

# Forum:[Hack Forums](#)

Thread: [Algorithm Series - 3 | Sieve of Eratosthenes](#)

**Deadbyte** - February 13, 2015, 8:13 am

## Algorithm Series - 3 | Sieve of Eratosthenes

### Intro

The Sieve of Eratosthenes is a simple test to check if a number is a prime number. If someone wants to check for a certain number  $n$  to be prime, they do as follows:

Write numbers from 2 to that number onto a piece of paper.

Do the following:

Get the next number that is not crossed out. Cross out all multiples of that number on your sheet. Repeat this for all numbers until you reach the square root of  $n$ . If  $n$  has been crossed out it is not prime.

Animated:

[img][http://upload.wikimedia.org/wikipedia/commons/b/b9/Sieve\\_of\\_Eratosthenes\\_animation.gif](http://upload.wikimedia.org/wikipedia/commons/b/b9/Sieve_of_Eratosthenes_animation.gif)[img]

### Algorithm

1. Create a list of consecutive integers from 2 through  $n$ : (2, 3, 4, ...,  $n$ ).
2. Initially, let  $p$  equal 2, the first prime number.
3. Starting from  $p$ , enumerate its multiples by counting to  $n$  in increments of  $p$ , and mark them in the list (these will be  $2p$ ,  $3p$ ,  $4p$ , ...; the  $p$  itself should not be marked).
4. Find the first number greater than  $p$  in the list that is not marked. If there was no such number, stop. Otherwise, let  $p$  now equal this new number (which is the next prime), and repeat from step 3.

### Example

To find all the prime numbers less than or equal to 30, proceed as follows.

First generate a list of integers from 2 to 30:

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

First number in the list is 2; cross out every 2nd number in the list after it by counting up from 2 in increments of 2 (these will be all the multiples of 2 in the list):

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

Next number in the list after 2 is 3; cross out every 3rd number in the list after it by counting up from 3 in increments of 3 (these will be all the multiples of 3 in the list):

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

Next number not yet crossed out in the list after 3 is 5; cross out every 5th number in the list after it by counting up from 5 in increments of 5 (i.e. all the multiples of 5):

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

Next number not yet crossed out in the list after 5 is 7; the next step would be to cross out every 7th number in the list after 7, but they are all already crossed out at this point, as these numbers (14, 21, 28) are also multiples of smaller primes because  $7 \times 7$  is greater than 30. The numbers left not crossed out in the list at this point are all the prime numbers below 30:

2 3 5 7 11 13 17 19 23 29

C++ implementation:

PHP Code

```
<?php
```

```
#include <stdio.h>
```

```
#include <string.h>
```

```
// marks all multiples of 'a' ( greater than 'a' but less than equal to 'n') as 1.
```

```
void markMultiples(bool arr[], int a, int n)
```

```
{
    int i = 2, num;
    while ( (num = i*a) <= n )
    {
        arr[ num-1 ] = 1; // minus 1 because index starts from 0.
        ++i;
    }
}
```

```
// A function to print all prime numbers smaller than n
```

```
void SieveOfEratosthenes(int n)
```

```
{
    // There are no prime numbers smaller than 2
    if (n >= 2)
    {
        // Create an array of size n and initialize all elements as 0
        bool arr[n];
        memset(arr, 0, sizeof(arr));

        /* Following property is maintained in the below for loop
           arr[i] == 0 means i + 1 is prime
           arr[i] == 1 means i + 1 is not prime */
        for (int i=1; i<n; ++i)
        {
            if ( arr[i] == 0 )
            {
                //(i+1) is prime, print it and mark its multiples
                printf("%d ", i+1);
                markMultiples(arr, i+1, n);
            }
        }
    }
}
```

```
// Driver Program to test above function
```

```
int main()
```

```
{
    int n = 30;
```

```
printf("Following are the prime numbers below %d\n", n);
SieveOfEratosthenes(n);
return 0;
}
```

Notice:

The algorithms that I am introducing in the series are simplistic some inefficient and slow and can be optimized but the series are for educational purpose for beginners in competitive programming.

**0xDEAD10CC** - February 13, 2015, 4:52 pm

A C++ implementation using <stdio.h> and <string.h>? :S This is a joke... There's absolutely nothing about that code which closely resembles C++... I would suggest not spreading bad code and information like this.

**Deadbyte** - February 13, 2015, 6:30 pm

(Feb 13, 2015 05:52 PM)0xDEAD10CC Wrote: [\[CITE\]](#) A C++ implementation using <stdio.h> and <string.h>? :S This is a joke... There's absolutely nothing about that code which closely resembles C++... I would suggest not spreading bad code and information like this.

#<iostream> with cin and cout instead of print would make a big change ? Sorry if I seem ignorant but you can help me improve ?

**SirLord** - February 14, 2015, 10:20 am

(Feb 13, 2015 07:30 PM)Deadbyte Wrote: [\[CITE\]](#) #<iostream> with cin and cout instead of print would make a big change ? Sorry if I seem ignorant but you can help me improve ?

<http://www.parashift.com/c++-faq-lite/iostream-vs-stdio.html>

I hope that this can help you :)

**0xDEAD10CC** - February 14, 2015, 12:04 pm

(Feb 13, 2015 07:30 PM)Deadbyte Wrote: [\[CITE\]](#) #<iostream> with cin and cout instead of print would make a big change ? Sorry if I seem ignorant but you can help me improve ?

<iostream> is the proper C++ header to be using. Functions like printf() are not type safe.

**Kantos** - February 15, 2015, 12:56 pm

Or you can just include stdbool.h and call it a C implementation.

**Deadbyte** - February 15, 2015, 1:02 pm

Okay guys thank you for the advice.