



# B M S. COLLEGE OF ENGINEERING

(Autonomous Institution)

## LABORATORY CERTIFICATE

# RECORD OF PRACTICAL WORK

NAME : Derek Stanley Kannathe  
SUBJECT : U.S.P LAB  
SEMESTER : IV BRANCH : CSE  
ROLL NO : 1BNI9C8045 USN : 1BNI9C1045

# Particulars of the Experiments Performed CONTENTS

Expt No.	Date	Experiment	Marks Obtained	Page No.
1	25/10/21	Prog 1 : Leap Year		1
2	25/10/21	Prog 2 : Area of a circle		2
3	25/10/21	Prog 3 : Positive negative zero		3
4	25/10/21	Prog 4 : Biggest of three numbers		4
5	8/11/21	Prog 5 : Factorial of a number		5
6	8/11/21	Prog 6 : Calculate Salary		6
7	8/11/21	Prog 7 : Fahrenheit to Celsius		7
8	8/11/21	Prog 8 : Arithmetic Operation		8
9	15/11/21	Prog 9 : Sum of even numbers		9
10	15/11/21	Prog 10 : Print combination of 123		10
11	15/11/21	Prog 11 : Find power of a number		11
12	15/11/21	Prog 12 : Find sum of n natural numbers		12
13	29/11/21	Prog 13 : Display Past class of a student		13
14	29/11/21	Prog 14 : Find Fibonacci series upto n		15
15	29/11/21	Prog 15 : Check number of three words & characters <small>vowels</small>		17
16	29/11/21	Prog 16 : check no of lines, words & characters		16
17	3/1/22	Prog 17 : Output content of environment variable		18
18	3/1/22	Prog 18 : Emulate 'ls' command		20
19	3/1/22	Prog 19 : Print posix defined config options		21
20	3/1/22	Prog 20 : Demonstrates 11^c b/w reader writer process		23

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

```
#!/bin/sh
echo "Enter a year :"
read year
if [ $(year % 4) -eq 0 -a $(year % 100) -ne 0 ]
then
    echo "Leap Year"
else
    if [ $(year % 400) -eq 0 ]
    then
        echo "Leap Year"
    else
        echo "Non Leap Year"
    fi
fi
```

DBSERTATION:

Enter a year:

300

Non leap year

Enter a year:

400

Leap Year

Enter a year:

1964

Leap Year.



Q Shell script to find area of circle.

```
#! /bin/sh
echo "Enter the radius of the circle:"
read r
area = `expr $r \* $r \* 3.14 | bc`
echo "Area: "
echo $area
```

OBSERVATION :

Enter the radius of the circle :

7

Area :

153.86

Q

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

```
#!/bin/rh  
echo "Enter a number : "  
read n  
if [ $n -gt 0 ]  
then  
    echo "Positive"  
else  
    if [ $n -lt 0 ]  
    then  
        echo "Negative"  
    else  
        echo "Zero"  
    fi  
fi
```

OBSERVATION:

=

Enter a number:

21

Positive

Enter a number:

0

Zero

Enter a number:

-9

Negative

d Shell script to find the biggest of three numbers.

~~#!/bin/bash~~

echo "Enter first number : "

read a

echo "Enter second number : "

read b

echo "Enter third number : "

read c

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

~~then~~

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

~~then~~

echo \$a

else

echo \$c

fi

else

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

~~then~~

echo \$b

else

echo \$c

fi

fi

OBSERVATION

Enter first number:

2

Enter second number:

5

Enter third number:

3

5

d) Shell script to find the factorial of a number.

```
#!/bin/sh
echo "Enter the number"
read fno
fact = 1
n = $fno
while [ $fno -gt 1 ]
do
    fact = $(($fact * $fno))
    fno = $(($fno - 1))
done
echo "Factorial of $n is $fact"
```

## OBSERVATION:

Enter the number  
6  
Factorial of 6 is 720

Q Shell script to compute gross salary of an employee

#!/bin/bash

echo "Enter basic salary:"

read bSal

hra = `expr \$bSal \* 20 / 100`

da = `expr \$bSal \* 10 / 100`

gSal = `expr \$bSal + \$hra + \$da`

echo "Gross salary is : \$gSal"

OBSERVATION:

Entire basic salary:

5000

Gross salary is: 6500

Q Shell script to convert Farenheit to Celsius.

