Sieve of Eratosthenes

Difficulty Level : Medium ● Last Updated : 31 Mar, 2021

Given a number n, print all primes smaller than or equal to n. It is also given that n is a small number.

Example:

Input : *n =10*
Output : *2 3 5 7*

Input : *n = 20*
Output: *2 3 5 7 11 13 17 19*

The sieve of Eratosthenes is one of the most efficient ways to find all primes smaller than n when n is smaller than 10 million or so (Ref Wiki).

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

Following is the algorithm to find all the prime numbers less than or equal to a given integer *n* by the Eratosthene's method:
When the algorithm terminates, all the numbers in the list that are not marked are prime.

Explanation with Example:
Let us take an example when n = 50. So we need to print all prime numbers smaller

than or equal to 50.

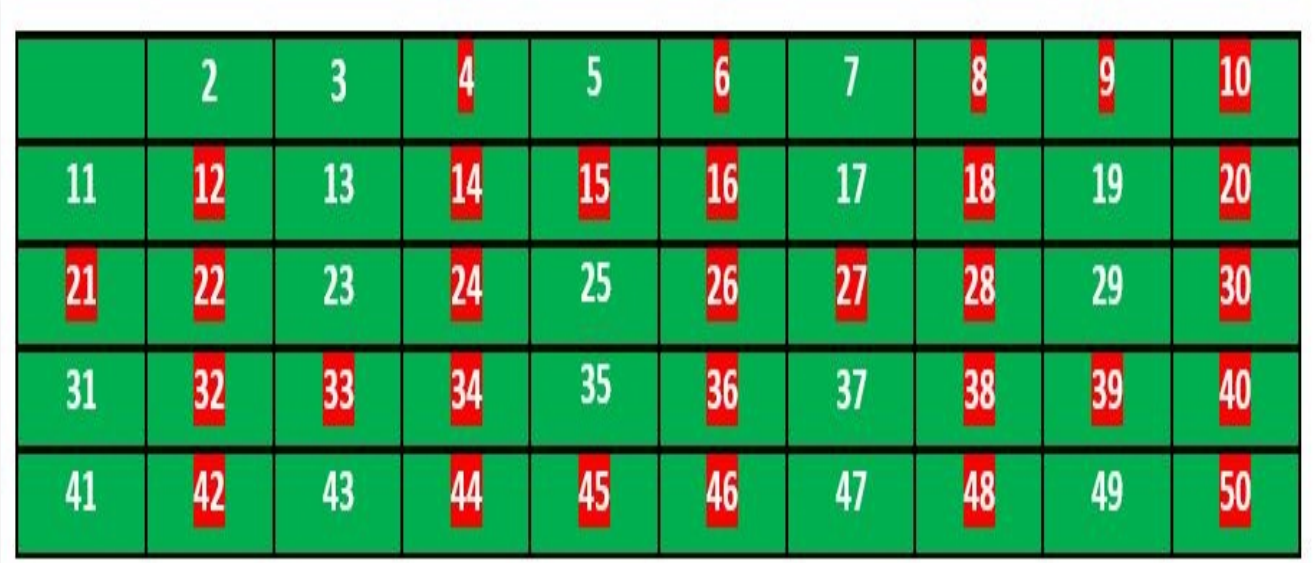
We create a list of all numbers from 2 to 50.

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50

According to the algorithm we will mark all the numbers which are divisible by 2 and are greater than or equal to the square of it.

