

1) shell script to find given number is leap or not

⇒ #!/bin/bash
echo "Enter the year"
read y
if [\$((\$a % 4)) -eq 0 -a \$((\$b % 100)) -ne 0 -o -a \$((\$c % 400)) -eq 0]

then

echo "\$y is a leap year"

else

echo "\$y is not a leap year"

fi

Teacher's Signature :

Output (1)

- Enter the year

2024

2024 is a leap year

- Enter the year

3035

3035 is not a leap year

2) Shell script to find area of circle

=> `#!/bin/bash`

`echo "Enter the radius of circle : "`
`read r`

`echo "Area of circle is : "`

`echo " 3.14 * $r * $r " | bc`

3) Shell script to check whether number is (+ve), (-ve) or zero

=> `#!/bin/bash`

`echo "Enter a number "`

`read n`

`if [$n -gt 0]`

`then`

`echo "It is positive "`

`elif [$n -lt 0]`

`then`

`echo "It is negative "`

`else`

`echo "It is zero "`

`fi`

Teacher's Signature :

Output (2)

Enter the radius of circle:

3

Area of circle is:

28.26

Output (3)

- Enter a number:

3

It is positive

- Enter a number

-2

It is negative

- Enter a number

0

It is zero

④ shell script to find biggest of 3 numbers

```
#!/bin/sh
echo "Enter 3 numbers"
read a b c
if [ $a -gt $b ]
then
    if [ $a -gt $c ]
    then
        echo "$a is maximum"
    elif [ $c -gt $a ]
    then
        echo "$c is maximum"
    fi
else
    if [ $c -gt $b ]
    then
        echo "$c is maximum"
    else
        echo "$b is maximum"
    fi
fi
```

Output(4)

- Enter 3 numbers:

3 1 4

4 is maximum

5) Shell script to find factorial of a number

```
→ #!/bin/bash  
echo "Enter a number"  
read num  
fact = 1  
while [ $num -gt 1 ]  
do  
    fact=$((fact * num))  
    num=$((num - 1))  
done  
echo "The factorial is : "  
echo $fact
```

Teacher's Signature :

Output(s)

Enter a number

5

The factorial is

120

6) Shell script to find gross salary of an employee

=> echo "Enter the basic salary"

read bs

gross = \$(((bs + (bs/100)*20 + (bs/100)*10))

echo "gross salary is : "

echo \$gross

Teacher's Signature :

output(6)

Enter the basic salary:

25000

gross salary is: ➤

32.500

⇒ Shell script to convert the temperature Fahrenheit to Celsius

=> #!/bin/bash

```
echo "Enter temperature (in F)"  
read temp
```

c = \$(((temp - 32) * 5 / 9))

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

Output (?)

Enter the temperature (in F) :

74

The temperature in celsius is:

23.3

8) Shell script to perform arithmetic operation on given 2 numbers

```
#!/bin/sh
echo "Enter 2 numbers"
read a
read b
echo "Enter the operation to be performed (+,-,*,/)"
read ch
case $ch in
    '+') echo "Sum : `expr $a + $b`"
    ;;
    '-') echo "Difference : `expr $a - $b`"
    ;;
    '*') echo "Product : `expr $a * $b`"
    ;;
    '/') echo "Division : `expr $a / $b`"
    ;;
    *) echo "Invalid input"
    ;;
esac
```

Output (s)

Enter 2 numbers:

20

10

Enter operation to be performed (+, -, *, /)

Case (1) +

Sum : 30

Case (2) -

~~Subtract~~ 10

Difference : 10

Case (3) *

Product : 200

Case (4) /

Division : 2

q) Shell script to find the sum of even numbers upto n

```
#!/bin/sh  
echo "Enter the upper limit:"  
read n  
i=0  
while [ $i -le $n ]  
do  
    sum=$((sum + i))  
    i=$((i+2))
```

done

echo "The sum of even numbers upto \$n is: \$sum"

