

## Sieve of Eratosthenes

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

For example, if  $n$  is 10, the output should be "2, 3, 5, 7". If  $n$  is 20, the output should be "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](#)).

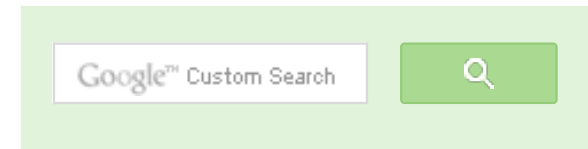
Following is the algorithm to find all the prime numbers less than or equal to a given integer  $n$  by Eratosthenes' method:

1. Create a list of consecutive integers from 2 to  $n$ : (2, 3, 4, ...,  $n$ ).
2. Initially, let  $p$  equal 2, the first prime number.
3. Starting from  $p$ , count up in increments of  $p$  and mark each of these numbers greater than  $p$  itself in the list. These numbers will be  $2p$ ,  $3p$ ,  $4p$ , etc.; note that some of them may have already been 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 number (which is the next prime), and repeat from step 3.

When the algorithm terminates, all the numbers in the list that are not marked are prime.

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

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



GeeksforGeeks



53,525 people like [GeeksforGeeks](#).



[Facebook](#)

[Interview Experiences](#)

[Advanced Data Structures](#)

[Dynamic Programming](#)

[Greedy Algorithms](#)

[Backtracking](#)

[Pattern Searching](#)

[Divide & Conquer](#)

[Mathematical Algorithms](#)

[Recursion](#)

[Geometric Algorithms](#)

```
// marks all multiples of 'a' ( greater than 'a' but less than equal to
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;
}
```

Output:

Shouldn't you expect  
a cloud with:

# SYSTEM MONITORING

Plus the experts to help run it?

**TRY MANAGED CLOUD ▶**



## Popular Posts

All permutations of a given string

Memory Layout of C Programs

Understanding "extern" keyword in C

Median of two sorted arrays

Tree traversal without recursion and without stack!

Structure Member Alignment, Padding and Data Packing

Intersection point of two Linked Lists

Lowest Common Ancestor in a BST.

Check if a binary tree is BST or not

Sorted Linked List to Balanced BST

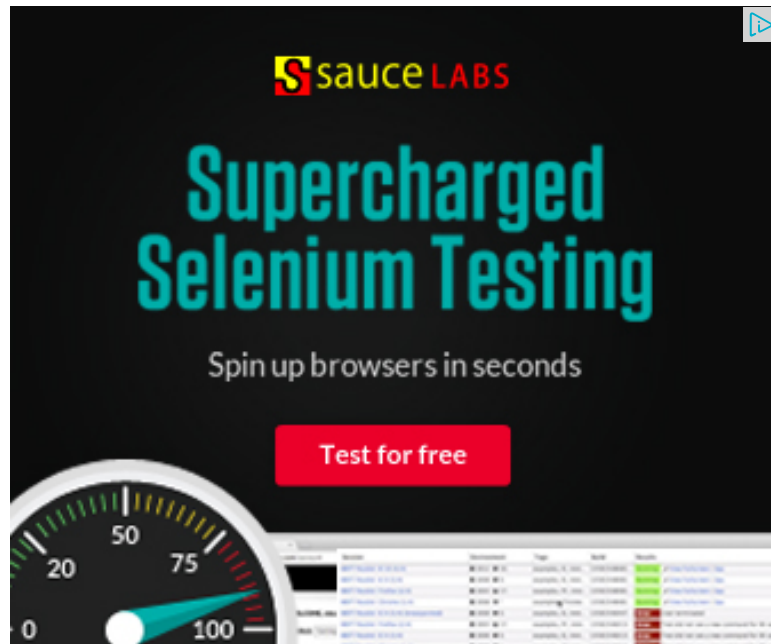
Following are the prime numbers below 30

2 3 5 7 11 13 17 19 23 29

References:

[http://en.wikipedia.org/wiki/Sieve\\_of\\_Eratosthenes](http://en.wikipedia.org/wiki/Sieve_of_Eratosthenes)

This article is compiled by **Abhinav Priyadarshi** and reviewed by GeeksforGeeks team. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above



## Related Topics:

- [Backtracking | Set 8 \(Solving Cryptarithmic Puzzles\)](#)
- [Tail Recursion](#)
- [Find if two rectangles overlap](#)
- [Analysis of Algorithm | Set 4 \(Solving Recurrences\)](#)
- [Print all possible paths from top left to bottom right of a mXn matrix](#)
- [Generate all unique partitions of an integer](#)
- [Russian Peasant Multiplication](#)

# HIGH-PERFORMANCE COMPUTING ON A UNIVERSITY BUDGET

*The quest to find tailored server capabilities on a shoestring*

To conduct successful university research and crunch massive amounts of data, you need the highest-performing hardware on the market. Unfortunately, the price tag of high performance can be daunting for researchers who rely on grants to fund their projects.

