



# B M S. COLLEGE OF ENGINEERING

(Autonomous Institution)

## RECORD OF PRACTICAL WORK

NAME :.....Aishwarya M.....  
SUBJECT :.....VSR-Lab.....  
SEMESTER :.....5<sup>th</sup>.....BRANCH :.....CS.....  
ROLL NO :.....USN : JBM20CSH01

# Particulars of the Experiments Performed

## CONTENTS

Expt No.	Date	Experiment	Marks Obtained	Page No.
1	25/10/21	Shell script to find leap year		01
2	25/10/21	Shell script to find area of circle		02
3	25/10/21	Shell script to find the biggest of 3 number		03
4	25/10/21	Shell script to check whether the number is zero / positive / negative		04
5	8/11/21	Shell script to find the factorial of a number		05
6.	8/11/21	Shell script to compute the gross salary of an employee		06
7.	8/11/21	Shell script to compute the temperature Fahrenheit to Celsius		07
8	8/11/21	Shell script to perform arithmetic operations on given 2 numbers		08
9.	15/11/21	Shell script to find the sum of even numbers upto n.		09
10	15/11/21	Shell script to print the combination of numbers 1 & 3		10
11	15/11/21	Shell script to find the power of a number		11
12.	15/11/21	Shell script to find the sum of n natural numbers.		12

# Particulars of the Experiments Performed CONTENTS

Expt No.	Date	Experiment	Marks Obtained	Page No.
13		shell script to display the pass class of a student		13
14		shell script to find the fibonacci series upto n		15
15		shell script to count the number of vowels of a string		16
16.		shell script to check no: of lines, words & character		17
17.		write a c/c++ prog to that o/p's the contents of its environment - list		18
18.		write a c/c++ prog to enumerate the unix command		19
19.		write a c/c++ prog POCIX command that prints the POCIX defined		21
20.		write c/c++ prog which demonstrate Interprocess communication b/w a reader process & writer		23
		use mkfifo, open, read write and close api's in your program		21

### Program: 01

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

```
echo "Enter the year"
```

```
read leap
```

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

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

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

```
then echo "The year entered is a leap year"
```

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

```
then echo "The year entered is not a leap year"
```

```
else
```

```
echo "The year entered is not a leap year"
```

```
fi
```

## Outputs

Enter the year

2024

This year is a leap year.

Enter the year

2001

This year is not a leap year.

Program : 2.

Shell script to find the area of a circle.

echo "Enter the radius of the circle"

read rad

pi = 3.14

area = 'echo \$pi\*\$rad^\*\$rad | bc'

echo "Area of circle is \$area"

Output:

Enter the radius of the circle :  $\frac{2}{2}$

area of the circle is 12.56.

Program : 03

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

echo "Enter a number"

read num

if [ \$num -eq 0 ]

then echo "zero"

elif [ \$num -gt 0 ]

then echo "positive"

else

echo "negative"

fi.

Output:

Enter a number

-50

Negative

Program : 04

Shell script to find the biggest of three numbers.

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 -a \$n1 -gt \$n3 ]

then echo "\$n1 is the greatest"

elif [ \$n2 -gt \$n1 -a \$n2 -gt \$n3 ]

then echo "\$n2 is the greatest"

else

echo "\$n3 is the greatest"

fi.

Output:

Enter the first number

2

Enter the second number

3

Enter the third number

5

5 is the biggest number.

Program : 05

shell script to compute factorial of a number

echo "Enter a number"

read num

fact = 1

for ((i=2 ; i<=num ; i++))

do

fact = \$((fact \* i))

done

echo \$fact

Output:

Enter a number

3

→ 6

Program : 06

Shell script to compute the gross salary of an employee

echo "Enter the basic salary of the employee"  
read basic

da = 'echo 0.1 \* \$basic | bc'

hra = 'echo 0.2 \* \$basic | bc'

gross = 'echo \$basic + \$da + \$hra | bc'

echo "Gross salary : \$gross"

## Output

Enter the basic salary of the employee

1000

Gross salary : 1300.0

Program : 07.

shell script to convert the temperature Fahrenheit  
to Celsius.

echo "Enter the temperature in Fahrenheit"  
read temp

t='echo "scale=4; 5/9;" | bc'

cel='echo \$((temp - 32)) | \* \$t | bc'

echo "Temperature in celsius : \$cel"

Output

Enter the temperature in fahrenheit

32

Temperature is celsius : 0

Program: 08

shell script to perform arithmetic operations  
on given two numbers.

echo "Enter 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"

Output

Enter two numbers.

