

Program :- 1

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 "The year entered is a leap year"

elif [\$(leap % 4) -eq 0]

then echo "The year entered is a leap year"

elif [\$(leap % 100) -ne 0]

then echo "The year entered is not a leap year"

else

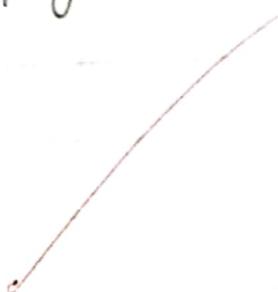
echo "The year entered is not a leap year"

fi

Output :- Enter the Year

2000

Enter year is a leap year



Program :- 2

Shell Script to find the area of a circle.

#!/bin/bash

echo Radius:

read r

pi = 3.142

echo = Area of the Circle:

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

echo \$ area

Output :- Radius

4.5

Area of the circle

68.625

Program:-3

Shell Script to check whether the no. zero / Positive or Negative.

echo enter a number

read num

if [\$num -gt 0]; then

echo "Positive"

elif [\$num -lt 0]; then

echo "Negative"

else

echo "Zero"

fi

Output :- Enter a number

-50

Negative

Program :- 4

Shell Script to find the biggest of three nos.

echo "Enter the first number"

read n1

echo "Enter the Second number"

read n2

echo "Enter the Third number"

read n3

if [\$n1 -gt \$n2 -a \$n1 -gt \$n3]

then echo "\$n1 is the greatest"

elif [\$n2 -gt \$n1 -a \$n2 -gt \$n3]

then echo "\$n2 is the greatest"

else

echo "\$n3 is the greatest"

fi

Output : Enter the 1st Number

2

Enter the 2nd Number

3

Enter the 3rd Number

5

5 is the biggest number

Program :- 5

Shell Script to find the factorial of a number

```
echo "Enter a number"
```

```
read num
```

```
fact = 1
```

```
for ((i = 2; i < num; i++))
```

```
do
```

```
fact = $((fact * i))
```

```
done
```

```
echo $fact
```

Output :- Enter a Number

3

6

Date 8/11/21

Expt. No. 6

Page No. 6

Program:- 6

Shell Script to compute the gross salary of an employee.

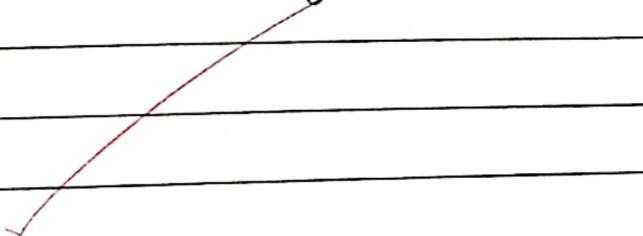
echo "Enter the basic salary of the employee"
read basic

da='echo 0.1 * \$ basic | bc'

hra='echo 0.2 * \$ basic | bc'

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

echo "Gross salary : \$ gross"



Output : Enter the basic salary of the employee

1000

Gross salary : 1300.0

Program :- 7

Shell Script to Convert the Temperature Fahrenheit to Celsius.

echo "Enter the Temperature in fahrenheit"

read temp

t = `echo "scale=4; \$temp - 32) * 5/9;" | bc`

cld = `echo \$t`

echo "Temperature in celsius : \$cld"

Output : Enter the temperature in Fahrenheit

25

Temperature in celsius = -3.8885

Program :- 8

Shell script to perform arithmetic operation on given two nos.

```
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
```

```
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'~~~~;;~~~~5) res=~~

```
echo "Result : $ res"
```

Teacher's Signature :

Output : Enter Two Numbers

2

4

Enter Choice:

1. Addition
2. Subtraction
3. Multiplication
4. Division.

1

Result : 6

Program :- 9

Shell Script to find the sum of even no. upto n

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

Output : Enter the Number

7

12

Program :- 10

Shell Script to print the combination of num 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

done

done

done

Output :- 1 1 1

1 1 2

1 1 3

1 2 1

1 2 2

1 2 3

1 2 3

1 3 1

1 3 2

1 3 2

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 1 2

3 1 3

3 2 1

3 2 1

3 2 2

3 2 3

3 3 1

3 3 2

3 3 3

Program:-11

Shell Script to find the power of a no.

#!/bin/sh

echo "Enter a number and power"

read a

read b

p=\$b

res=1

while [\$b -gt 0]

do

res=\$(echo \$res * \$a" 1bc")

b=\$(echo \$b-1" 1bc")

done

echo "Result is \$res"



Output : Enter a Number and Power

2

3

Result is 8

Program :- 12

Shell Script to find the sum of n natural nos.

who "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

Output:- Enter Size (N)

5

Enter Number

10

20

30

40

50

150

Program :- 13

Shell script to display the pass class of a student

pass = 0

fail = 0

for ((i=0; i<6; i+1))

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 "\$ 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"

pass = \$((pass + 1))

```
if [ $total -gt 10 ]  
then echo "Grade"  
pass=$(( pass+1 ))  
else echo "Fail"  
fail=$(( fail+1 ))  
fi
```

done

