Date

Expt. No. 01

Page No. 1

Program - 01

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 "entered year is leap year"
elif [ $((leap%.4)) -eq 0 ]; then
echo "entered year is leap year"
elif [ $((leap%.100)) -ne 0 ] ; then
echo "entered year is not a leap year"
else
echo "entered year is leap year"
fi
```

Teacher's Signature : _____

Output

```
Enter the year
  2024
This year is a leap year

Enter the year
 2001
This year is not a leap year
```

Expt. No. 02 Date

Page No. 2

Program - 02

Shell Scoript to find the area of a circle

```
echo Radius:
read r
pi = 3.142
area = `echo $pi\*$r\*$r | bc`
echo Area of the circle:
echo $area
```

Teacher's Signature : _____

Output

Radius : 2

Area of the circle is 12.56

Expt. No. 03

Date

Page No. 3

Program - 03

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

```
echo Enter a Number:
read no
if [ $no -eq 0 ]
then
echo "The Number is 0"
elif [ $no -gt 0 ]
then
echo "The Number is Positive"
else
echo "The Number is Negative"
fi
```

Teacher's Signature : _____

Output:

Enter a Number
-50
The Number is Negative

Enter a Number
10
The Number is positive

Enter a Number
0
The Number is zero

Expt. No. 04 Date

Page No. 4

Program - 04

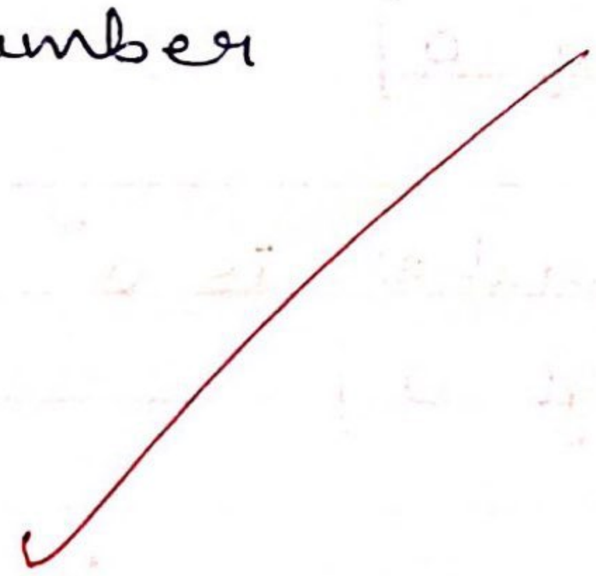
Shell script to find the biggest of three numbers

```
echo "enter the first number"
read a
echo "enter the second number"
read b
echo "enter the third number"
read c
if [ $a -gt $b -a $a -gt $c ]
then
echo "$a is biggest number"
elif [ $b -gt $a -a $b -gt $c ]
then
echo "$b is biggest number"
else
echo "$c is biggest number"
fi
```

Teacher's Signature : _____

Output :

```
Enter first number
 2
Enter second number
 3
Enter third number
 5
5 is biggest number
```



Date _____

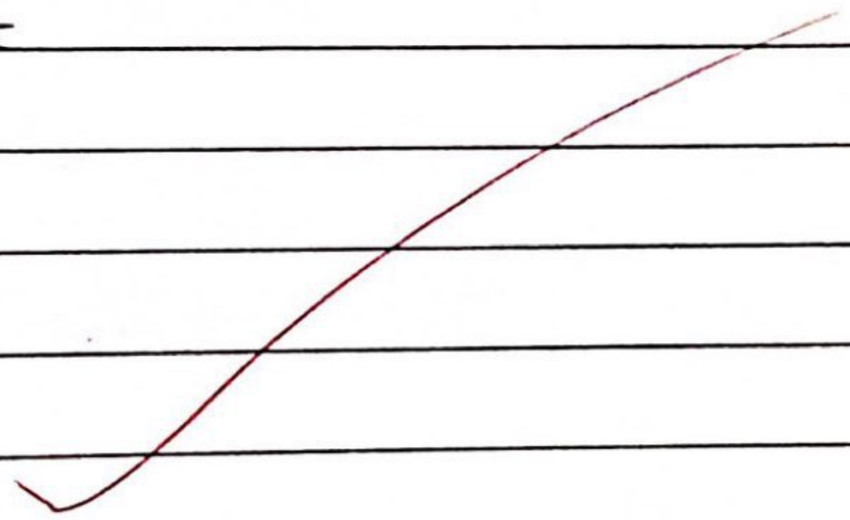
Expt. No. 05

Page No. 5

Program - 05

Shell script to find the factorial of a number