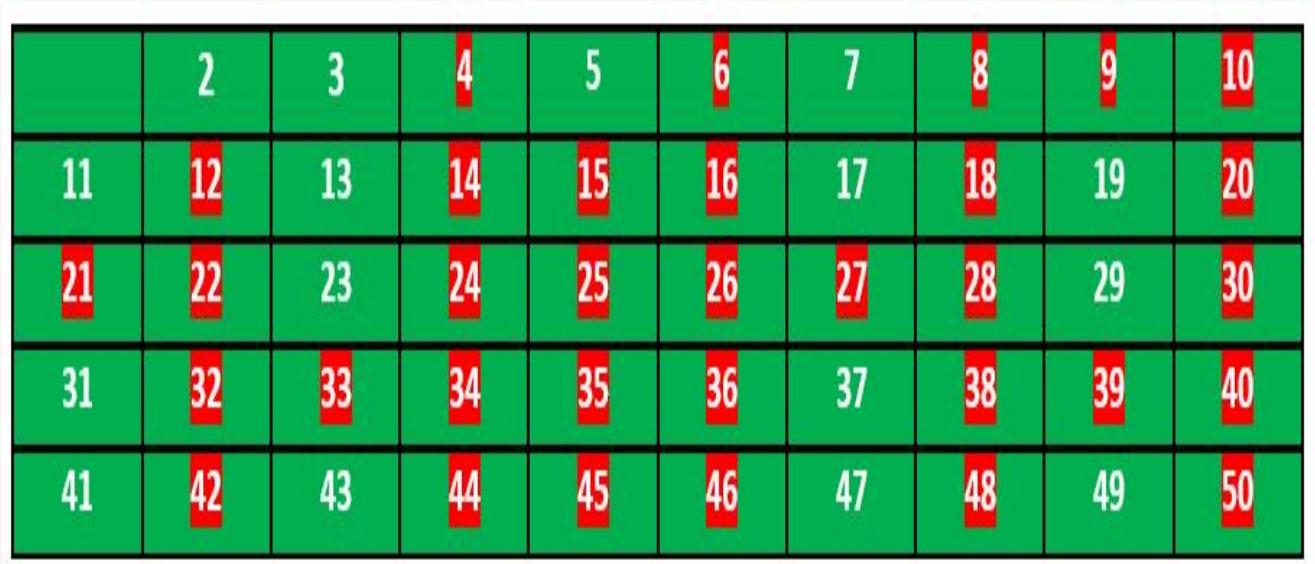
	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50

Now we move to our next unmarked number 3 and mark all the numbers which are multiples of 3 and are greater than or equal to the square of it.



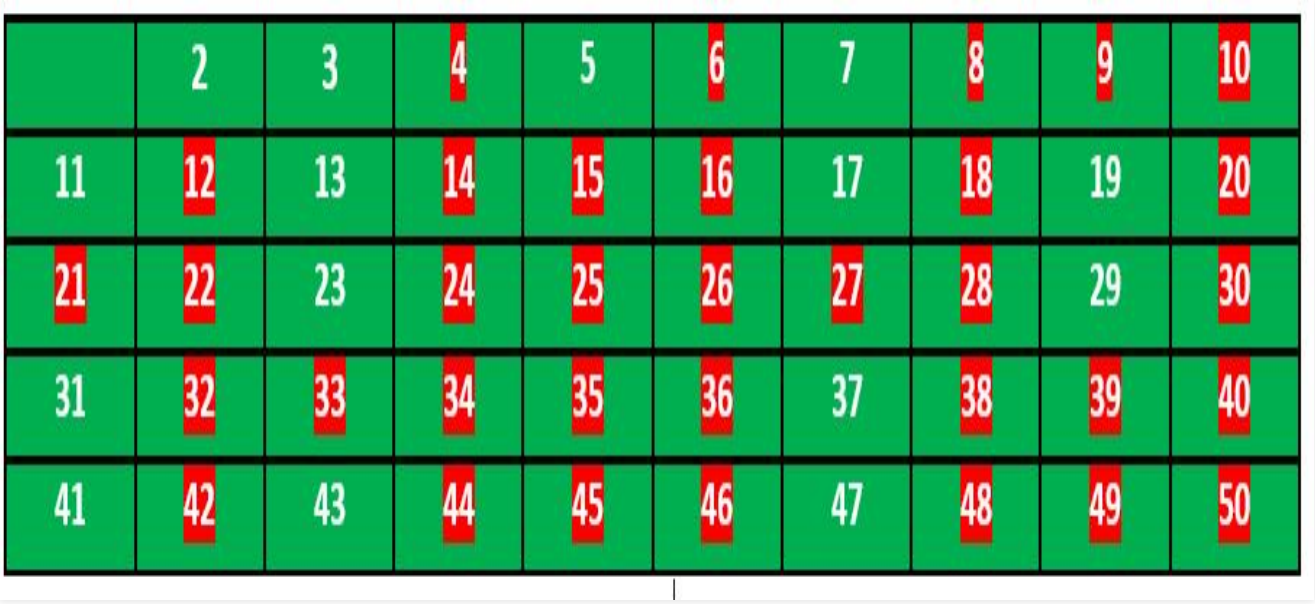
	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50

We move to our next unmarked number 5 and mark all multiples of 5 and are greater than or equal to the square of it.



	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50

We continue this process and our final table will look like below:



So the prime numbers are the unmarked ones: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47.

Thanks to **Krishan Kumar** for providing above explanation.

Implementation:

Following is the implementation of the above algorithm. In the following implementation, a boolean array arr[] of size n is used to mark multiples of prime numbers.