```
echo "No of subjects passed : $pass"  
echo "No of subjects failed = $fail"
```

Enter your ai marks out of 50

23

Enter your sc marks out of 50

23

E Grade

Enter your ai marks out of 50

12

Enter your sc marks out of 50

12

Fail

Enter your ai marks out of 50

34

Enter your sc marks out of 50

45

B Grade

Number of Subjects passed : 2

Number of Subject failed : 1

Program 14:-

Shell Script to find the Fibonacci Series up to a

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

Enter the value of n

6

0

1

1

2

3

5

Program 15

Shell script to count the no. 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" = "u"-o "$c" = "o"`
then
    count = $((count + 1))
fi
string = `expr "$string" : '^\([ -\]'`"
```

done

```
echo "The number of vowels: $count"
```

Enter a String

Hello

The number of vowels : 2

Program 16

Shell script to check no. of lines, words, character in a file.

echo "Number of lines":

cat prog15.sh | wc -l

echo "Number of words":

cat prog15.sh | wc -w

echo "Number of characters":

cat prog15.sh | wc -c

Number of lines

15

Number of words

55

Number of characters:

312

Program 17:-

Write a C/C++ program to that output 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 :

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
OLDDWD=/root/tan
QTDIR=/root/lib/qt-3.3
QTINC=/root/lib/qt-3.3/include
USER=root
LS_COLORS=no=00;fi=00:di=00;34:
GNOME_KEYRING_SOCKET=/tmp/keyring-wsDBVL/socket
SSH_AUTH_SOCK=/tmp/ssh-SEWJH3149/agent.3149
KDEDIR=/usr
SESSION_MANAGER=local/localhost.localdomain:/tmp/
ICE-unix/1349
MAIL=/var/spool/mail/root
DESKTOP_SESSION=default
PATH=/usr/lib/qt-3.3/bin:/usr/kde3/bin:/usr/
kerberos/bin:/usr/local/bin:/usr/kde3/bin:/usr/
kerberos/bin:
GDM_XSERVER_LOCATION=local
INPUTRC=/etc/inputrc
PWD=/root/tan/usr
XMODIFIERS=@im=none

KDE_IS_PRELINKED=1

LANG=en-US.UTF-8

GDMSESSION=default

SSH_ASKPASS=/usr/libexec/openssh/gnome-ssh-askpass

HOME=/root

DISPLAY=:0.0

G_BROKEN_FILERAMES=1

COLORTERM=gnome-terminal

XAUTHORITY=/tmp/.gdm5X71UW

=/root.

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], "d") == 0))
    {
        printf ("Usage : ./a.out [s] <org_file>\n"
               "<new_link> \n");
        exit (-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");
    }
}
```

Teacher's Signature :

```
else  
    print ("Hard link created in");  
}  
return 0;  
}
```

• ./a.out[-s] <arg_file> <new_link>

• ./a.out 1234

• ./a.out[-s] <arg_file> <new>

• ./a.out l.cz

Hard link created

ls -l

-rwxr--r-- 2 root 657 Mar 27 14:22 la.c

-rwxr--r- 2 root 657 Mar 27 14:22 z

• ./a.out la.cz

Cannot create hard link

• ./a.out -S la.cz z

Symbolic link created

ls -l

-rwxr--r-- 2 root root 657 Mar 27 14:22 la.c

lrwxrwxrwx 1 root root 4 Apr 1 18:32 z -> la.c

readlink z

la.c

Program 19:-

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

```
#define _POSIX_SOURCE
#define _POSIX_C_SOURCE 199309L
#define <stdio.h>
#define <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
    #if defined _POSIX_SAVED_IDS
    printf("System supports saved set-CID and set-GID\n");
    #else
    printf("System does not support saved set-CID and set-GID\n");
    #endif
    #if defined _POSIX_CHOWN_RESTRICTED
    printf("chown RESTRICTED");
    #else
    printf("System does not support chown restricted
option\n");
    #endif
    #if defined _POSIX_NO_TRUNC
```

Teacher's Signature :

```
printf("Pathname trim option is %d\n", POSIX_NO_TRUNC);
#else
printf("System does not support system wide
pathname trim option\n");
#endif
#endif if d != POSIX_DISABLE
printf("Disable character for terminal files is %d\n")
#else
printf("System does not support POSIX_DISABLE\n");
#endif
return 0;
}
```

System supports job control

System supports saved set - UID and saved set - GID
chown - restricted option is 1

Pathname trim option is 1

Disable character for terminal files is 0

Program 20:-

Write a C/C++ program which demonstrates Interprocess Communication between a reader process and a writer process. Use `mknod`, `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 <stropts.h>

int main (int argc, char *argv[])
{
    int fd;
    char buf[256];
    if (argc != 2 && argc != 3)
    {
        printf("USAGE: %s <file> [arg1]\n", argv[0]);
        return 0;
    }
    mknod(argv[1], S_IFIFO | S_IRWXV | S_IWUSR | S_IRWXU);
    if (argc == 2)
    {
        fd = open(argv[1], O_RDONLY | O_NONBLOCK);
        while (read(fd, buf, sizeof(buf)) > 0)
```

Teacher's Signature : _____

```
printf("%d", luf);
```

```
else
```

```
{
```

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

```
writen(fd, arg[2], strlen(arg[2]));
```

```
}
```

```
close(fd);
```

```
}
```

Terminal 1

- /a.out FIFO "This is USP lab"

Terminal 2

- /a.out FIFO 1

This is USP lab.