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 number to find factorial: " n

sum=1;

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

do

sum=$(( $sum\*$i ))

done

echo $sum

2.Write a program that takes a command-line argument n and prints the nth harmonic

number. Harmonic Number is of the form

#!/bin/bash

read -p "Enter a number to print Harmonic value: " n

harmonic=0;

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

do

harmonic=$(( $harmonic+$(( 1/$i | bc -l )) ))

done

echo $harmonic;

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 number to print table: " n

base=2;

val=$(( 2\*\*n ))

for (( i=1; i<=$(( $val/2 )); i++ ))

do

sum=$(( $base\*$i ))

echo $sum;

done

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

read -p "Enter a Num to Check: " num

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

do

while [ $(($num%$i)) -eq 0 ]

do

num=$(($num/$i))

echo "$i"

done

done

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 number: " M

for (( i=2; i <= $M; i++ ))

do

p=0

for (( j=2; j <= $i-1; j++ ))

do

if [ `expr $i % $j` -eq 0 ]

then

p=1

break

fi

done

if [ `expr $p` -eq 0 ]

then

echo $i is a prime number

fi

done

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

#!/bin/bash -x

read -p "Enter the number to check: " n

if [ $n -eq 1 ]

then

echo "$n is not a prime number"

exit

else

for (( i=2; i<=n/2; i++ ))

do

if [ $((n%i)) -eq 0 ]

then

echo "$n is not a prime number."

exit

fi

done

fi

echo "$n is a prime number."