```
echo "enter a number"
read num
fact = 1
while [ $num -gt 1 ]
do
    fact = $((fact * num))      #fact = fact * num
    num = $((num - 1))          # num  = num - 1
done
echo $fact
```



Teacher's Signature : _____

Output:

Enter a number

3

→ 6

Date

Expt. No. 06 Page No. 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"
```

Teacher's Signature : _____

Output:

Enter the basic salary of employee
1000
Gross salary : 1300.0

Expt. No. 07

Date

Page No. 7

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"
```

Teacher's Signature : _____

Output:

Enter the temperature in frahrenheit
32
Temperature in celcius : 0

Expt. No. 08 Date

Page No. 8

Program - 08

Shell script to perform arithmetic operations on given two numbers

```
echo "enter two number:"
read a
read b
echo "enter choice:"
echo "1. Addition"
echo "2. Subtraction"
echo "3. Multiplication"
echo "4. Division"
read ch
```

Shell script to perform arithmetic operations on 2 no.

```
case $ch in
1) res = `echo $a + $b | bc`
;;
2) res = `echo $a - $b | bc`
;;
3) res = `echo $a \* $b | bc`
;;
4) res = `echo "scale=2; $a / $b" | bc`
;;
esac
echo "Result : $res"
```

Teacher's Signature : _____

Output:

```
Enter two numbers
  6
  3
enter choice
1. Addition
2. Subtraction
3. Multiplication
4. Division
 1
Result : 9
```

Expt. No. 09

Date

Page No. 9

Program - 09

shell script to find the sum of even numbers upto n

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

Teacher's Signature : _____

Output:

Enter the number
7
12

Expt. No. 10

Date

Page No. 10

Program - 10

Shell script to print the combinations of numbers 123

```
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 $j -ne $k -a $i -ne $k ]
then echo "$i$j$k"
fi
echo $i $j $k
done
done
done
```

Teacher's Signature : _____

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 3 2
3 3 3

Date _____

Expt. No. 11

Page No. 11

Program - 11

Shell script to find the power of a number

```
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"
```

Teacher's Signature : _____

Output:

Enter a number and power
2
6
Result is 64

Date _____

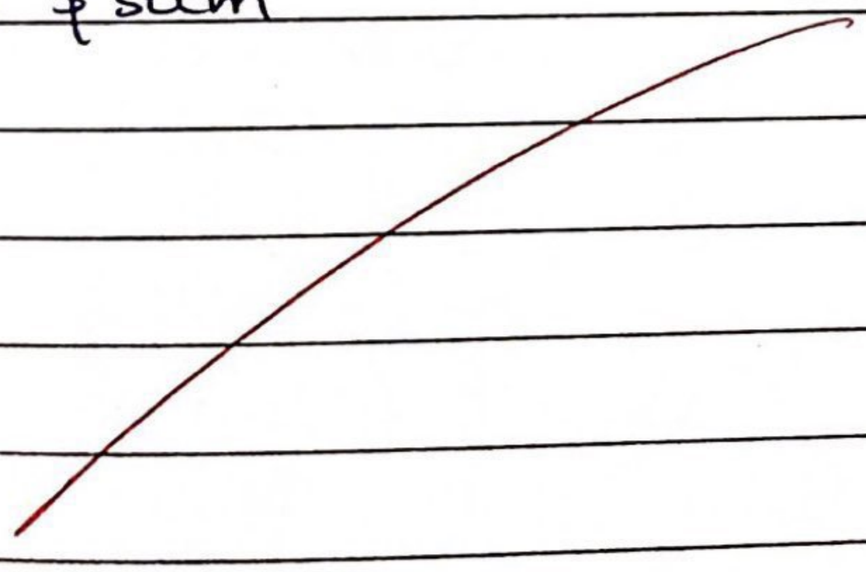
Expt. No. 12

Page No. 12

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 $N ]
do
 read num                    #get number
 sum =$((sum + num))  #sum+= num
 i=$((i+1))
done
echo $sum
```



Teacher's Signature : .

