B M S. COLLEGE OF ENGINEERING

(Autonomous Institution)

RECORD OF PRACTICAL WORK

NAME : Neelam Godihal

SUBJECT : USP

SEMESTER : V BRANCH : CSE

ROLL NO : USN : 1BM19CS220

Date 25-10-21

Expt. No. 01 Page No. 01

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

```
#!/bin/sh
echo "Enter year"
read year

if [ $((year %100)) -eq 0 ]
then
    if [ $((year %400)) -eq 0 ]
    then
        echo "$year is a leap year"
    else
        echo "$year is not a leap year"
    fi
elif [ $((year %4)) -eq 0 ]
then
    echo "$year is a leap year"
else
    echo "$year is not a leap year"
fi
```

Teacher's Signature :

Output:

```
Enter year
1600
1600 is a leap year

Enter year
1800
1800 is not a leap year
```

Date 25-10-21

Expt. No. 02 Page No. 02

Shell script to find the area of a circle

```
#!/bin/sh
echo "Enter the radius"
read r
pi = 3.14
area = 'echo $pi\* $r\* $r | bc'
echo "The area of the circle is" $area
```

Teacher's Signature :

Output:

Enter the radius
2
The area of the circle is 12.56

Expt. No.

Date 25-10-21

Expt. No. 03 Page No. 03

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

```
#!/bin/sh

echo "Enter the number"
read n

if [ $n -gt 0 ]
then
  echo "Positive"
elif [ $n -lt 0 ]
then
  echo "Negative"
else
  echo "Zero"
fi
```

Teacher's Signature :

Output:

```
Enter the number
5
Positive

Enter the number
0
Zero

Enter the number
-1
Negative
```

Date 25-10-21

Expt. No. 04 Page No. 04

Shell script to find the biggest of three numbers.

```
#!/bin/sh

echo "Enter the 3 numbers"
read a b c
if [ $a -gt $b -a $a -gt $c ]
then
    echo "A is the biggest"
elif [ $b -gt $a -a $b -gt $c ]
then
    echo "B is the biggest"
else
    echo "C is the biggest"
fi
```

Teacher's Signature : _____

Output:

```
Enter the 3 numbers
8 3 1
A is the biggest

Enter the 3 numbers
10 21 3
B is the biggest

Enter the 3 numbers
3 7 9
C is the biggest
```

Date 08-11-21

Expt. No. 05 Page No. 05

Shell script to find the factorial of a number

```
#!/bin/bash
echo "Enter the number"
read n

fact=1

for((i=2;i<=n;i++))
do
fact=$((fact*i))
done

echo "Factorial is: "$fact
```

Teacher's Signature :

Output :

Enter the number
4
Factorial is : 24

Expt. No.

Date 08-11-21

Expt. No. 06 Page No. 06

Shell script to compute the gross salary of an employee.

```
#!/bin/sh

echo "Enter the basic salary"
read basic

hra='echo 0.2 * $basic | bc'
da='echo 0.1 * $basic | bc'
gross='echo $basic + $hra + $da | bc'

echo "The gross salary is" $gross
```

Teacher's Signature :

Output :

Enter the basic salary

1000

The gross salary is 1300.0

Date 09-11-21

Expt. No. 07 Page No. 07

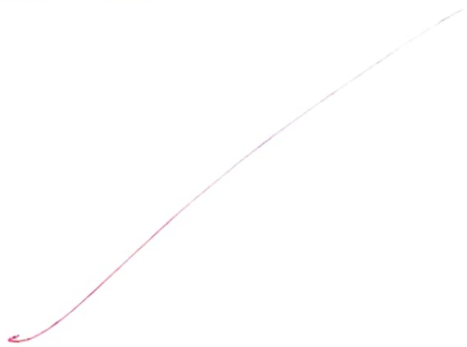
Shell script to convert the temperature Fahrenheit to Celsius.

```
#!/bin/sh

echo "Enter temperature in Fahrenheit"
read f

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

echo "The temperature in celsius is "$c
```

Teacher's Signature : _____

Output:

```
Enter temperature in Fahrenheit
32
The temperature in celcius is 0.
```



Date 08-11-22

Expt. No. 08 Page No. 05

Shell script to perform arithmetic operations on given two numbers.

```
#!/bin/sh

echo "Enter number 1"
read num1
echo "Enter number 2"
read num2

echo "Please enter your choice : 1. Addition
2. Substraction 3. Multiplication 4. Division
5. Mod"
read option

case $option in
1) echo "sum is $((num1+num2))"
;;
2) echo "Difference is $((num1-num2))"
;;
3) echo "Product is $((num1*num2))"
;;
4) echo "Division is $((num1/num2))"
;;
5) echo "Module is $((num1%num2))"
;;
*) echo "Enter a valid number"
;;
esac
```