6

3

Addition : 9

Subtraction: 3

Multiplication: 18

Division: 2.00

Program: 09

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

```
#!/bin/bash
echo enter limit
read n
i=2
while [ $i -lt $n ]
do
    sum=$((sum+i))
    i=$((i+2))
done
echo $sum.
```

Output :

Enter the number

7

12

Program : 10

shell script to print the combinations of numbers 1 & 3.

for i in 1 2 3

do

for j in 1 2 3

do

for k in 1 2 3

do

if [ \$i -ne \$j -ne \$k -a \$i -ne \$k ]  
-then echo "\$i \$j \$k"

fi

echo \$i \$j \$k

done

done

done

Output:

1 3 3

2 1 1

2 1 2

2 1 3

2 1 3

2 2 1

2 2 2

2 2 3

2 3 1

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

## Program: 11

Shell script to find the power of a number.

```
#!/bin/sh
echo "enter a number and power"
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 "result is $res"
```

Output

Enter a number and power.

2

6

result is 64

Program : 12.

Shell script to find the sum of n natural numbers

```
echo "Enter size(N)"  
read N  
i=1  
sum=0  
echo "Enter Numbers"  
while [ $i -le $N ]  
do  
    read num  
    sum=$((sum+num))  
    i=$((i+1))  
done  
echo $sum.
```

## Output

Enter size (N)

5

Enter numbers

3

8

55

2

1

69

## Program : 13

shell script to display the pass class of a student

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

```
echo "Enter the no of subject"
```

```
read n
```

```
cie=0
```

```
sel=0
```

```
i=1
```

```
fail=0
```

```
while [ $i -le $n ]
```

```
do
```

```
echo "Enter the marks of subject $i"
```

```
echo "cie"
```

```
read cmarks
```

```
cce=$((cie + cmarks))
```

```
echo "sel"
```

```
read smarks
```

```
sel=$((sel + smarks))
```

```
if [ ${cmarks} -lt 40 -o ${smarks} -lt 40 ]
```

```
then
```

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

```
echo "Subject $i : Fail"
```

```
else
```

```
echo "Subject $i : Pass"
```

```
fi
```

$i = \$((i+1))$

done

total = \$(((cii + see)/2))

final = \$((total/n))

echo \$finalneeb.telegram

if [ \$final -ge 90 ]

then

echo "GRADE S"

elif [ \$final -ge 80 ]

then

echo "GRADE A"

elif [ \$final -ge 70 ]

then

echo "GRADE B"

elif [ \$final -ge 60 ]

then

echo "GRADE C"

elif [ \$final -ge 50 ]

then

echo "GRADE D"

elif [ \$final -ge 40 ]

then

echo "GRADE E"

else

echo "GRADE F"

fi

echo "Total Subject Failed : \$fail"

Output :

Enter your civ marks out of 50

23

Enter your sci marks out of 50

23

E grade

Enter your civ marks out of 50

12

Enter your sci marks out of 50

12

Fail

Enter your civ marks out of 50

45

Enter your sci marks out of 50

35

B Grade .

Program: 14.

A shell script to find the fibonacci series up to n.

echo "Enter a number"

read n

x=0

y=1

i=2

echo "Fibonacci series \$n : "

echo \$n

else

echo \$x

echo \$y

while [ \$i -lt \$n ]

do

i=\$((i+1))

z=\$((x+y))

echo \$z

x=\$y

y=\$z

done.

Output:

Enter the value of n

3

0

1

1

Enter the value of n

6

0

1

1

2

3

5

Program : 15

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

```
#!/bin/bash
echo "string"
read string
count = 0
for ((i=0; i<${#string}; i++)); do
    char=${string:$i:1}
    if [ $char = 'a' -o $char = 'e' -o $char = 'i' -o
        $char = 'o' -o $char = 'u' ]
    then
        count=$(expr $count + 1)
    fi
done
echo "number of vowels :" $count.
```

Output:

Enter a string  
adfgijkl

1

Enter a string  
aoine

5

Program: 16.

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

```
#!/bin/sh
echo "enter the file name"
read fname
echo "number of characters"
wc -c < $fname
echo "number of words"
wc -w < $fname
echo "number of lines"
wc -l < $fname
```

Addition of two numbers - 2022

Subtraction of two numbers

Multiplication of two numbers

Division of two numbers

Modulo operation of two numbers

Comparison of two numbers

Concatenation of two strings

Length of a string

Reverse of a string

Capitalization of a string

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

Output :

program.sh  
number of characters

312

number of words

55

number of character lines

15 .

