"
```

Teacher's Signature :

Output:

Enter basic salary:
900.98
Gross salary : 1171.274

Expt. No. 7

Date

Page No. 7

* Shell script to convert the temperature Fahrenheit to celsius

```
#!/bin/sh
echo "Enter temperature:"
read fah
a=`echo "scale=4; 5/9" | bc`
b=`echo "scale=4;8((fah-32))*8a" | bc`
echo "Temperature in celsius: 8b"
```

Teacher's Signature :

output

Enter temperature

32

Temperature in celsius : 0

Expt. No. 8 Date 8/11/21

Page No. 8

* Shell script to perform Arithmetic operations on two numbers

```
#!/bin/sh
echo "Enter two numbers:"
read n1
read n2
echo "Addition: $((n1+n2))"
echo "Subtraction: $((n1-n2))"
echo "Multiplication: $((n1*n2))"
d=`echo "scale=4; $n1/$n2" | bc`
echo "Division: $d"
```

Teacher's Signature :

Output

```
Enter two numbers:
20
5
Addition : 25
Subtraction : 15
Multiplication : 100
Division : 4.0000
```

Expt. No. 9 Date 15/11/24

Page No. 9

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

```
#!/bin/bash
echo "Enter the sum limit"
read n
sum=0
for (( i=0 ; i<=$n; i=i+2 ))
do
sum=$((sum+i))
done
echo "Even sum: $sum"
```

Teacher's Signature :

output

Enter limit the sum limit
10
Even sum : 30

Expt. No. 10

Date 15/1/21

Page No. 10

* Shell script to print the combination of numbers 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:

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. 11 Date 15/11/24

Page No. 11

Shell script to find the power of a number

```
#!/bin/bash
echo "Enter base: "
read b
echo "Enter power: "
read p
pow=1
for (( i=1 ; i<=$p ; i++ ))
do
pow=$(($pow * $b | bc ))
done
echo $pow
```

Teacher's Signature :

Output

```
Enter base:
2
Enter power:
5
32
```

Date 12/11/24

Expt. No. 12

Page No. 12

* Shell script to find sum of n natural numbers

```
#!/bin/bash
echo "Enter sum limit: "
read n
sum=0
for ((i=0 ; i<=$n ; i=i+1))
do
sum=$((sum + i))
done
echo "Sum: $sum"
```

Output

Enter sum limit:

4

Sum : 10

Expt. No. 13 Date 29/11/21

Page No. 13

* Shell script to display the 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
ebho "Enter the marks obtained in subject3"
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 $avg -le 90]
then
    echo "Grade B"
elif [$avg -gt 40 -a $avg -le 65]
then
    echo "Grade C"
else
    echo "Fail"
fi
```

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
```

Expt. No. 14 Date 24/11/21

Page No. 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 :

Output:

```
Enter positive values
5
Fibonacci series
0
1
1
2
3
```

Date 29/11/21

Expt. No. 15

Page No. 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 :

output:

```
Enter the string
great
length of string = 5
No of vowels = 2
```

Date 29/11/21

Expt. No. 16

Page No. 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"
read 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:

```
Enter filename
vowel.sh
Lines = 15
Characters = 301
Words = 49
```

Expt. No. 17 Date 31/12

Page No. 17

* Write a c/c++ program to that output 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:

Output for this program is system dependent

Date 8/11/22

Expt. No. 15

Page No. 15

* Write a 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(" cannot create hard link");
else
printf ("Hard link created \n");
}
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.c zz

Symbolic link created

readlink zz

1a.c

Date 3/11/22

Expt. No. 19

Page No. 19

* Write a C/C++ POSIX complaint program that prints the POSIX defined configuration option 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");
#endif
#ifdef
#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 doesnot support chown_restricted option\n");
#endif
```

Teacher's Signature :

Date 3/1/22

Expt. No. _____ Page No. 20

```
#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 files is %d \n", _POSIX_VDISABLE);
#else
printf("System does not support _POSIX_VDISABLE \n");
#endif
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 trunc option is 1
Disable character for terminal files is 0.

Date :

Expt. No. 20 Page No. 21

Write a c program/c++ program which demonstrate inter process communication between a reader process and writer process. Use mkfifo, open, read, write and close APIs in your program.

```
#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 <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);
}
```

Teacher's Signature :

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

output:

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