Teacher's Signature : _____

Output:

```
Enter number 1
6
Enter number 2
3
Please enter your choice : 1. Addition
2. Subtraction 3. Multiplication 4. Division 5. Mod
2
Difference is 3

Enter number 1
5
Enter number 2
2
Please enter your choice : 1. Addition
2. Subtraction 3. Multiplication 4. Division 5. Mod
5
Modulus is 1
```

Date 15-11-21

Expt. No. 09 Page No. 09

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

```
#!/bin/sh

echo "Enter n"
read n

i=1
sum=0

while [ $i -le $n ]
do
    if [ $((i%2)) -eq 0 ]
    then
    sum=$((sum+i))
    fi
    i=$((i+1))
done

echo "The sum of even numbers till n
      is " $sum.
```

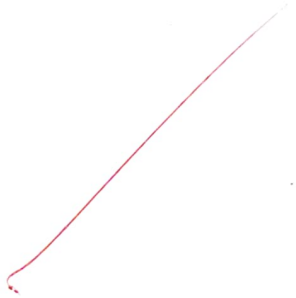
Teacher's Signature : _____

Output:

Enter n

4

The sum of even numbers till 4 is 6

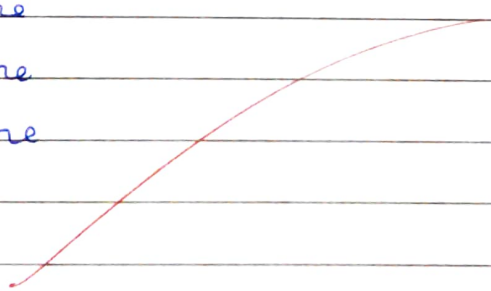


Date 15-11-22

Expt. No. 10 Page No. 10

Shell script to print the combinations of numbers 123.

```
#!/bin/bash

for (( i=1 ; i<4; i++))
do
for (( j=1 ; j<4; j++))
do
for (( k=1; k<4; k++))
do
   if [ $i -ne $j -a $j -ne $k -a $i -ne $k ]
   then
        echo $i $j $k
   fi
done
done
done
```



Teacher's Signature :

Output :

123
132
213
231
312
321

Date 15-11-21

Expt. No. 11 Page No. 11

Shell script to find the power of a number

```
#!/bin/sh

echo "Enter number"
read num
echo "Enter power"
read pow

res=1

while [ $pow -ne 0 ]
do
    res=$((res*num))
    pow=$((pow-1))
done

echo "The result is " $res
```

Teacher's Signature :

Output:

```
Enter number
2
Enter power
3
The result is 8
```

Date 15-11-21

Expt. No. 12 Page No. 12

Shell script to find the sum of n number

```
#!/bin/sh

echo "Enter n"
read n

i=1
sum=0

while [ $i -le $n ]
do
    sum=$((sum+i))
    i=$((i+1))
done

echo "The sum of n natural numbers is
" $sum
```

Teacher's Signature :

Output:

Enter n
5
The sum of 5 natural numbers is 15

Date 13-12-21

Expt. No. 13 Page No. 13

Shell script to display the pass class of a student

```
#!/bin/sh

i=1
pass=0
fail=0

while test $i -le 6
do
  echo "\nEnter the CIE marks out of 50 for subject $i:"
  read cie
  echo "\nEnter the SEE marks out of 100 for subject $i:"
  read see
  see=$((see/2))

  total=`echo $cie+$see | bc`

  if [ $total -ge 90 ]
  then
    echo "S grade"
    pass=$((pass+1))
  elif [ $total -ge 80 ]
  then
    echo "A grade"
```

Teacher's Signature :

Date _____

Expt. No. _____ Page No. 14

```
pass=$((pass+1))
elif [ $total -ge 70 ]
then
    echo "B grade"
    pass=$((pass+1))
elif [ $total -ge 60 ]
then
    echo "C grade"
    pass=$((pass+1))
elif [ $total -ge 50 ]
then
    echo "D grade"
    pass=$((pass+1))
elif [ $total -ge 40 ]
then
    echo "E grade"
    pass=$((pass+1))
else
    echo "F grade"
    fail=$((fail+1))
fi
i=$((i+1))
done

echo "\nNumber of subjects passed: $pass"
echo "\nNumber of subjects failed: $fail"
```

Teacher's Signature : _____

Output:

```
Enter the CIE marks out of 50 for subject 1
45
Enter the SEE marks out of 100 for subject 1
96
S grade
Enter the CIE marks out of 50 for subject 2
42
Enter the SEE marks out of 100 for subject 2
90
A grade
Enter the CIE marks out of 50 for subject 3
40
Enter the SEE marks out of 100 for subject 3
82
A grade
Enter the CIE marks out of 50 for subject 4
37
Enter the SEE marks out of 100 for subject 4
76
B grade
Enter the CIE marks out of 50 for subject 5
40
Enter the SEE marks out of 100 for subject 5
80
A grade
Enter the CIE marks out of 50 for subject 6
44
Enter the SEE marks out of 100 for subject 6
```

