Efficient program to print all prime factors of a given number

Given a number n, write an efficient function to print all prime factors of n. For example, if the input number is 12, then output should be “2 2 3”. And if the input number is 315, then output should be “3 3 5 7”.

Recommended: Please solve it on “*PRACTICE*” first, before moving on to the solution.

Following are the steps to find all prime factors.
1) While n is divisible by 2, print 2 and divide n by 2.
2) After step 1, n must be odd. Now start a loop from i = 3 to square root of n. While i divides n, print i and divide n by i, increment i by 2 and continue.
3) If n is a prime number and is greater than 2, then n will not become 1 by above two steps. So print n if it is greater than 2.

C/C++

```
// Program to print all prime factors
# include <stdio.h>
# include <math.h>

// A function to print all prime factors of a given number n
void primeFactors(int n)
{
    // Print the number of 2s that divide n
    while (n%2 == 0)
    {
        printf("%d ", 2);
        n = n/2;
    }

    // n must be odd at this point.  So we can skip
    // one element (Note i = i +2)
    for (int i = 3; i <= sqrt(n); i = i+2)
```

```
            printf("%d ", i);
            n = n/i;
        }
    }

    // This condition is to handle the case when n
    // is a prime number greater than 2
    if (n > 2)
        printf ("%d ", n);
}

/* Driver program to test above function */
int main()
{
    int n = 315;
    primeFactors(n);
    return 0;
}
```

Java

```
// Program to print all prime factors
import java.io.*;
import java.lang.Math;

class GFG
{
    // A function to print all prime factors
    // of a given number n
    public static void primeFactors(int n)
    {
        // Print the number of 2s that divide n
        while (n%2==0)
        {
            System.out.print(2 + " ");
            n /= 2;
        }

        // n must be odd at this point.  So we can
        // skip one element (Note i = i +2)
        for (int i = 3; i <= Math.sqrt(n); i+= 2)
        {
            // While i divides n, print i and divide n
            while (n%i == 0)
            {
                System.out.print(i + " ");
                n /= i;
            }
        }
```

```
            System.out.print(n);
    }

    public static void main (String[] args)
    {
        int n = 315;
        primeFactors(n);
    }
}
```

Python

```
# Python program to print prime factors

import math

# A function to print all prime factors of
# a given number n
def primeFactors(n):

    # Print the number of two's that divide n
    while n % 2 == 0:
        print 2,
        n = n / 2

    # n must be odd at this point
    # so a skip of 2 ( i = i + 2) can be used
    for i in range(3,int(math.sqrt(n))+1,2):

        # while i divides n , print i ad divide n
        while n % i== 0:
            print i,
            n = n / i

    # Condition if n is a prime
    # number greater than 2
    if n > 2:
        print n

# Driver Program to test above function

n = 315
primeFactors(n)

# This code is contributed by Harshit Agrawal
```

C#

```
namespace prime
{
public class GFG
{

    // A function to print all prime
    // factors of a given number n
    public static void primeFactors(int n)
    {
        // Print the number of 2s that divide n
        while (n % 2 == 0)
        {
            Console.Write(2 + " ");
            n /= 2;
        }

        // n must be odd at this point. So we can
        // skip one element (Note i = i +2)
        for (int i = 3; i <= Math.Sqrt(n); i+= 2)
        {
            // While i divides n, print i and divide n
            while (n % i == 0)
            {
                Console.Write(i + " ");
                n /= i;
            }
        }

        // This condition is to handle the case whien
        // n is a prime number greater than 2
        if (n > 2)
            Console.Write(n);
    }

    // Driver Code
    public static void Main()
    {
        int n = 315;
        primeFactors(n);
    }

}
}

// This code is contributed by Sam007
```

PHP

```
<?php
// PHP Efficient program to print all
```

```php
function primeFactors($n)
{
    // Print the number of
    // 2s that divide n
    while($n % 2 == 0)
    {
        echo 2," ";
        $n = $n / 2;
    }

    // n must be odd at this
    // point. So we can skip
    // one element (Note i = i +2)
    for ($i = 3; $i <= sqrt($n);
                  $i = $i + 2)
    {

        // While i divides n,
        // print i and divide n
        while ($n % $i == 0)
        {
            echo $i," ";
            $n = $n / $i;
        }
    }

    // This condition is to
    // handle the case when n
    // is a prime number greater
    // than 2
    if ($n > 2)
        echo $n," ";
}

    // Driver Code
    $n = 315;
    primeFactors($n);

// This code is contributed by aj_36
?>
```

Output:

```
3 3 5 7
```

How does this work?

The steps 1 and 2 take care of composite numbers and step 3 takes care of prime numbers. To

remaining prime factor must be odd (difference of two prime factors must be at least 2), this explains why i is incremented by 2.

Now the main part is, the loop runs till square root of n not till n. To prove that this optimization works, let us consider the following property of composite numbers.

Every composite number has at least one prime factor less than or equal to square root of itself.

This property can be proved using counter statement. Let a and b be two factors of n such that a*b = n. If both are greater than √n, then a.b > √n, * √n, which contradicts the expression "a * b = n".

In step 2 of the above algorithm, we run a loop and do following in loop

a) Find the least prime factor i (must be less than √n,)

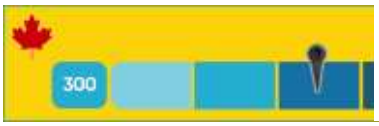
b) Remove all occurrences i from n by repeatedly dividing n by i.

c) Repeat steps a and b for divided n and i = i + 2. The steps a and b are repeated till n becomes either 1 or a prime number.

Related Article :

Prime Factorization using Sieve O(log n) for multiple queries

Thanks to **Vishwas Garg** for suggesting the above algorithm. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above



Recommended Posts:

Efficient program to print the number of factors of n numbers

C Program for efficiently print all prime factors of a given number

Java Program for efficiently print all prime factors of a given number

Program to print factors of a number in pairs

Print all prime factors and their powers

Prime factors of a big number

Sum of Factors of a Number using Prime Factorization

Number which has the maximum number of distinct prime factors in the range M to N

Product of unique prime factors of a number

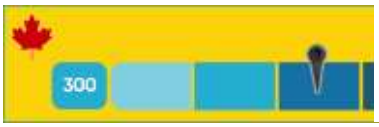
Number of steps to convert to prime factors

Maximum number of unique prime factors

Super Ugly Number (Number whose prime factors are in given set)

Print the nearest prime number formed by adding prime numbers to N

Improved By : jit_t, NehaChoutapelly



Article Tags : Dynamic Programming, Mathematical, combionatrics, Prime Number, prime-factor, sieve, Yahoo

Practice Tags : Yahoo, Dynamic Programming, Mathematical, Prime Number, sieve

17

To-do Done

2.4

Based on **120** vote(s)

Feedback/ Suggest Improvement | Add Notes | Improve Article

Please write to us at contribute@geeksforgeeks.org to report any issue with the above content.

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments | Share this post!

A computer science portal for geeks

5th Floor, A-118,
Sector-136, Noida, Uttar Pradesh - 201305
feedback@geeksforgeeks.org

COMPANY
About Us
Careers
Privacy Policy
Contact Us

LEARN
Algorithms
Data Structures
Languages
CS Subjects
Video Tutorials

PRACTICE
Company-wise
Topic-wise
Contests
Subjective Questions

CONTRIBUTE
Write an Article
Write Interview Experience
Internships
Videos