



B M S. COLLEGE OF ENGINEERING
(Autonomous Institution)

RECORD OF PRACTICAL WORK

NAME : ADITHI GIRIMAJI
SUBJECT : U.S.P.
SEMESTER : V BRANCH : CSE
ROLL NO : USN : 1B M19C8005

Particulars of the Experiments Performed CONTENTS

Expt No.	Date	Experiment	Marks Obtained	Page No
1	25/10/21	Program 1 - leap year		1
2	25/10/21	Program 2 - area of circle		2
3	25/10/21	Program 3 - +ve/-ve/negative		3
4	25/10/21	Program 4 - largest of 3 numbers	Not done 8/11/21	4
5	8/11/21	Program 5 - factorial of a number		5
6	8/11/21	Program 6 - gross salary of an emp		6
7	8/11/21	Program 7 - Fahrenheit to Celsius		7
8	8/11/21	Program 8 - arithmetic operations		8
9	15/11/21	Program 9 - sum of even upto n		9
10	15/11/21	Program 10 - combination of 123	Not done 15/11/21	10
11	15/11/21	Program 11 - power of a number		11
12	15/11/21	Program 12 - sum of n natural nos.		12
13	29/11/21	Program 13 - pass marks of a student		13-14
14	29/11/21	Program 14 - fibonacci series		15
15	29/11/21	Program 15 - number of vowels		16
16	29/11/21	Program 16 - lines, words, characters in a file		17
17	3/1/22	Program 17 - environment list		18
18	3/1/22	Program 18 - to emulate ln command		19
19	3/1/22	Program 19 - POSIX defined configuration options		20-21
20	3/1/22	Program 20 - fifo		22

Expt. No. 01

```
chmod 777 one.sh  
./one.sh 2000  
Leap year  
./one.sh 2001  
Not a leap year
```

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

```
#!/bin/sh
if [ $(($1 \% 400)) -eq 0 ]
then
echo "Leap year"
elif [ $(($1 \% 4)) -eq 0 ]
then
echo "Leap year"
elif [ $(($1 \% 100)) -ne 0 ]
then
echo "Not a leap year"
else
echo "Not a leap year"
fi
```

2/10/21
2/10/21

Teacher's Signature :

Date 25/10/21

Page No. 2

Expt. No. 02

chmod 777 two.sh

./two.sh

Enter the radius

2

12.56 is the area of the circle



Shell script to find the area of a circle

#!/bin/sh

echo "Enter the radius"

read r

pi=3.14

a='echo \$pi*\$r*\$r|bc'

else "The area of the circle"

✓
25.10.21
2021

Teacher's Signature :

chmod 777 three.sh
./three.sh 0
Argument is equal to zero
./three.sh 1
Argument is positive
./three.sh -6
Argument is negative ✓

Expt. No. 03

Date 25/10/21

Page No. 3

Shell script to check whether a number is positive,
negative or zero.

```
#!/bin/sh
if [ $1 -eq 0 ]
then
echo "Argument is equal to zero"
elif [ $1 -lt 0 ]
then
echo "Argument is negative"
else
echo "Argument is positive"
```

if
fi

Teacher's Signature : _____

Expt. No. 04

```
chmod 777 four.sh
./four.sh
Enter the 3 numbers
1
2
3
3 is the greatest
./four.sh
Enter 3 numbers
-1
0
-5
0 is the greatest
./four.sh
Enter 3 numbers
10
4
-3
10 is the greatest
```



Shell script to find the 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 greatest"
elif [ $b -gt $c -a $b -gt $a ]
then
echo "$b is the greatest"
elif [ $c -gt $a -a $c -gt $b ]
then
echo "$c is the greatest"
```

Teacher's Signature : _____

Expt. No. 05

```

chmod 777 five.sh
./five.sh
Enter the number
5
Factorial is 120
./five.sh
Enter the number
6
Factorial is 720

```

Shell script to find the factorial of a number

