1. Write a shell script to perform integer arithmetic operations.
echo "Enter the number "
read a
echo "Enter the number"
read b
$\mathrm{c}={ }^{`}$ expr $\$ \mathrm{a}+\$ \mathrm{~b}$ '
echo "Addition= \$c"
d=`expr \$a - \$b`
echo "subtraction=\$d"
$\mathrm{e}=` \operatorname{expr} \$ \mathrm{a} \backslash^{*} \$ \mathrm{~b}^{`}$
echo "Multipliccation=\$e"
$\mathrm{f}=`$ expr \$a / \$b`
echo "Division=\$f"

## OUTPUT:

> Enter the number 4
Enter the number 2
Addition=6
subtraction=2
Multipliccation=8
Division=2
B) Write a shell script to perform floating point arithmetic operations.
echo "Enter the number "
read a
echo "Enter the number"
read b
$\mathrm{c}=`$ echo $\$ \mathrm{a}+\$ \mathrm{~b} \mid \mathrm{bc}{ }^{\prime}$ echo "Addition= \$c" $\mathrm{d}=`$ echo $\$ \mathrm{a}-\mathrm{\$ b} \mid \mathrm{bc}$ echo "subtraction=\$d" $\mathrm{e}=`$ echo $\$ \mathrm{a} \backslash^{*} \$ \mathrm{~b} \mid \mathrm{bc}{ }^{`}$
echo "Multipliccation=\$e"
$\mathrm{f}=`$ echo $\$ \mathrm{a} / \mathrm{\$ b} \mid \mathrm{bc}$ echo "Division=\$f"

## OUTPUT:

Enter the number 4.2

Enter the number 2.1

Addition=6.3
subtraction=2.1
Multipliccation $=8.82$
Division=2
2. Write a shell script to display first $\mathbf{1 0}$ natural numbers.

```
i=1
while[ $i -le 10]
do
echo $i
i=`expr $i+1`
done
```


## OUTPUT:

1
2
3
4
5
6 7
8
9
10
3. Write a shell script to find out the factorial of the given number.

```
i=1
f=1
echo " Enter the number"
read n
while[ $i -le $n]
do
f=`expr $f \* $i`
i=`expr $i + 1`
done
echo FACTORIAL = $f
```


## OUTPUT:

$$
\begin{aligned}
& \text { Enter the number } \\
& 5 \\
& \text { FACTORIAL }=120
\end{aligned}
$$

4. Write a shell script to find out whether the given number is prime number or not.
echo enter the number
read $n$
$\mathrm{i}=1$
$\mathrm{c}=0$
while [ \$n-ge \$i ]
```
do
    if [`expr $n % $i` -eq 0 ]
then
c=`expr $c + 1`
fi
i=`expr $i + 1`
done
if [ $c -eq 2 ]
then
echo prime
else
echo not prime
fi
```


5. write a shell script to find out sum of the digits of the given number.
sum=0
echo " Enter the number"
read $n$
while [ $\$ \mathrm{n}$-gt 0 ]
do

```
temp=`expr $n % 10`
sum=`expr $sum + $temp`
n=`expr $n / 10`
done
echo $sum
```


## OUTPUT

Enter the number
123
6
6. Write a shell script that takes a command line argument and reports on whether it is directory, a file, or something else.

```
    if [-f \$1]
then
    echo "it is a file"
    elif [-d \$1]
    then
        echo it is a directory
            else
        echo "something else"
    fi
OUTPUT:
```

\$sh file1 abc
abc is a file

1. AIM: Write shell script that will add two nos, which are supplied as command line argument, and if this two nos are not given show error and its usage
if [ \$\# -ne 2 ]
then
echo "Usage - \$0 x y"
echo " Where x and y are two nos for which I will print sum"
exit 1
fi
echo "Sum of \$1 and \$2 is `expr \$1+\$2`"
OUTPUT:
Shell.sh 122
14
2. AIM: Write Shell script to find out biggest number from given three nos. Numbers are supplied as command line arguments. Print error if sufficient arguments are not supplied.
```
    if [ $# -ne 3 ]
    then
            echo "$0: number1 number2 number3 are not given" >&2
        exit 1
    fi
    n1=$1
    n2=$2
    n3=$3
    if [ $n1 -gt $n2 ] && [ $n1 -gt $n3 ]
    then
echo "$n1 is Bigest number"
    elif [ $n2 -gt $n1 ] && [ $n2 -gt $n3 ]
    then
        echo "$n2 is Bigest number"
    elif [ $n3 -gt $n1 ] && [ $n3 -gt $n2 ]
    then
        echo "$n3 is Bigest number"
    elif [ $n1 -eq $n2 ] && [ $n1 -eq $n3 ] && [ $n2 -eq $n3 ]
    then
        echo "All the three numbers are equal"
    else
        echo "I can not figure out which number is biger"
    fi
```

OUTPUT:
Big.sh 123
Biggest no.is 3
3. AIM: Write script to print nos as $\mathbf{5 , 4 , 3 , 2 , 1}$ using while loop.
$\mathrm{i}=5$
while test $\$ \mathrm{i}!=0$
do

$$
\begin{aligned}
& \text { echo "\$i" } \\
& \text { i=`expr \$i-1` }
\end{aligned}
$$

done
OUT PUT:
Dec.sh
5
4
3
2
1
4. AIM: Write Script, using case statement to perform basic math operation as follows: + addition, - subtraction, x multiplication, / division.
if test \$\# = 3
then
case $\$ 2$ in