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

```
echo "Enter temperature in Farenheit"
```

```
read f
```

```
c = $(($f - 32) * 5 / 9))
```

```
echo "$f farenheit is $c in celsius"
```

OBSERVATION:

Find temperature in Fahrenheit:

-40

-40 farenheit is -40 in celsius

Q

Shell script to compute/perform arithmetical operation on given two numbers.

```

#!/bin/sh
echo "Enter first number: "
read fno
echo "Enter second number: "
read sno
echo "Enter number as choice : 1. ADD 2. SUBTRACT 3. MULTIPLY
4. DIVIDE"
read choice
case $choice in
    1) ans=$((fno+sno));;
    2) ans=$((fno-sno));;
    3) ans=$((fno*sno));;
    4) ans=`expr $fno / $sno`;;
    *) echo "Invalid option"
esac
echo "Result of the operation is : $ans"

```

OBSERVATION

Enter first number:

6

Enter the second number:

3

Enter number as choice : 1. ADD 2. SUBTRACT 3. MULTIPLY 4. DIVIDE

4

Result of the operation is: 2 ✓

d

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

#!/bin/bash

if [ \$# -eq \$1 ]

do

sum=\$((sum+c))

c=\$((c+2))

done

echo "Sum of even number till \$1 is : \$sum"

fi

Observation

. /prog9 - sumEven. sh 2

Sum of even number till 8 n : 20

Q Shell script to print the 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

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

echo "\$i\$j\$k"

fi

/ done

done

done

Observation

123

132

213

231

312

321

Q Shell script to find the power of a number.

#!/bin/sh

if [ \$# -eq 2 ]

then

temp = \$2

num = 1

while [ \$temp -gt 0 ]

do

num=`expr \$num \\* \$1`bc

temp = \$((\$temp - 1))

done

echo "The power of \$1 to \$2 is : \$num"

fi

Observation

\$ ./prog11 - power.sh 3 6

The power of 3 to 6 is : 729

Q Shell script to find the sum of n natural numbers.

#!/bin/sh

if [ \$# -eq 1 ]

then

c=1

sum=0

while [ \$c -le \$1 ]

do

sum=\$((sum + c))

c=\$((c+1))

done

echo " sum of natural numbers : \$sum"

fi

✓  
MM  
5/11/21

Observation

./prog12 - sumN.sh 10

sum of natural numbers : 55

Q Shell script to display the pass class of a student

```

#!/bin/sh
echo "Enter marks for CIE and SEE"
for i in 0 1 2 3 4 5
do
    echo "Subject $((i+1)):"
    echo "CIE:"
    read cie[$i]
    echo "SEE:"
    read see[$i]
    total[$i]=$(($cie[$i]+$see[$i]))
done
echo "--- GRADES ---"
for i in 0 1 2 3 4 5
do
    echo "Subject $((i+1)):"
    if [ ${total[$i]} -gt 90 ]; then
        echo "S"
    else; if [ ${total[$i]} -gt 80 ]; then
        echo "A"
    else; if [ ${total[$i]} -gt 70 ]; then
        echo "B"
    else; if [ ${total[$i]} -gt 60 ]; then
        echo "C"
    else; if [ ${total[$i]} -gt 50 ]; then
        echo "D"
    else; if [ ${total[$i]} -gt 40 ]; then
        echo "E"
    fi
    fi
    fi
    fi
    fi
    fi
fi

```

Teacher's Signature :

echo "E"

else

echo "F"

fi

fi

fi

fi

fi

done

Enter marks for CIE and SEE

Subject 1:

CIE:

34

SEE:

45

Subject 2:

CIE:

36

SEE:

27

Subject 3:

CIE:

38

SEE:

39

Subject 4:

CIE:

47

SEE:

36

Subject 5:

CIE:

49

SEE:

47

Subject 6:

CIE:

10

SEE:

26

--- GRADES -- -

Subject 1:

B

Subject 2:

C

Subject 3:

B

Subject 4:

A

Subject 5:

S

Subject 6:

Fail

Q Shell script to find the Fibonacci series up to n.

#!/bin/sh

echo "Enter a Number"

read n

n=\$((n-2))

a=0

b=1

echo "\$a"

echo "\$b"

while [ \$n -gt 0 ]

do

temp=\$b

b=\$((a+b))

a=\$temp

echo "\$b"

n=\$((n-1))

done

Enter a Number

60-11235

Q Shell script to check number of lines, words, characters in a

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

```
echo "Enter the filename"
```

```
read file
```

```
c='cat $file | wc -c'
```

```
w='cat $file | wc -w'
```

```
l='cat $file | wc -l'
```

```
s='grep -c "\." $file'
```

```
echo "Number of characters in $file is $c"
```

```
echo "Number of words in $file is $w"
```

```
echo "Number of lines in $file is $l"
```

Enter the filename  
prog14-Fibonacci.ch

Number of characters in prog14-Fibonacci.ch is 164

Number of words in prog14-Fibonacci.ch is 28

Number of lines in prog14-Fibonacci.ch is 20

Q Shell script to count the number of vowels of a string

#!/bin/sh

echo "Enter a string"

read str

len=\$(expr length \$str)

count=0

while [ \$len -gt 0 ]

do

ch=\$(echo \$str | cut -c \$len)

case \$ch in [aeiouAEIOU]

count=\$(( \$count + 1 ))

echo \$ch

;;

esac

len=\$(( \$len - 1 ))

done

echo \$count

Enter a string  
Good Morning

4

Q Write a C/C++ program to that outputs the content of its environment list.

```
#include<stdio.h>
```

```
int main()
```

```
{
```

```
    int i;
```

```
    char **ptr;
```

```
    extern char **environ;
```

```
    for(Cptr = environ; *ptr != 0; #ptr++)
```

```
        printf("%s\n", *ptr);
```

```
    return 0;
```

```
}
```

SHELL = /bin/bash

SESSION\_MANAGER=local/bmsec:@/tmp/ICE-unix/1188,unix/bmsec:/tmp/ICE-unix/1188

AT\_ACCESSIBILITY=1

COLORTERM = TrueColor

XDG\_CONFIG\_DIRS = /etc/xdg/ndg-ubuntu:/etc/xdg

SSH\_AGENT\_LAUNCHER = gnome-keyring

XDG\_MENU\_PREFIX = gnome.