C++

```
// C++ program to print all primes
// smaller than or equal to
// n using Sieve of Eratosthenes
#include <bits/stdc++.h>
using namespace std;

void SieveOfEratosthenes(int n)
{
    // Create a boolean array
    // "prime[0..n]" and initialize
    // all entries it as true.
    // A value in prime[i] will
    // finally be false if i is
    // Not a prime, else true.
    bool prime[n + 1];
    memset(prime, true, sizeof(prime));

    for (int p = 2; p * p <= n; p++)
    {
        // If prime[p] is not changed,
        // then it is a prime
        if (prime[p] == true)
        {
```

```
            // Update all multiples
            // of p greater than or
            // equal to the square of it
            // numbers which are multiple
            // of p and are less than p^2
            // are already been marked.
            for (int i = p * p; i <= n; i += p)
                prime[i] = false;
        }
    }

    // Print all prime numbers
    for (int p = 2; p <= n; p++)
        if (prime[p])
            cout << p << " ";
}

// Driver Code
int main()
{
    int n = 30;
    cout << "Following are the prime numbers smaller "
         << " than or equal to " << n << endl;
    SieveOfEratosthenes(n);
    return 0;
}
```

Java

```
// Java program to print all
// primes smaller than or equal to
// n using Sieve of Eratosthenes

class SieveOfEratosthenes {
    void sieveOfEratosthenes(int n)
    {
        // Create a boolean array
        // prime, else true.
        boolean prime[] = new boolean[n + 1];
        for (int i = 0; i <= n; i++)
            prime[i] = true;

        for (int p = 2; p * p <= n; p++)
        {
            // If prime[p] is not changed, then it is a
            // prime
            if (prime[p] == true)
```

Related Articles

```
            {
                // Update all multiples of p
                for (int i = p * p; i <= n; i += p)
                    prime[i] = false;
            }
        }

        // Print all prime numbers
        for (int i = 2; i <= n; i++)
        {
            if (prime[i] == true)
                System.out.print(i + " ");
        }
    }

    // Driver Code
    public static void main(String args[])
    {
        int n = 30;
        System.out.print(
            "Following are the prime numbers ");
        System.out.println("smaller than or equal to " + n);
        SieveOfEratosthenes g = new SieveOfEratosthenes();
        g.sieveOfEratosthenes(n);
    }
}

// This code has been contributed by Amit Khandelwal.
```

Python

```
# Python program to print all
# primes smaller than or equal to
# n using Sieve of Eratosthenes


def SieveOfEratosthenes(n):

    # Create a boolean array
    # "prime[0..n]" and initialize
    #  all entries it as true.
    # A value in prime[i] will
    # finally be false if i is
    # Not a prime, else true.
    prime = [True for i in range(n+1)]
    p = 2
    while (p * p <= n):

        # If prime[p] is not
        # changed, then it is a prime
        if (prime[p] == True):
```

```
            # Update all multiples of p
            for i in range(p * p, n+1, p):
                prime[i] = False
        p += 1

    # Print all prime numbers
    for p in range(2, n+1):
        if prime[p]:
            print p,


# Driver code
if __name__ == '__main__':
    n = 30
    print "Following are the prime numbers smaller",
    print "than or equal to", n
    SieveOfEratosthenes(n)
```

C#

```
// C# program to print all primes
// smaller than or equal to n
// using Sieve of Eratosthenes
using System;

namespace prime {
public class GFG {

    public static void SieveOfEratosthenes(int n)
    {

        // Create a boolean array
        // "prime[0..n]" and
        // initialize all entries
        // it as true. A value in
        // prime[i] will finally be
        // false if i is Not a
        // prime, else true.

        bool[] prime = new bool[n + 1];

        for (int i = 0; i < n; i++)
            prime[i] = true;

        for (int p = 2; p * p <= n; p++)
        {
            // If prime[p] is not changed,
            // then it is a prime
            if (prime[p] == true)
            {
                // Update all multiples of p
                for (int i = p * p; i ; i += p)
```

```
                    prime[i] = false;
                }
            }

            // Print all prime numbers
            for (int i = 2; i <= n; i++)
            {
                if (prime[i] == true)
                    Console.Write(i + " ");
            }
        }

        // Driver Code
        public static void Main()
        {
            int n = 30;
            Console.WriteLine(
                "Following are the prime numbers");
            Console.WriteLine("smaller than or equal to " + n);
            SieveOfEratosthenes(n);
        }
    }
    }

    // This code is contributed by Sam007.
```