90

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

Date 13-12-21

Expt. No. 14 Page No. 15

Shell script to find the Fibonacci series upto n.

```
#!/bin/sh

echo "Enter the value of n:"
read n

a=0
b=1
echo "\n Fibonacci series of $n terms:"
echo "$a\n$b"

i=3

while [ $i -le $n ]
do
        c=`echo $a+$b|bc`
        echo "$c"
        a=$b
        b=$c
        i=$((i+1))
done
```

Teacher's Signature :

Output:

i) Enter the value of n
3

Fibonacci series of 3 terms:
0
1
1

ii) Enter the value of n
5

Fibonacci series of 5 terms
0
1
1
2
3

Date 13-12-21

Expt. No. 15 Page No. 16

Shell script to count the number of lines, words, characters ~~words~~ of a string file.

```
#!/bin/sh

echo "Enter the filename :"
read fname

wrctr = 'wc -w < $fname'
lctr = 'wc -l < $fname'
cctr = 'wc -c < $fname'

echo "Number of lines : $lctr"
echo "Number of words : $wrctr"
echo "Number of characters : $cctr"
```

Teacher's Signature :

Output :

Enter the filename :
test.txt

Number of lines : 19
Number of words : 60
Number of characters : 192

Date 13-12-21

Expt. No. 16 Page No. 17

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

```
#!/bin/sh

echo "Enter the string :"
read s

len = `expr "$s" | wc -c`        # to get length
len = $((len-1))

ctr=0
while [ $len -gt 0 ]
do
    ch = `expr "$s" | cut -c $len`
    case $ch in

    [AEIOU, aeiou])  ctr = $((ctr+1))
                     ;;

    esac
    len = $((len-1))
done

echo "Number of vowels is $ctr".
```

Teacher's Signature : _____

Output ;

i) Enter the string :
Neelam
Number of vowels is 3
ſ
ii) Enter the string :
Engineering
Number of vowels is 5

Date 08-01-22

Expt. No. 17 Page No. 18

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 : _____

Output :

SSH_AGENT_PID = 3207
HOSTNAME = localhost.localdomain
DESKTOP_STARTUP_ID =
SHELL = /bin/bash
TERM = xterm
HISTSIZE = 1000
KDE-NO_IPV6 = 1
GTK_RC_FILES = /etc/gtk/gtkrc : /root/.gtkrc-1.2-
gnome2
WINDOWID = 44040273
OLDPWD = /root/tar
QTDIR = /usr/lib/qt-3.3
QTINC = /usr/lib/qt-3.3/include
USER = root
DESKTOP_SESSION = default
PATH = /usr/lib/qt-
3.3/bin : /usr/kerberos/sbin : /usr/kerberos/bin : /
usr/local/sbin : /usr/local/bin : /sbin : /bin : /usr/
sbin : /usr/bin
GDM_XSERVER_LOCATION = local
INPUTRC = /etc/inputrc
PWD = /root/tar/mspl
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
LOGNAME = root

Date 03-02-22

Expt. No. 18

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");
    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 )
      printf("Cannot create hard link\n");
    else
      printf("Hard link created\n");
  }
  return 0;
}
```

Teacher's Signature : _____

Output:

```
$ ./program18
Usage : ./a.out [-s] <org-file> <new_link>

$ ./program18 test.txt hard.txt
Hard link created

$ ./program18 -s text.txt soft
Symbolic link created
```

Expt. No. 19

Date 03-01-22

Page No. 20

Write a C/C++ POSIX compliant program that prints the POSIX defined configuration options 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");
#else
    printf("System does not support
            saved set-UID and saved
            set-GID\n");
#endif
```

Teacher's Signature : _____

Date _____

Expt. No. _____ Page No. 22

```
#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

#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 0
Pathname trunc option is 1
Disabled character for terminal files is 0.

Date 03-01-22

Expt. No. 20 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 %s <file> [<arg>]\n",
            argv[0]);
    return 0;
}

mkfifo(argv[1], S_IFIFO | S_IRWXU | S_IRWXG |
        S_IRWXO);
```

Teacher's Signature : _____

```
if (argc == 2){
    fd = open(argv[1], O_RDONLY|O_NONBLOCK);
    while(read(fd, buf, sizeof(buf))>0)
    printf("%s", buf);
}
else{
    fd = open(argv[1], O_WRONLY);
    write(fd, argv[2], strlen(argv[2]));
}
close(fd);
}
```

Teacher's Signature : _____

Output :

Terminal 1 : Write process

$ ./program20
USAGE ./program20 <file> [<arg>]

$ ./program20 fifo "Hello there!"
(waits for reader process)
(Returns terminal after reader process ends)

Terminal 2 : Reader process

$ ./program20 fifo
Hello there!