```

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

```

N
S/1/1

Teacher's Signature :

chmod 777 ex.sh
./ex.sh
Enter the basic salary
1000
The gross salary is: 1300.0

Expt. No. 06

Date 8/11/21

Page No. 6

Small script to compute the gross salary of an employee

```
#!/bin/bash
echo "Enter the basic salary"
read b
da='echo 0.1 * $b | bc'
hra='echo 0.2 * $b | bc'
gross='echo $b + $da + $hra | bc'
echo "The gross salary is: $gross"
```

N
8/11/21

Teacher's Signature :

Expt. No. 07

chmod 777 swen.sh

./swen.sh

Enter the temperature in Fahrenheit

32

Temperature in Celsius is 0

./swen.sh

Enter the temperature in Fahrenheit

54

Temperature in Celsius is 12.2210

Shell script to convert the temperature from Fahrenheit
to Celsius

#!/bin/sh

echo "Enter the temperature in Fahrenheit"

read f

g=\$((scale=4; \$f/9) | bc')

c=\$(((\$f - 32) * 5/9) | bc')

echo "Temperature in Celsius is \$c"

8/11/21

Teacher's Signature :

Expt. No. 08

Shell script to perform arithmetic operations on given 2 numbers.

```
#!/bin/sh
echo "Enter the 2 numbers"
read a
read b
echo "Addition: $(($a + $b))"
echo "Subtraction: $(($a - $b))"
echo "Multiplication: $(($a * $b))"
N g='echo "Scale=2; $a/$b;" | bc'
$N echo "Division: $g"
echo "Modulus: $(($a % $b))"
```

chmod 777 eight.sh
./eight.sh
Enter the 2 numbers
4
3
Addition: 7
Subtraction: 1
Multiplication: 12
Division: 1.33
Modulus: 1

Expt. No. 09

Shell script to find the sum of even numbers upto n

```
#!/bin/bash
echo "Enter the number"
read n
sum=0
for((i=1; i<=n; i++))
do
if [ $((i%2)) -eq 0 ]
then
sum=$((sum+i))
fi
done
```

N echo "Sum of even numbers upto \$n is \$sum"

15/11/21

Teacher's Signature :

chmod 777 ten.sh
./ten.sh

111
112
113
121
122
123
131
132
133
211
212
213
221
222
223
231
232
233
311
312
313
321
322
323
331
332
333

Date 15/11/21
Expt. No. 10
Page No. 10

Shell script to print all the combinations of numbers

123.

```
#!/bin/bash
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
```

111
112
113

Teacher's Signature :

Expt. No. 11

Shell script to find the power of a number

```
#!/bin/bash
```

```
echo "Enter the number and its power"
```

```
read n
```

```
read p
```

```
pro=1
```

```
for((i=1; i<=p; i++))
```

```
do
```

```
pro=$((pro * n))
```

```
done
```

```
echo "Power is $pro"
```

15/11/21

Teacher's Signature :

chmod 777 eleven.sh
 ./eleven.sh
 Enter the number and its power
 2
 10
 Power is 1024
 ./eleven.sh
 Enter the number and its power
 5
 2
 Power is 25 ✓

chmod 777 twelve.sh
./twelve.sh
Enter the number
3
Sum of all natural numbers upto 3 is 6
./twelve.sh
4
Sum of all natural numbers upto 4 is 10

Expt. No. 12

Date 15/11/21

Page No. 12

Shell script to find the sum of n natural numbers

```
#!/bin/bash
echo "Enter the number"
read n
sum=0
for((i=1; i<=n; i++))
do
sum=$((sum+i))
done
echo "Sum of all natural numbers upto $n is $sum"
```

Teacher's Signature : _____

Expt. No. 13

Output:
 Enter the SEE marks for 50
 50
 Enter the CIE marks for 50
 50
 Grade: D
 Enter the SEE marks for 50
 10
 Enter the CIE marks for 50
 10
 Grade: E
 Enter the SEE marks for 50
 20
 Enter the CIE marks for 50
 20
 Grade: F
 Enter the SEE marks for 50
 50
 Enter the CIE marks for 50
 50
 Grade: A
 Enter the SEE marks for 50
 50
 Enter the CIE marks for 50
 50
 Grade: B
 Enter the SEE marks for 50
 50
 Enter the CIE marks for 50
 50
 Grade: C

Shell script to display the pass class of a student.