PHP

```
<?php
// php program to print all primes smaller
// than or equal to n using Sieve of
// Eratosthenes

function SieveOfEratosthenes($n)
{
    // Create a boolean array "prime[0..n]"
    // and initialize all entries it as true.
    // A value in prime[i] will finally be
    // false if i is Not a prime, else true.
    $prime = array_fill(0, $n+1, true);

    for ($p = 2; $p*$p <= $n; $p++)
    {

        // If prime[p] is not changed,
        // then it is a prime
        if ($prime[$p] == true)
        {

            // Update all multiples of p
            for ($i = $p*$p; $i <= $n; $i += $p)
                $prime[$i] = false;
```

```
            }
        }

        // Print all prime numbers
        for ($p = 2; $p <= $n; $p++)
            if ($prime[$p])
                echo $p." ";
    }

    // Driver Code
        $n = 30;
        echo "Following are the prime numbers "
         ."smaller than or equal to " .$n."\n" ;
        SieveOfEratosthenes($n);

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

Javascript

```
<script>

// javascript program to print all
// primes smaller than or equal to
// n using Sieve of Eratosthenes


function sieveOfEratosthenes(n)
{
    // Create a boolean array
    // "prime[0..n]" and
    // initialize all entries
    // it as true. A value in
    // prime[i] will finally be
    // false if i is Not a
    // prime, else true.
    prime = Array.from({length: n+1}, (_, i) => true);

    for (p = 2; p * p <= n; p++)
    {
        // If prime[p] is not changed, then it is a
        // prime
        if (prime[p] == true)
        {
            // Update all multiples of p
            for (i = p * p; i <= n; i += p)
                prime[i] = false;
        }
    }

    // Print all prime numbers
    for (i = 2; i <= n; i++)
```

```
    {
        if (prime[i] == true)
            document.write(i + " ");
    }
}

// Driver Code
var n = 30;
document.write(
    "Following are the prime numbers ");
document.write("smaller than or equal to " + n+"<br>");
sieveOfEratosthenes(n);

// This code is contributed by 29AjayKumar

</script>
```

Output

```
Following are the prime numbers smaller  than or equal to 30
2 3 5 7 11 13 17 19 23 29
```

Time complexity : O(n*log(log(n)))

You may also like to see :

- How is the time complexity of Sieve of Eratosthenes is n*log(log(n))?
- Segmented Sieve.
- Sieve of Eratosthenes in O(n) time complexity

Attention reader! Don't stop learning now. Get hold of all the important DSA concepts with the **DSA Self Paced Course** at a student-friendly price and become industry ready. To complete your preparation from learning a language to DS Algo and many more, please refer **Complete Interview Preparation Course**.

In case you wish to attend live classes with industry experts, please refer **Geeks Classes Live** and **Geeks Classes Live USA**

Like 0

Previous Next

RECOMMENDED ARTICLES

Page : 1 2

01 **Sum of all Primes in a given range using Sieve of Eratosthenes**
17, Oct 18

02 **How is the time complexity of Sieve of Eratosthenes is n*log(log(n))?**
15, Jan 20

03 **Sieve of Eratosthenes in 0(n) time complexity**
22, Jan 17

04 **Segmented Sieve**
24, Nov 15

05 **Segmented Sieve (Print Primes in a Range)**
24, Nov 15

06 **Sieve of Sundaram to print all primes smaller than n**
12, Jan 16

07 **Number of unmarked integers in a special sieve**
29, May 18

08 **Bitwise Sieve**
26, Nov 17

Article Contributed By :

GeeksforGeeks

Vote for difficulty

Current difficulty : Medium

Easy | Normal | Medium | Hard | Expert

Improved By : Mithun Kumar, radhesh, shivamnayak, diwakarjaiswal880, vsasvipul, siddhantj17, mrmgssingh, 29AjayKumar

Article Tags : GE, MAQ Software, number-theory, Prime Number, Qualcomm, sieve, VMWare, Dynamic Programming, Mathematical

Practice Tags : VMWare, MAQ Software, Qualcomm, GE, number-theory, Dynamic Programming, Mathematical, Prime Number, sieve

Improve Article | Report Issue

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

Load Comments

GeeksforGeeks
5th Floor, A-118,
Sector-136, Noida, Uttar Pradesh - 201305

feedback@geeksforgeeks.org

Courses
Company-wise
Topic-wise
How to begin?
Write an Article
Write Interview Experience
Internships
Videos