

## LAB PROGRAM - 1

To find if the given year is leap or not.

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

```
echo "Enter a year : "
read year
```

```
if test $((year % 400)) -eq 0
then
```

```
    echo "$year is a Century Leap Year"
    if test $((year % 4)) -eq 0
    then
```

```
        if test $((year % 100)) -eq 0
        then
```

```
            echo "$year is NOT a Leap Year"
        else
```

```
            echo "$year is a Leap Year"
        fi
```

```
else
    echo "$year is NOT a Leap Year"
fi
```

Teacher's Signature :

OUTPUT :

Enter a year :

2016

2016 is a Leap Year

Enter a year :

1600

1600 is a Century Leap Year

Enter a year :

2007

2007 is NOT a Leap Year.

## LAB PROGRAM - 2

Shell script to find the area of a circle.

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

pi = 3.1415

echo "Enter the radius : "

read r

```
echo $pi * $r * $r | bc
```

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

10/10/26

19

Q

1 - main(11) M.

OUTPUT :

Enter the radius : 4.5

Enter the radius : 63.6151

4.5

63.6151

o pr. ((0 as raff)) is true

\* next get positive o if raff \* then  
o pr. ((1 as raff)) is true

o pr. ((2 as raff)) is true

\* next get o raff & raff \* then  
o pr. ((3 as raff)) is true

\* next get o if raff \* then  
o pr. ((4 as raff)) is true

\* next get o true is raff & raff

## Lab PROGRAM - 3

Shell script to check if number is zero or positive or negative.

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

```
echo "Enter a number :"  
Read n
```

```
if test $n -lt 0  
then
```

```
    echo "NEGATIVE number"
```

```
elif test $n -gt 0  
then
```

```
    echo "POSITIVE Number"
```

```
else
```

```
    echo "Number is ZERO"
```

```
fi
```

Teacher's Signature :

OUTPUT :

Enter a number :

45

POSITIVE Number

Enter a number :

0

Number is ZERO

Enter a number :

-19

~~Not~~ available for

NEGATIVE Number.

## LAB PROGRAM - 04

Shell script to find the largest of 3 numbers.

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

```
echo "Enter first number:"
```

```
read a
```

```
echo "Enter second number:"
```

```
read b
```

```
echo "Enter third number:"
```

```
read c
```

```
if test $a -ge $b
```

```
then
```

```
if test $a -ge $c
```

```
then
```

```
echo "$a is the largest"
```

```
else
```

```
echo "$c is the largest"
```

```
fi
```

```
else
```

```
if test $b -ge $c
```

```
then
```

```
echo "$b is the largest"
```

```
else
```

```
echo "$c is the largest"
```

```
fi
```

```
fi
```

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

OUTPUT.

Enter first number:

4

Enter second number:

7

Enter third number:

1

7 is the largest

Enter first number:

-1

Enter second number:

-4

Enter third number:

-1

-1 is the largest

## LAB PROGRAM - 5

Shell script to find the factorial of a number.

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

```
echo "Enter the value of n : "
```

```
read n
```

```
f=1;
```

```
i=1;
```

```
while test $i -le $n
```

```
do
```

```
    f=$((f * i))
```

```
    i=$((i + 1))
```

```
done
```

```
printf "Factorial of %d is %d\n" $n $f
```

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

OUTPUT :

Enter the value of n:

0

Factorial of 0 is 1

Enter the value of n:

5

Factorial of 5 is 120

## LAB PROGRAM - 6

Shell script to find the gross salary of an employee.

Given: DA = 10% of Basic

HRA = 20% of basic

Gross = Basic + DA + HRA.

#!/bin/bash

echo "Enter the basic salary:"  
read sal

if test \$sal -le 0  
then

echo "Salary should be greater than 0"  
else

a=0.1

b=0.2

da='echo \$sal \* \$a | bc'

hra='echo \$sal \* \$b | bc'

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

echo "DA is \$da"

echo "HRA is \$hra"

echo "Gross salary is \$gross"

fi

Teacher's Signature :

OUTPUT :

Enter the basic salary :

1000

DA is 100.0

HRA is 200.0

Gross salary is 1300.0

Enter the basic salary :

-555

Salary should be greater than 0

## LAB PROGRAM - 7

Shell script to convert temperature fahrenheit to Celsius.

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

```
echo "Enter the temperature in fahrenheit : "
read temp
```

```
cel='echo "scale=2; ($temp - 32) * (5/9)" | bc'
echo "The temperature in Celsius is $cel"
```

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

2. Input/Output

OUTPUT :

Enter the temperature in Fahrenheit : 32

The temperature in Celsius is 0

Enter the temperature in Fahrenheit :

0

The temperature in Celsius is -17.78

## LAB PROGRAM - 8

Shell script to perform arithmetic operations of 2 numbers.