Teacher's Signature : _____

output(9)

Enter the upper limit:

6

The sum of even numbers upto 6 is: 12

10) Shell script to print the combination of numbers 1 2 3

```
#!/bin/bash
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 (10)

1 1 1

1 1 2

1 1 3

2 2 1

1 2 2

1 2 3

1 3 1

1 3 2

1 3 3

2 1 1

2 1 2

2 1 3

2 2 1

2 2 2

2 2 3

2 3 1

2 3 2

2 3 3

3 1 1

3 1 2

3 1 3

3 2 1

3 2 2

3 2 3

3 3 1

3 3 2

3 3 3

i) shell script to find power of a number

```
=> #!/bin/sh  
echo "Enter the number : "  
read n  
echo "Enter the power : "  
read p  
pow=$p  
ans=1  
while [ $p -ne 0 ]  
do  
    ans=$((ans * n))  
    p=$((p - 1))  
done  
echo "$n to the root power $pow is : "  
echo $ans
```

Teacher's Signature : _____

Output (1)

Enter the number

2

Ents the power

4

So 2 to the power 4 is

16

Ques) Shell script to find sum of n natural numbers

Ans) #!/bin/bash

echo "Enter value of n"

read n

i=1

while [\$i -le \$n]

do

sum=\$((sum+i))

i=\$((i+1))

done

echo "Sum of first \$n natural numbers is : \$sum"

Teacher's Signature :

Output (12)

Enter value of n:

6

Sum of first 6 natural numbers is 21

13) Shell script to display pass class of student

!/bin/sh

n=0

pass=0

while [\$n -lt 6]

do

echo " Enter cie marks of \$((n+1)) subject : "

read cie

echo " Enter the re marks of \$((n+1)) subject : "

read ree

marks=\$((cie + ree))

if [\$cie -gt 20]

then

if [\$ree -ge 40]

then

pass=\$((pass + 1))

if [\$marks -gt 90]

then

echo " The grade is S "

elif [\$marks -ge 80 -a \$marks < 90]

then

echo " The grade is A "

elif [\$marks -ge 70 -a \$marks < 80]

then

echo " the grade is B "

Teacher's Signature : _____

if [\$marks -ge 60 -a \$marks -lt 70]
then

echo "The grade is C"

elif [\$marks -ge 50 -a \$marks -lt 60]
echo "The grade is D"

else

echo "The grade is F"

fi

else

echo "The grade is F"

fi

else

echo "The grade is F"

fi

n= \$((n+1))

done

echo "The ~~total~~ number of subject passed is : \$pan"

Teacher's Signature : _____

Output

Enter cie marks of 1 subject:

98

Enter re marks of 1 subject

64

The grade is C

Enter cie marks of 2 subject

19

The grade is F

Enter cie marks of 3 subject

44

Enter re marks of 3 subject

38

The grade is F

Enter cie marks of 4 subject

45

Enter re marks of 4 subject

90

The grade is S

Enter cie marks of 5 subject

40

Enter re marks of 5 subject

60

The grade is B

Enter cie marks of 6 subject

36

Enter six marks of 6 subjects

88

The grade is A

The number of subject passed is 4.

ii) Shell script to find fibonacci series upto n