Output:

Enter size (N)
5
Enter numbers
3
8
55
2
1
69

Expt. No. 13 Date

Page No. 13

Program - 13

Shell script to display the pass class of a student

```
pass=0
fail=0
for((i=0; i<6; i++))
do
echo "Enter your cie marks out of 50"
read cie
echo "Enter your see marks out of 50"
read see
total=$((cie+see))

if [ $total -gt 90]
then echo "S grade"
pass=$((pass+1))
elif [ $total -gt 80]
then echo "A grade"
pass=$((pass+1))
elif [ $total -gt 70]
then echo "B grade"
pass=$((pass+1))
elif [ $total -gt 60]
then echo "C Grade"
pass=$((pass+1))
elif [ $total -gt 50]
then echo "D grade"
```

Teacher's Signature : _____

Date _____

Expt. No. _____ Page No. 14

```
pass=$((pass+1))
elif [$total -gt 40]
then echo "E grade"
pass=$((pass+1))
else
echo "Fail"
fail=$((fail+1))
fi
done
echo "Number of subjects passed: $pass"
echo "Number of subjects failed: $fail"
```

Teacher's Signature : _____

Output :

Enter your cie marks out of 50
23
Enter your see marks out of 50
23
E grade
Enter your cie marks out of 50
12
Enter your see marks out of 50
12
Fail
Enter your cie marks out of 50
45
Enter your see marks out of 50
35
B Grade

Date _____

Expt. No. 14

Page No. 15

Program-14

Shell script to find the Fibonacci series up to n

```
echo "Enter the value of n"
read n
a=0
b=1
for ((i=0;i<n;i++))
do
echo "$a"
fn=$((a+b))
a=$b
b=$fn
done
```

Teacher's Signature : _____

Output:

```
Enter the value of n
3
0
1
1
1

Enter the value of n
6
D
1
1
2
3
5
```

Date _____

Expt. No. 15

Page No. 4

Program - 15

Shell script to count the number of vowels of a string

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

Teacher's Signature : _____

Output:

```
Enter a string
asdfghjkl
1
Enter a string
aoive
5
Enter a string
Bhavana
3
```

Date _____

Expt. No. 16 Page No. 17

Program - 16

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

```
echo "Number of lines:"
cat progl5.sh | wc -l
echo "Number of words:"
cat progl5.sh | wc -w
echo "Number of characters:"
cat progl5.sh | wc -c
```

Teacher's Signature : _____

Output :

```
Number of lines:
15
Number of words:
55
Number of characters:
312
```

Date _____

Expt. No. 17

Page No. 18

Program-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 : _____

Output

```
SHELL = /bin/bash
SESSION_MANAGER = local/bmscecse-HP-Pro-3330-MT:@/tmp/
QT_ACCESSIBILITY=1
COLORTERM= truecolor
.
.
.
.
.
.
.
.
.
.
:
:
GDMSESSION = ubuntu
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus
_=./a.out
```

Date _____

Expt. No. 18 Page No. 19

Program - 18

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
```

Teacher's Signature : _____

Expt. No. _____ Date _____

Page No. 20

```
    printf (" Hard link created \n");
    }
    return 0;
    }
```

Teacher's Signature : _____

Output:

```
./a.out prog18.c z
Hard link created
ls -l
total 212
-rw-rw-r--   1 bmscecse  bmscecse   49 Oct 25 19156 91.
-rwxrwxrwx-x 1 bmscecse  bmscecse 16808 Jan 3 15:22 a.
    .
    .
    .
    .
-rw-rw-r--   8 bmscecse  bmscecse  514 Jan 3 15:22 z
./a.out prog18.c z
Cannot create hard link
./a.out prog18.c zz
Hard link created
./a.out -s prog18.c zz
Cannot create symbolic link
```

Date _____

Expt. No. 19 Page No. 21

Program-19

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 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_REST
                                    -RICTED);
#else
printf("system does not support chown-restricted
                                    option\n");
#endif
```

Teacher's Signature : _____

Date _____

Expt. No. _____ Page No. 22

```
#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:

```
$gcc prog19.c -o prog19
$./prog19
System supports Job Control
System supports saved set-UID and saved set-GID
chown_restricted option is 1
Pathname trunc option is 1
Disable characters for terminal files is 0
```

Date .....

Expt. No. 20

Page No. 27

Program-20

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);
 if(argc==2)
 {
   fd = open(argv[1], O_RDONLY | O_NONBLOCK);
   while(read(fd, buf, sizeof(buf))>0)
```

Teacher's Signature : _____

Date .....

Expt. No. .....

Page No. 24

```
        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

```
$gcc prog20.c -o prog20.
$./prog20
USAGE ./prog20 <File> [cargs]
$./prog20 pipeNam "Good Morning"
        [waits for reorder process]
$bmsce@bmsce:$                    [Ends after reader]
```

Terminal - 2 : Reorder Process

```
$./prog20 pipeNam
Good Morning bmsce@bmsce:$
```