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

echo "Enter the first number : "

read a

echo "Enter the second number : "

read b

sum=`echo \$a+\$b | bc`

diff=`echo \$a-\$b | bc`

pro=`echo \$a\times\\$b | bc`

quo=`echo \\$a/\\$b | bc`

mod=`echo \\$a \% \\$b | bc`

echo "Sum : \$sum\nDifference : \$diff\nProduct : \$pro\nQuotient : \$quo\nRemainder : \$mod"

Teacher's Signature :

## Program 16

OUTPUT :

Enter the first number :  
4

Enter the second number :  
5

Sum : 9

Difference : -1 [ "(num1)\*"(num2) ] is used " who "

Product : 20 [ " is used " who " is used " who "

Quotient : 0

Remainder : 4

## LAB PROGRAM - 09

Shell script to find the sum of even numbers upto n

#!/bin/sh

echo "Enter the value of n:"

read n

sum=0

i=0

while test \$i -le \$n

do

    sum=\$((sum+i))

    i=\$((i+1))

done

echo "Sum of even numbers upto \$n : \$sum"

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

OUTPUT:

Enter the value of n:

10

Sum of even numbers upto 10 : 30

Enter the value of n:

0

Sum of even numbers upto 0 : 0

## LAB PROGRAM - 10

Shell script to print the combinations of numbers 123

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

```
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
```

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

## OUTPUT:

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

## LAB PROGRAM - 11

Shell script to find the power of a number.

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

```
echo "Enter the base:"
```

```
read b
```

```
echo "Enter the power:"
```

```
read p
```

```
prod=1
```

```
i=1
```

```
while test $i -le $p
```

```
do
```

```
prod=$((prod * b))
```

```
i=$((i+1))
```

```
done
```

```
echo "$b to the power $p is : $prod"
```

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

OUTPUT :

Enter the base:

5

Enter the power:

0

5 to the power 0 is : 1

Enter the base:

8

Enter the power:

3

8 to the power 3 is : 512

## LAB PROGRAM - 12

Shell script to find the sum of n natural numbers

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

```
echo "Enter the value of n : "
read n
```

```
if test $n -le 0
```

```
then
```

```
echo "Value of n should be greater than 0 !"
else
```

```
num=0
```

```
i=1
```

```
while test $i -le $n
```

```
do
```

```
num=$((num+i))
```

```
i=$((i+1))
```

```
done
```

```
echo "Sum of $n natural number is : $num"
```

```
fi
```

Teacher's Signature :

OUTPUT:

11 - maxwell 47

Enter the value of n:  
5  
Sum of 5 natural numbers is : 15

Enter the value of n:  
-8

Value of n should be greater than 0!

11 - maxwell 47

11 - maxwell 47

## LAB PROGRAM - 13

Shell script to display the pass class of a student. (Take n=6)

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

```
i=1
```

```
pass=0
```

```
fail=0
```

```
while test $i -le 6
```

```
do
```

```
echo "Enter the CIE marks out of 50 for subject $i"
```

```
read $cie
```

```
echo "Enter the SEE marks out of 100 for subject $i"
```

```
read $see
```

```
see=$((see/2))
```

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

```
if test $tot -ge 90
```

```
then
```

```
echo "S grade"
```

```
pass=$((pass+1))
```

```
elif test $tot -ge 80
```

```
then
```

```
echo "A grade"
```

```
pass=$((pass+1))
```

```
elif test $tot -ge 70
```

```
then
```

```
echo "B grade"
```

```
pass=$((pass+1))
```

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

If but \$tot -ge 60  
then

echo "C grade"

$$\text{pass} = \$((\text{pass} + 1))$$

If but \$tot -ge 50  
then

echo "D grade"

$$\text{pass} = \$((\text{pass} + 1))$$

If but \$tot -ge 40  
then

echo "E grade"

$$\text{pass} = \$((\text{pass} + 1))$$

else

echo "F grade"

$$\text{fail} = \$((\text{fail} + 1))$$

fi

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

done

echo "Number of subjects passed = \$pass"

echo "Number 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

20

S grade

Enter the CIE marks out of 50 for subject 2

48

Enter the SEE marks out of 100 for subject 2

88

S grade

Enter the CIE marks out of 50 for subject 3

30

Enter the SEE marks out of 100 for subject 3

18

F grade

Enter the CIE marks out of 50 for subject 4

20

Enter the SEE marks out of 100 for subject 4

360

D grade

Enter the CIE marks out of 50 for subject 5

46

Enter the SEE marks out of 100 for subject 5

98

S grade

Enter the CIE marks out of 50 for subject 6

15

Enter the SEE marks out of 100 for subject 6

48

F grade

Number of subjects passed = 4

Number of subjects failed = 2.

## LAB PROGRAM - 14

Shell script to find the fibonacci series upto n (n terms).