```
#!/bin/sh
echo "Enter number of terms : "
read n
x=0
y=1
i=2
echo "Fibonacci series upto n term is : "
echo "$x"
echo "$y"
while [ $i -lt $n ]
do
    z=$((x+y))
    echo "$z"
    x=$y
    y=$z
done
```

Teacher's Signature : _____

Output

Enter number of terms:

5

Fibonacci series upto 5 terms is:

0 1 1 2 3

(15) Shell script to find number of vowels of string.

=> #!/bin/bash

echo "Enter the string"

read str

len= \${#str} # length of str

count= 0

while [\$len -gt 0]

do

ch= \${str:0:1} # cut -c len

case \$ch in

[aeiouAEIOU])

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

;

done

len= \${len:1}

done

echo "The number of vowel is : \$count"

Output

Enter the string:

DhrubaabCE

The number of vowel is 4

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

>

#!/bin/bash

echo "Enter the file name"

read f

l='wc -l \$f'

c='wc -c \$f'

w='wc -w \$f'

echo "Lines = \$l \n Words = \$w \n Characters = \$c"

Teacher's Signature : _____

Output

cat abc.txt

Hii Hello

My name is Dhruva.

Enters fib name:

abc.txt

Lines = 2

Words = 6

Characters = 21

17) Write a C/C++ program that outputs the contents of the 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 : _____

18) Write a c/c++ program to emulate the 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-fil> <new link>"), .
        return 1;
    }
    else if (argc==4) {
        if (symlink(argv[2], argv[3]) == -1) {
            printf ("cannot make symbolic link"), .
        }
        else {
            printf ("Symbolic link created"), .
        }
    }
    else {
        if (link(argv[1], argv[2]) == -1) {
            printf ("cannot make hard link"), .
        }
        else {
            printf ("Hard link created"), .
        }
    }
    return 0;
}

```

Teacher's Signature : _____

Activities Terminal Jan 21 19:08
usp@usp:~\$ gcc link.c
usp@usp:~\$./a.out ex.c ac
Hard link created
usp@usp:~\$ ls -l ex.c ac
-rw-rw-r-- 3 usp usp 63 Jan 10 15:13 ac
-rw-rw-r-- 3 usp usp 63 Jan 10 15:13 ex.c
usp@usp:~\$./a.out -s ex.c ad
Symbolic link created
usp@usp:~\$ ls -l ad
lrwxrwxrwx 1 usp usp 4 Jan 21 19:08 ad -> ex.c
usp@usp:~\$

19) C or C++ program to find the configuration supported using 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 SET-GID\n");
    #else
        printf("It does not support SET-UID and SET-GID\n");
    #endif
    #ifdef _POSIX_CHOWN_RESTRICTED
        printf("System provides CHOWN RESTRICTED %d\n",
               _POSIX_CHOWN_RESTRICTED);
    #else
        printf("System does not support CHOWN options.\n");
    #endif
}
```

Teacher's Signature : _____

ifdef - POSIX_NO_TRUNC

printf ("Path name trunc option is %d\n")

else

printf ("Path name trunc option is not supported\n"),

endif

ifdef - POSIX_VDISABLE

printf ("Dual character is %d\n",

else

POSIX_VDISABLE),

printf ("Dual character is not supported\n"),

endif

}

Activities Terminal ▾

```
usp@usp:~$ gcc con  
config.c    contents.c  
usp@usp:~$ gcc config.c  
usp@usp:~$ ./a.out  
System supports job control  
System supports saved set-UID and saved set-GID  
chown_restricted option is 0  
Pathname trunc option is 1  
Disable character for terminal files is 0  
usp@usp:~$
```

20) Write a C/C++ program which demonstrates interprocess communication b/w reader process and writer process

```

#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 * args[7] )
{
    int fd,
        char buf[256],
        if (argc != 2 && argc != 3) {
            printf (" USAGE %s <file> [<args>]\n", args[0]);
            return 0;
        }
        mkfifo ( args[1], S_IFIFO | S_IRWXU | S_IWUSR | S_IRWXO );
        if (argc == 2)
        {
            fd = open ( args[1], O_RDONLY | O_NONBLOCK );
            while (read ( fd, buf, sizeof (buf) ) > 0)
                printf ("%s", buf);
        }
        else
    }

```

Date _____

Expt. No. _____

Page No. _____

fd = open (argv[1], O_WRONLY),
write (fd, argv[2], strlen (argv[2])),
close (fd);
}

Teacher's Signature : _____

Activities

Terminal ▾

```
usp@usp:~$ gcc inter_co.c
usp@usp:~$ ./a.out go
HI govinda
usp@usp:~$
```