Program: 17

Write a C/C++ program to the output the contents of its environment list.

```
#include <f.h>; stdio; h.fgt;
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:

SSH\_AGENT\_PID = 3209  
HOSTNAME = localhost.localdomain  
DESKTOP\_STARTUP\_ID =  
SHELL = /bin/bash  
TERMS = xterm  
HISTSIZE = 1000  
KDE\_NO\_IPV6 = 1  
GTK\_RC\_FILES = /etc/gtk/gtkrc: /root/.gtkrc-1.2-gnome2  
WINDOWID = /root/tan  
QTDIR = /usr/lib/qt-3.3  
QTINC = /usr/lib/qt-3.3/include  
USER = root  
INPUTRC = /etc/inputrc  
PWD = /root/tan/usr  
XMODIFIERS = @im = none  
KDE\_IS\_PRELINKED = 1  
LANG = en-US.UTF-8  
GDMSESSION = default  
SSH\_ASKPASS = /usr/libexec/openssh/gnome-ssh-askpass  
HOME = /root  
SHLVL = 2  
GNOME\_DESKTOP\_SESSION\_ID = DEFAULT  
DISPLAY = 0.0  
G-BROKEN-FILENAMES = 1  
COLORTERMS = gnome-terminal  
XAUTHORITY = /tmp/gdm5X71UN  
= /root.

Program: 18

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

```
#include <fcntl.h>; #include <stdio.h>;  
#include <fcntl.h>; #include <sys/types.h>;  
#include <fcntl.h>; #include <unistd.h>;  
#include <fcntl.h>; #include <string.h>;  
int main(int argc, char *argv[]){  
    if(argc != 3 || argc > 4 || (argc == 4 &&  
        argv[3] != NULL && strcmp(argv[1], "-s") != 0))  
    {  
        printf("Usage: ./a.out [-s] file1;file2;new-link\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(  
        if(link(argv[1], argv[2]) == -1)  
            printf("Hard link created\n");  
    }
```

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

Date \_\_\_\_\_

Page No. 20

3

medium O;

3

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

## Output .

Usage : ./a.out [-s] <org-file> <new-link>  
[root@localhost usp]# ./a.out 1 2 3 4.

Usage : ./a.out [-s] <org-file> <new-link>  
[root@localhost usp]# ./a.out 1.c z.  
Hard link created.

[root@localhost usp]# ./a.out -s 1a.c  
Symbolic link created.

Programs: 19

Write c/c++ .POSIX compliant program  
that prints the POSIX defined configuration options  
supported on given system using feature macros.

```
#define _POSIX_SOURCE
#define _POSIX_C_SOURCE 199309L
#include <sys/types.h>;
#include <sys/conf.h>;
#include <sys/conf.h>;
#include <sys/conf.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 -dn,
           _POSIX_CHOKIN_RESTRICTED");
    #endif
}
```

```

#else
printf("f quiet; system does not support chosen
       restricted option in f quiet;");
#endif
#ifndef _POSIX_NO_TRUNC
printf("f quiet; Pathname trunc option is 1'd1 in
       f quiet; , _POSIX_NO_TRUNC);
#else
printf("f quiet; system does not support system
       -wide pathname trunc option in f quiet);
```

#endif

#ifndef \_POSIX\_VDISABLE

printf("f quiet; Disable character for terminal
 files is 1'd1 in f quiet), \_POSIX\_VDISABLE );

#else

printf("f quiet; system does not support \_POSIX-
 VDISABLE in f quiet);

#endif

return 0;

3

## 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

Program: 20

Write a C/C++ program which demonstrate 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 <sys/stat.h>
#include <unistd.h>
#include <fcntl.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_IROWXO);
    if( argc == 2 )
    {
        fd = open(argv[1], O_RDONLY | O_NONBLOCK);
        while( read(fd, buf, sizeof(buf)) > 0 )
```

} `printf ("i.s", buf);`

`else`

{

`fd = open(argv[1], O_WRONLY);`  
`writen(fd, argv[2], strlen(argv[2]));`

}

`close(fd);`

}

## Output :

1# Terminal 1 - writer process \* /

[root@localhost uspl] # . /a.out FIFO1 "this is  
USP Lab."

1# Terminal 2 - reader process \* /

[root@localhost 13] # . /a.out FIFO1  
this is UP Lab.