GNOME\_DESKTOP\_SESSION\_ID = this-is-deprecated

LANGUAGE = en\_IN:en

GNOME\_SHELL\_SESSION\_MODE = ubuntu

SSH\_AUTH\_SOCK = /run/user/1000/keyring/ssh

XMODIFIERS=@im=ibus

DESKTOP\_SESSION = ubuntu

GTK\_MODULES = gail:atk-bridge

PWD = /home/bmsec/Desktop

LOGNAME = bmsec

XDG\_SESSION\_DESKTOP = ubuntu

XDG\_SESSION\_TYPE = wayland

SYSTEM\_EXEC\_PID = 8238

XAUTHORITY = /run/user/1000/.multi-xwaylandauth.5NSD42

HOME = /home/bmsec

USERNAME = bmsec

IM\_CONFIG\_PHASE = 1

LANG = en\_IN

LS\_COLORS = sc=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;

XDG\_CURRENT\_DESKTOP = ubuntu:GNOME

VTE\_VERSION = 6402

WAYLOAD\_DISPLAY = wayload=0

G\_ENABLE\_DIAGNOSTIC = 0

GNOME\_TERMINAL\_SCREEN = /usr/lib/gnome-terminal/screen/33d4-33-4nd  
e674643cab9

INVOCATION\_ID = 3ddfc640ed56d475l5fsf07674ec0b8005

MANAGERID = 1052

GNOME\_SETUP\_DISPLAY = :1

LSSDOSE = /usr/bin/lesippe 0.0s 0.0s

OLDPWD = /home/bmrcel/shell

PATH = /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin

CONNECTION = ubuntu

JOURNAL\_STREAM = 8:48740

DISPLAY = :0

USER = bmrcel

Q 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") == 0))
    {
        printf("USAGE: ./a.out [-s] <org.file> <new.file> \n");
        return 1;
    }

    if (argc == 4)
    {
        if (symlink(argv[2], argv[3]) == -1)
            printf("cant able to 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("Hard Link created \n");
    }

    return 0;
}
```

```
$ gcc prog18-1n.c  
$ ./a.out  
USAGE ./a.out [-s] <arg-file> <new-file>  
$ ./a.out Hello.txt HardHello.txt  
Hard Link created  
$ ./a.out -s Hello.txt softHello.txt  
Symbolic link created
```

Q C POSIX compliant program to print the POSIX defined config options supported on given system"

```
#define _POSIX_SOURCE
#define _POSIX_C_SOURCE 199309L
#define 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 doesn't supports saved set-UID, set-GID\n");
    #endif

    #ifdef _POSIX_CHOWN_RESTRICTED
        printf("chown-restricted option is %d\n", _POSIX_CHOWN_RESTRICTED);
    #else
        printf("System doesn't support chown-restricted option\n");
    #endif

    #ifdef _POSIX_NO_TRUNC
        printf("Pathname trunc option is %d\n", _POSIX_NO_TRUNC);
    #else
        printf("System doesn't support posix no trunc option");
    #endif
}
```

```
#ifdef _POSIX_VDISABLE  
printf(" Disabled characters for terminal files is %d \\n", _POSIX_  
      VDISABLE);  
#else  
printf(" System does not support _POSIX_VDISABLE \\n");  
#endif  
return 0;  
}
```

\$ gec prog19 const.c -fPIC -o main.o -c  
\$ ./a.out

System Supports JOB control

System supports saved SEL-VID, SEL-GID

shown - restricted option is 0

Pathname trunc option is L

Disabled character for terminal files is 0

Q C Program to demonstrate the IPC between reader & writer process using mkfifo, open, read, write and close api in program.

```
#include <sys/types.h>
#include <unistd.h>
#include <fcntl.h>
#include <sys/std.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 ./a.out <file> [arg] \n", argv[0]);
        return 0;
    }
    if (argc == 2)
    {
        fd = open(argv[1], O_RDONLY | O_NONBLOCK);
        while (read(fd, buf, sizeof(buf))) printf("%c", buf);
    }
    else
    {
        mkfifo(argv[1], S_IFIFO | S_IRWXU | S_IWXA | S_RWXO);
        fd = open(argv[1], O_WRONLY);
        write(fd, argv[2], strlen(argv[2]));
    }
    close(fd);
}
```

## Terminal 2 Writer Process

\$ gcc prog20-comm.c

\$ ./a.out

USAGE: ./a.out <file> [<arg>]

+ ./a.out pipeName "Good Morning"

## Terminal 2 Reader Process

\$ ./a.out > pipeName

Good Morning