There are many off-the-shelf server options available, of course, and your department may even have a standardized, prescribed server.

The best bang for your buck, however, may be to configure a custom server to fit your exact high-performance computing needs.

WHAT NETWORKING INTERFACES ARE NEEDED?

☐ 1GbE ☐ InfiniBand  
☐ 10GbE ☐ Unsure  
☐ 40GbE

HOW MANY CPU CORES DO YOU WANT?

☐ 4 ☐ 16  
☐ 8 ☐ Unsure  
☐ 12

HOW MUCH STORAGE DO YOU WANT?

☐ 1TB ☐ 1PB  
☐ Unsure

WHAT OPERATING SYSTEM ARE YOU RUNNING?

☐ Linux Variant  
☐ Windows Server  
☐ VMware  
☐ Unsure

Unsure of how to get the best performance out of your application? Do you need assistance answering any of these questions?

Download To See Entire Infographic

**SERVICESDIRECT.**

- Russian Fast Multiplication
- Closest Pair of Points | O(nlogn) Implementation



5



Tweet

0



0

Writing code in comment? Please use [ideone.com](https://ideone.com) and share the link here.

11 Comments

GeeksforGeeks

Sort by Newest ▾



Join the discussion...



vish\_anything · 8 months ago

Can you please tell the time complexity of the algo?

1 ^ | v ·



Guest · 8 months ago

Can you please show how the time complexity of the algorithm is  $n(\log(\log n))$ ?

1 ^ | v ·



Tushar Rokade · 9 months ago

A tightened sieve (y)

^ | v ·



Megha Mantri · 9 months ago

I think it is not required to loop till  $i=n-1$ . once I reaches  $\sqrt{n}$ , all the composi marked.

3 ^ | v ·



poonam → Megha Mantri · 2 months ago

AND then explicitly traverse the array as the numbers left from squarer marked yet... we need to print those too

705



Subscribe

## Recent Comments

Abhi You live US or India?

Google (Mountain View) interview · 15 minutes ago

Aman Hi, Why arent we checking for conditions...

Write a C program to Delete a Tree. · 55 minutes ago

kzs please provide solution for the problem...

Backtracking | Set 2 (Rat in a Maze) · 58 minutes ago

Sanjay Agarwal bool

tree::Root\_to\_leaf\_path\_given\_sum(tree...

Root to leaf path sum equal to a given number · 1 hour ago

GOPI GOPINATH @admin Highlight this sentence "We can easily...

Count trailing zeroes in factorial of a number · 1 hour ago

newCoder3006 If the array contains negative numbers also. We...

Find subarray with given sum · 1 hour ago

AdChoices ▶

▶ [C++ Code](#)

▶ [Sieve Analysis](#)

▶ [Prime Numbers](#)

[▶ C++ Example](#)

[▶ C++ Program](#)

[▶ C++ Run Time](#)

[▶ Test C++](#)

[▶ Sieve Set 3](#)

[▶ Int](#)



**TulasiRam** · 2 years ago

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

/* sieve eratostanese method of listing primes */

void unsetBit(char *b,int i){
    int blk=(i-1)/8;
    int shift=((i-1)%8);
    b[blk] &= ~(1<<shift);
}
int bitStatus(char *b,int i){
    int k,t,blk=(i-1)/8;
    int shift=((i-1)%8);
    k=b[blk]&(1<<shift);
```

[see more](#)



**bleu** · 2 years ago

Nice!...



**Venki** · 2 years ago

Prime numbers are fixed. Don't depend on runtime system to generate them, compile time using meta program and generics.

^ | v .



**rohan** • 2 years ago

good!!!

```
/* Paste your code here (You may delete these lines if not writing cor
```

^ | v .



**bepo** • 2 years ago

```
/* #include<iostream>
using namespace std;
int main()
{
    int n,p;
    cin>>n;
    int * arr;

    arr= new  int[n+1];

    for (int i=1; i<=n; i++)
    {
        arr[i]=0;
    }
    for(int i=2;i<=n;i++){
        if(arr[i]==0)
        {
            p=i;
```

see more

^ | v .



bepo · 2 years ago

```
#include<iostream>
using namespace std;
int main()
{
    int n,p;
    cin>>n;
    int * arr;

    arr= new  int[n+1];

    for (int i=1; i<=n; i++)
    {
        arr[i]=0;
    }
    for(int i=2;i<=n;i++){
        if(arr[i]==0)
        {
            p=i;
```

see more



