

OUTPUT

```
medir@kritik-script:~/Desktop$ ./basic_arith.sh
cd kritik-script
gedit basic.sh
.....
basicarith.sh

Enter a integer !
Enter a integer 1
Enter a integer 2
.....
echo "Addition = " $((a+b))
echo "Subtraction = " $((a-b))
echo "Multiplication = " $((a*b))
echo "Division = " $((a/b))
echo "Modulus = " $((a%b))
echo "Exponent = " $((a**b))
```

Program

- 1) Program to perform basic arithmetic operations

2) /

OUTPUT

Enter three numbers 6 10 2

Program

```
read -p "Enter three numbers" a b c
if [ $a -gt $b -a $a -gt $c ]
then
    echo "$a is the largest number"
elif [ $b -gt $a -a $b -gt $c ]
then
    echo "$b is the largest number"
else
    echo "$c is the largest number"
fi
```

3) Menu driven program for calculator

Output
program

```
Enter Two numbers 15 15
Enter the choice:
1
Result : 30
Press Enter and press q to quit
[2] 15 15
q
read ch
case {ch in
    1) res = $(($a + $b))
    2) res = $(($a - $b))
    3) res = $(($a * $b))
    4) res = ${((($a / $b))})
    *) echo "invalid choice"
esac
echo "Result : $res"
```

4) program to read n numbers and print the sum

output
program

Enter the limit
5
Enter the number 1
Enter the number 2
Enter the number 3
Enter the number 4
Enter the number 5
The sum is
done

```
echo "Enter the limit"
read n
sum=0
for ((i=1;i<=n;i++))
do
    echo -n "Enter the number" num
    sum=$((sum+num))
done
echo "The sum = $sum"
```

The sum is

22

5) program to print factorial of a number

output

program

Enter the numbers

read -p "Enter the number" a

f=1

for ((i=1 ; i <= a ; i++))

do

f=\$((f*i))

done

echo "Factorial of \$a is \$f"

OUTPUT

Enter the limit 5

0 1 1 2 3 5 8

6) program to print first N fibonacci series

Program

read -p "Enter the limit" a

f=0

s=1
echo -n \$f
echo \$s

echo \$((i=0; i<a; i+1))

do

t=\$((f+s))

echo -n " \$t "

f=

\$s

s=t

done

echo "

1) Program to display all files and directory under
home directory

program

```
i=1
cd ~
for item in *
do
    echo "$item ${!item}: $item"
done
```

8) program to check whether a number is prime or not

Output

Enter a number 5

5 is a prime number

```
program
read -p "Enter a number" a
f=0
for (( i=2 ; i<a ; i++ ))
do
    if [ $(( ${a%$i})) -eq 0 ]
    then
        f=1
    fi
done
if [ $f -eq 0 ]
then
    echo "$a is prime number"
else
    echo "$a is not a prime number"
fi
```

a) program to check whether a number is
palindrome or not

Output

Program

```
read -p "Enter a number" a
s=0
b=$a
while [ ${a%10} -gt 0 ]
do
    f=$(( ${a%10} ))
    s=$(( $s*10 + $f ))
    a=$(( ${a%10} ))
done
if [ $b -eq $s ]
then
    echo "$b is palindrome"
else
    echo "$b is not palindrome"
fi
```

Ques

Output

Enter a number 153

153 is a Armstrong number

Program

```
read -p "Enter a number" a
s=0
b=${a}
while [ ${a} -gt 0 ]
do
    f=$(( ${a}%10 ))
    s=$(( $s + $(($f * $f * $f)) ))
    a=$(( ${a}/10 ))
done
if [ ${s} -eq ${b} ]
then
    echo "${b} is Armstrong number"
else
    echo "${b} is not Armstrong number"
fi
```

OUTPUT

Enter the range 1 10

The prime numbers are

```
1) Program to display all prime numbers between  
a specific range  
  
program  
read -p "Enter the range " a b  
echo "The prime numbers are : "  
while [ $a -le $b ]  
do  
    f=0  
    for ((i=2;i<=a;j++))  
    do  
        if [ $((a%j)) -eq 0 ]  
        then  
            f=1  
        fi  
    done  
    if [ $f -eq 0 ]  
    then  
        echo "$a"  
    fi  
    a=$((a+1))
```

OUTPUT

Enter a name **Malayalam**
Malayalam is palindrome

12) Program to check whether a string is palindromic or not

Program

```
echo "Enter a name"
read name
name1=$(echo $name | rev)
if [ "$name" == "$name1" ]
then
    echo "$name is palindrome"
else
    echo "$name is not a palindrome"
fi
```

13) Menu driven program to check whether a file exists or not. If file exists check whether it is

- a. ordinary file
- b. directory
- c. readable, writable, executable or not and empty

Program

```
echo -n "Enter file name:"  
read fname  
echo  
if [-e $fname]  
then  
    echo "{$fname exists"  
    if [-f $fname]  
    then  
        echo "{$fname is an ordinary file"  
    fi  
    if [-d $fname]  
    then  
        echo "{$fname is a directory"  
    else  
        echo "{$fname is not a directory"  
    fi
```

```
if [-r $fname]
then
    echo "{$fname is readable}"
else
    echo "{$fname is not readable}"
fi
if [-w $fname]
then
    echo "{$fname is writable}"
else
    echo "{$fname is not writable}"
fi
if [-x $fname]
then
    echo "{$fname is executable}"
else
    echo "{$fname is not executable}"
fi
if [-s $fname]
then
    echo "{$fname is not empty}"
else
    echo "{$fname is empty}"
fi
else
    echo "file $fname NOT EXISTS"
fi
```

14) Program to display sum of elements of an array

Program

```
declare -a ar
sum=0
echo "Enter number of elements"
read n
echo "Enter the elements"
for ((i=0;i<n;i++))
do
    read a[i]
    sum=$((sum + ${a[i]}))
done
echo "Array elements are "
echo ${a[@]}
echo "The sum is $sum"
```

15) Program to sort elements of an array

Program

```
declare -a ar
echo -p "Enter number of elements " n
echo "Enter the elements"
for ((i=0; i<n; i++))
do
    read a[i]
done
echo "Numbers before sorting"
echo ${a[@]}

for ((i=0; i<n; i++))
do
    for ((j=i+1; j<n; j++))
    do
        if [ ${a[i]} -gt ${a[j]} ]
        then
            t=${a[i]}
            a[i]=${a[j]}
            a[j]=${t}
        done
    fi
done
```

Enter the
n 1 2 3 4 5 8

```
read -p "Enter the numbers" n
for ((i=1; i<=10; i++))
do
    echo " ${a[@]} ${i} ${!((a*))}"
```

```
echo "Numbers after sorting"
echo ${a[@]}
```