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

echo "Enter the value of n : "

read n

a=0

b=1

if test \$n -eq 1

then

echo "fibonacci series of \$n terms is : \n0"

elif test \$n -eq 2

then

echo "fibonacci series of \$n terms is : \n0\n1"

else

i=3

echo "fibonacci series of \$n terms : \n\\$a\n\\$b"

while test \$i -le \$n

do

c=`echo \$a + \$b | bc`

echo "\$c"

a=\$b

b=\$c

i=\$((i+1))

done

fi

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

OUTPUT:

Enter the value of n:

5

fibonacci series of 5 terms:

0

1

1

2

3

Enter the value of n:

1

fibonacci series of 1 term:

0

## LAB PROGRAM - 15

Shell script to find the number of vowels of a string

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

```
echo "Enter a string:"
```

```
read s
```

```
len=`expr "$s" | wc -c`
```

```
len=${len%1}
```

```
ctr=0
```

```
while test $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:

Enter a string:  
Adarsh

Number of vowels is 2

Enter a string:

123456

Number of vowels is 0

## LAB PROGRAM - 16

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

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

echo "Enter the absolute path of the file :"  
read path

```
wctr = `wc -w < $path`
```

```
lctr = `wc -l < $path`
```

```
cctr = `wc -c < $path`
```

echo "Number of lines : \$lctr"

echo "Number of words : \$wctr"

echo "Number of characters : \$cctr"

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

OUTPUT :

Enter the absolute path of the file :  
\home\text.txt

Number of lines : 5

Number of words : 30

Number of characters : 101

## LAB 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;
    int **ptr;
    extern char **environ;
    for (ptr = environ; *ptr != 0; ptr++)
        printf ("%s\n", *ptr);
    return 0;
```

3

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

## OUTPUT:

SSH-AGENT-PID = 3207  
HOSTNAME = localhost.localdomain.  
DESKTOP-STARTUP-ID =  
SHELL = /bin/bash.  
TERM = xterm  
HISTSIZE = 1000  
KDE-NODE-IPV6 = 1.  
GTK\_RC\_FILES = /etc/gtk/gtkrc; /root/.gtkrc-1.2-gnome2.  
WINDOWID = 44040273  
OLDPWD = /root/tan.  
QTDIR = /usr/lib/qt-3.3  
QTINC = /usr/lib/qt-3.3/include  
USER = root  
LS-COLORS = no=00:fi=00:di=00;34:so=00;34:  
GNOME-KEYRING-SOCKET = /tmp/keyring-vsDBVL/socket.  
KDEDIR = /usr  
DESKTOP SESSION = default  
GDM-XSERVER-LOCATION = local  
INPUTRC = /etc/inputrc  
PWD = /root/tan/mpl  
XMODIFIERS = @im=none  
LANG = en-US.UTF-8  
GDMSESSION = default  
HOME = /root  
QTLIB = /usr/lib/qt-3.3/lib.

## LAB PROGRAM - 18

Write a C/C++ program to emulate the Linux 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[3], "-s") != 0))
        printf("Usage : ./a.out [-s] <org-file> <new-link> \n");
    return 1;
}

if (argc == 4)
{
    if ((symlink(argv[2], argv[3])) == -1)
        printf("Cannot make symbolic link \n");
    else
        printf("Symbolic link created \n");
}
else
{
    if ((link(argv[1], argv[2])) == -1)
        printf("Cannot make hard link \n");
}
```

Teacher's Signature :

Date 6/12/21

Expt. No. 18

Page No. 20

Ques

printf ("Headlink created\n");

return 0;

}

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

## OUTPUT:

```
for all .o files in current work dir create hard link  
.a.out 1 2 3 4  
Usage: ./a.out [-s] <org-file> <new-link>  
./a.out l.c => Hard link created  
./a.out l.c => Cannot create hard link (Because > already exists)  
.a.out -s l.c => symbolic link created.
```

## LAB 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 <stropts.h>
#include <unistd.h>
int main()
{
    #if defined _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-GID\n");
    #else
        printf("System does not support saved set-UID and
              saved-GID\n");
    #endif

    #ifdef _POSIX_CHOWN_RESTRICTED
        printf("Chown restricted option is %d\n", _POSIX_CHOWN_RESTRICTED);
    #endif
}
```

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

```
# elm
```

```
    printf ("System does not support chosen 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 ("Vdisnable 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 1  
Pathname Trunc option is 1  
Disable character for terminal files is 0.

## LAB PROGRAM - 20

Write a C/C++ program which demonstrates interprocess communication between a reader process and a writer process like mkfifo, open, read, write and close API's in your program.

```
#include <sys/types.h>
#include <unistd.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <istring.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_IWUSR | S_IRWXG | S_IRWXO);
    if (argc == 2)
    {
        fd = open(argv[1], O_RDONLY | O_NONBLOCK);
    }
```

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

```
while (read(fd, buf, sizeof(buf)) > 0)
```

{

else

{

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

```
    write(fd, argv[2], strlen(argv[2]));
```

```
} close(fd);
```

}

Teacher's Signature :

OUTPUT:

Terminal - 1.

.a.out FIFO "This is the Record of Adarsh.K.N"

Terminal - 2

.a.out FIFO

This is the Record of Adarsh.K.N.