```

#!/bin/bash
pass=0
fail=0
for((i=0; i<6; i++))
do
  echo "Enter the SEE marks for 50"
  read see
  echo "Enter the CIE marks for 50"
  read cie
  total=$((cie+$see))
  if [ $total -ge 90 ]
  then
    echo "Grade: S"
    pass=$((pass+1))
  elif [ $total -ge 80 ]
  then
    echo "Grade: A"
    pass=$((pass+1))
  elif [ $total -ge 70 ]
  then
    echo "Grade: B"
    pass=$((pass+1))
  elif [ $total -ge 60 ]
  then
    echo "Grade: C"
    pass=$((pass+1))
  else
    echo "Grade: F"
  fi
done
echo "Total number of students who passed = $pass"
echo "Total number of students who failed = $fail"
```

Teacher's Signature :

Number of subjects passed: 6
Number of subjects failed: 0

Date 29/11/21

Page No. 14

Expt. No. _____

```
if [ $total -ge 50 ]
then
echo "Grade: D"
pass = $(($pass+1))
elif [ $total -ge 40 ]
then
echo "Grade: E"
pass = $(($pass+1))
else
echo "Grade: F"
fail = $(($fail+1))
fi
done
echo "Number of subjects passed: $pass"
echo "Number of subjects failed: $fail"
```

Teacher's Signature : _____

Output: ./fourteen.sh
Enter the value of n

2
fibonacci series upto 2 numbers:

0
1
./fourteen.sh
Enter the value of n

3
fibonacci series upto 3 numbers:

0
1
1
./fourteen.sh
Enter the value of n

4
fibonacci series upto 4 numbers:

0
1
1
2

Expt. No. 14

Date 29/11/21

Page No. 15

Shell script to find the fibonacci series upto n.

```
#!/bin/bash
echo "Enter the value of n"
read n
if [ $n -eq 0 ]
then
exit
fi
echo "fibonacci series upto $n numbers:"
a=0
b=1
if [ $n -eq 1 ]; then
echo "0"
elif [ $1 -eq 2 ]; then
echo "0"
echo "1"
fi
for((i=2; i<n; i++))
do
c=$((a+b))
echo "$c"
a=$b
b=$c
done
```

Teacher's Signature :

Output:
*/fifteen.sh
Enter the string
good morning
Number of vowels is 4
*/fifteen.sh
Enter the string
action
Number of vowels is 5
*/fifteen.sh
Enter the string
sdqg
Number of vowels is 0

Expt. No. 15

Date 29/11/21

Page No. 16

Shell script to count the number of vowels of a string

```
#!/bin/bash
echo "Enter the string"
read s
count=0
l=`expr $s : '\.*'`
for ((i=0; i<${#l}; i++))
do
    c=`expr ${l:$i:1}`
    if [ "$c" = 'a' -o "$c" = 'e' -o "$c" = 'i' -o "$c"
        = 'o' -o "$c" = 'u' ]
    then
        vowel=$((count+1))
    fi
done
echo "Number of vowels is $count"
```

Teacher's Signature : _____

./fifteen.sh

Number of lines:

15

Number of words:

54

Number of characters:

287

Expt. No. 16

Date 29/11/21

Page No. 17

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

```
#!/bin/bash
```

```
echo "Number of lines:"
```

```
cat fifteen.sh | wc -l
```

```
echo "Number of words:"
```

```
cat fifteen.sh | wc -w
```

```
echo "Number of characters:"
```

```
cat fifteen.sh | wc -c
```

Teacher's Signature : _____

Expt. No. 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;
}
```

Teacher's Signature:

SHLL = /bin/bash
SESSION_MANAGER = local/ubuntu:@/tmp/.ICE-unix/4332.
QT_ACCESSIBILITY = 1
COLORTERM = truecolor
XDG_CONFIG_DIRS = /etc/xdg/xdg-ubuntu:/etc/xdg
SSH_AGENT_LAUNCHER = gnome-keyring
XDG_MENU_PREFIX = gnome-
GNOME_DESKTOP_SESSION_ID = this-is-deprecated
GNOME_DESKTOP_SESSION_MODE = ubuntu
SSH_AUTH_SOCK = /run/user/999/keyring/ssh
XMODIFIERS = @im-ibus
DESKTOP_SESSION = ubuntu
GDK_MODULES = gail:atk-bridge
PWD = /home/ubuntu/.wp
LOGNAME = ubuntu
XDG_SESSION_DESKTOP = ubuntu
XDG_SESSION_TYPE = x11
XDG_AGNENT_INFO = /run/user/999/gnupg/gpg-agent:0
XAUTHORITY = /run/user/999/gdm/Xauthority
WINDOWPATH = 0
HOME = /home/ubuntu
USERNAME = ubuntu
IM_CONFIG_PHASE = 1
LANG = en_US.UTF-8
LINES = 24
COLS = 80
TERM = xterm-256color
VTE_VERSION = 6.0.3
GNOME_TERMINAL_SCREEN = org.gnome.Terminal/screen/64da87e8-c345-43c2-8223-1f6f
LESSCLOSE = /usr/bin/lesspipe %s %s
XDG_SESSION_CLASS = Wayland
TERM = xterm-256color
LESSOPEN = /usr/bin/lesspipe %s
USER = ubuntu
GNOME_TERMINAL_SERVICE = 1.303
DISPLAY = 0.0
SHLVL = 1
XDG_DATA_DIRS = /usr/share/ubuntu:/usr/local/share/:/var/lib/maapd/distro
PATH = /usr/local/bin:/usr/bin:/bin:/usr/sbin:/bin:/usr/games:/usr/:/snap/bin
GID_MESSION = ubuntu
DISPLAY = :0.0
ADDRESS = unix:path=/run/user/999/bus
OLDPWD = /home/ubuntu

Output:

./a.out
Usage: ./a.out [-S] <org-file> <new-link>
./a.out tayc tay1.c
Hard link created
./a.out -S tayc tay2.c
Symbolic link created

Expt. No. 18

Date 3/11/22

Page No. 19

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");
    else
        if( (argc == 4) && ( 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 ("Hardlink created\n");
    return 0;
}
```

Teacher's Signature : _____

Output:
\$./a.out
System supports job control
System supports saved set-UID and saved set-GID
chown-restricted option is 0
Pathname-trunc option is 1
Disable character for terminal files is 0

Expt. No. 19

Date 31/1/22

Page No. 20

Write C/C++ POSIX compliant program that prints the
POSIX defined configuration options supported on
any given system using feature test macro.

```
#define _POSIX_SOURCE
#define _POSIX_C_SOURCE 199309L
#include <stropts.h>
#include <sys/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
```

Teacher's Signature : _____

```
#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
#ifndef 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:

Terminal 1:

```
gcc lab0.c  
./a.out  
USAGE: ./file [arg]  
./a.out fifo1 "This is USP Lab"
```

Terminal 2:

```
./a.out fifo1  
This is USP Lab
```

Expt. No. 20

Date 3/1/22

Page No. 22

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 <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: ./file [arg]\n", argv[0]);  
        return 0;  
    }  
    mkfifo ( argv[1], S_IFIFO | S_IRWXU | S_IROUX | S_IWUSR );  
    if (argc == 2)  
    {  
        fd = open ( argv[1], O_RDONLY | O_NONBLOCK );  
        while (read ( fd, buf, sizeof (buf) ) > 0)  
            printf ("%s", buf);  
        else  
    }  
    if (fd = open ( argv[1], O_WRONLY ))  
        write ( fd, argv[2], strlen ( argv[2] ));  
    close (fd);  
}
```

Teacher's Signature : _____