Write a program that takes a command-line argument n and prints a table of the powers of 2 that are less than or equal to 2^n.

```
#!/bin/bash -x
read -p "Enter the nth value= " n;
result=1;
for (( i=1; i<n; i++ ))
      do
            result=$(($result*2))
            echo 2^$i "    " $result
      done
```

Write a program that takes a command-line argument n and prints the nth harmonic number. Harmonic Number is of the form

```
#!/bin/bash -x
read -p "Enter the nth value= " n;
result=1/1;
for (( i=2; i<=n; i++ ))
      do
            result=$result+1/$i
      done
echo $result
```

Write a program that takes a input and determines if the number is a prime.

```
#!/bin/bash -x
read -p "enter the number to be checked= " n;
flag=0
for (( c=2; c<=$n/2; c++ ))
      do
      d=$(($n%$c))
            if [ $d -eq 0 ]
                  then
                        flag=1
            fi
      done

            if [ $n -eq 1 ]
                  then
                        echo $n is not prime or composite
            elif [ $flag -eq 0 ]
                  then
                        echo $n is prime
            else
```

```
                        echo $n is composite
            fi
```

Extend the program to take a range of number as input and output the Prime Numbers in that range.

```
#!/bin/bash -x
read -p "enter the starting number to be checked= " n;
read -p "enter the ending number to be checked= " en;
for (( i=n; i<en; i++ ))
      do
            flag=0
            for (( c=2; c<=$(($i/2)); c++ ))
                  do
                        d=$(($i%$c))
                  if [ $d -eq 0 ]
                        then
                              flag=1
                  fi
                  done

                  if [ $i -eq 1 ]
                        then
                              echo $i is not prime or composite
                        elif [ $flag -eq 0 ]
                              then
                                    echo $i is prime
                        else
                              echo $i is composite
                  fi
      done
```

Write a program that computes a factorial of a number taken as input.
5 Factorial – 5! = 1 * 2 * 3 * 4 * 5

```
#!/bin/bash -x
read -p "Enter the nth value= " n;
result=1;
for (( i=1; i<=n; i++ ))
      do
            result=$(($result*i))
      done
echo $n!=$(($result))
```

Write a program to compute Factors of a number N using prime factorization method.
Logic -> Traverse till i*i <= N instead of i <= N for efficiency.
O/P -> Print the prime factors of number N.

```
#!/bin/bash -x
counter=0
read -p "enter number for which factor " x;
for (( i=2; $x>1; i++ ))
    do
        z=$(($x%$i))
        while [ $z -eq 0 ]
            do
                factors[((counter++))]=$i
                x=$(($x/$i))
                z=$(($x%$i))
            done
    done
echo ${factors[@]}
```