+ ) let $\mathrm{z}=\$ 1+\$ 3 ;$;
-) let $\mathrm{z}=\$ 1-\$ 3$;;
/) let $\mathrm{z}=\$ 1 / \$ 3$;;
$\mathrm{x} \mid \mathrm{X})$ let $\mathrm{z}=\$ 1 * \$ 3$;;
*) echo Warning - \$2 invalied operator, only +,-,x,/ operator allowed exit;;
esac
echo Answer is $\$ \mathrm{z}$
else
echo "Usage - \$0 value1 operator value2"
echo " Where, valuel and value2 are numeric values"
echo " operator can be,,$+- /, \mathrm{x}$ (For Multiplication)"
fi
OUTPUT:
\$./math.sh
$3+6$
9

7. Write a shell script that computes the gross salary of a employee according to the following rules:
i) If basic salary is $<\mathbf{1 5 0 0}$ then $\mathrm{HRA}=\mathbf{1 0 \%}$ of the basic and $\mathrm{DA}=\mathbf{9 0 \%}$ of the basic.
ii) If the basic salary is $>=\mathbf{1 5 0 0}$ then HRA= Rs500 and DA $=\mathbf{9 8 \%}$ of the basic
The basic salary is entered interactively through the key board.
echo " Enter the Basic salary"
read bs
if [ \$bs -lt 1500 ]
```
    then
    hra=`echo $bs \*10 /100 | bc`
    da=`echo $bs \* 90 / 100 | bc`
    elif [ $bs -ge 1500 ]
    then
    hra=500
    da=`echo $bs \* 98 /100 | bc`
    fi
gs=`echo $bs + $hra + $da|bc`
echo "Gross salary=$gs"
```


## OUTPUT:

Enter the Basic salary
1000
Gross salary=2000
8. Write a shell script that displays a list of all the files in the current directory to which the user has read, write and execute permissions.
for name in `ls`
do
if [ -r \$name -a -w \$name -a -x \$name ]
then
echo \$name has read write and execute permission
fi
done

## OUTPUT:

abc has read write and execute permission
vits has read write and execute permission
9. Develop an interactive script that asks for a word and a file name and then tells how many times that word occurred in the file.
echo "Enter word"
read w
echo "enter file name"
$\operatorname{read} \mathrm{f}$
grep -c \$w \$f

## OUTPUT:

Enter word
vits
enter file name
sri
4
10. Write a shell script that takes a login name a s command-line argument and reports when that person logs in who | grep \$1
11. Write a shell script which receives two file names a $s$ arguments. It should check whether the two file contents are same or not. If they are same then second file should be deleted.

```
if cmp $1 $2
```

then
rm \$2
echo "two files are same, so second file $\$ 2$ is deleted"
else
echo " files are not same"
fi

## OUTPUT:

\$sh compare x y
files are not same
12. Write an interactive file-handling shell program. Let it offer the user the choice of copying, removing, renaming, or linking files. Once the user has made a choice, have the program ask the user for the necessary information, such as the file name and so on.

```
echo "1.COPY"
echo "2.RENAME"
echo "3.REMOVE"
echo "4.LINK"
echo "5.EXIT"
```

```
echo "Enter your choice"
read ch
case $ch in
1) echo "Enter the sources file"
    read s
        echo "Enter the destination file "
        read d
    cp $s $d
;;
2) echo "Enter the old file name"
    read of
        echo "enter the new file name"
        read nf
        mv $of $nf
    ;
3) echo "Enter file name to delete"
        read $df
        rm $df
        ;;
    4) echo "Enter the file1"
        read f1
        echo "enter the file2"
        read f2
        ln $f1 $f2
        ;
5) exit 0
    #
    esac
13. Write a shell script to search for particular element from an array of
elements.
```

echo "Enter total no of elements"
read $n$
$\mathrm{i}=0$
echo "Enter the elements"
while [ \$i-lt \$n ]
do
read a[i]
$\mathrm{i}=` \operatorname{expr} \$ \mathrm{i}+1 `$
done

```
echo " Enter the element to search"
read k
j=0
flag=0
while [ $j -lt $n ]
do if [ $k -eq ${a[j]} ]
then
flag=1
break
fi
j=`expr $j + 1`
done
if [ $flag -eq 1]
then
    echo "number is found at `expr $j + 1` position"
else echo " Element is not found"
fi
```

OUTPUT:

Enter total no of elements
5
Enter the elements
13456
Enter the element to search
4
number is found at 3 position
14. Write a shell script to accept a string in the command line and reverse the same string.

```
s=`echo $1 | wc -c`
while [ $s -gt 0 ]
    do
        temp=`echo $1 | cut -c $s`
        s=`expr $s - 1`
    templ="$temp1$temp"
    done
        echo "Reverse string is $temp1"
```


## OUTPUT:

\$sh reverse cse
Reverse string is esc
15. Write a shell script to find out whether the given string is Palindrome or not

```
    s=`echo $1 | wc -c`
    while [ $s -gt 0 ]
    do
        temp=`echo $1 | cut -c $s`
        s=`expr $s - 1`
    temp1="$temp1$temp"
    done
    echo "Reverse string is $temp1"
        if [ $1 = $temp1 ]
then
            echo "The given string is palindrome"
            else
        echo " Not palindrome"
            fi
```


## OUTPUT:

\$sh palin liril
The given string is palindrome

## 16. AIM: SHELL SCRIPT TO PRINT THE FIBONACCI NUMBERS

```
echo enter the number
read n
i=1
f=0
s=1
next=`expr $f + $s`
echo $f
echo $s
echo $next
while [ $i -le `expr $n-3`]
do
f=$s
s=$next
next=`expr $f + $s`
echo $next
i=`expr $i + 1`
done
```


## OUTPUT

\$fibonac.sh
enter the number : 7
11235813

