

Algorithms in C++: Programming Assignment 1

1. Objective

Your goal is to write a program to find prime numbers using the sieve of Eratosthenes.

2. Problem

You are to create a C++ project called *sieve*, a C++ source file called *sieve.cpp*, and class inside it called *PrimesSieve* that finds prime numbers. The program asks the user for a limit, and then finds primes up to and including that limit. When there is only one line of prime numbers in the output, the numbers should be displayed with one space between each (and no space at the end). When there are multiple lines of output, the numbers should be right-aligned to the width of the largest prime. See below.

```
***** Sieve of Eratosthenes *****
Search for primes up to: <user enters 20>
```

```
Number of primes found: 8
Primes up to 20:
2 3 5 7 11 13 17 19
```

```
***** Sieve of Eratosthenes *****
Search for primes up to: <user enters 200>
```

```
Number of primes found: 46
Primes up to 200:
 2  3  5  7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71
73 79 83 89 97 101 103 107 109 113 127 131 137 139 149 151 157 163 167 173
179 181 191 193 197 199
```

no extra space
at end of line

3. Tips

a. Use the template file found in Canvas. The framework for the program is there, and you will need to fill in the methods.

b. If a function/method is **inline**, the compiler places a copy of the code of that function/method at each point where it is called at compile time. **inline** is used for efficiency with short (one-line) functions/methods.

c. Look up **setw** in the **iomanip** library to set the width of a field that is to be outputted.

use `setw(3)`, wont need to manually count/enter spaces. just find # of digits in largest prime and use that in `setw`

d. **#include <cmath>** for the **sqrt** function.

e. You may have up to 80 characters on a line. If you cannot fit all the primes on one line, you should wrap around to the next line. To find the width of the maximum prime value and how many primes you can fit on a row, use the following code:

```
const int max_prime_width = num_digits(max_prime_),    num_digits - how many times you can divide by 10
      primes_per_row = 80 / (max_prime_width + 1);      +1 for the space in between the columns
```

Before printing each prime, determine how many spaces are needed to right-align the number, and set the field width accordingly.

if `primes_per_row > num primes`, then can use one line. If `< num_primes`, need multiple lines

f. Be sure to comment your code and put your name and Stevens pledge at the top.

g. Make sure your code works with the test.sh autograder shell script.

h. The algorithm for the sieve is found in pseudocode below.

```
***** Sieve of Eratosthenes *****  
Input: an integer  $n > 1$   
  
Let is_prime be an array of bool values, indexed by integers 2 to  $n$ ,  
initially all set to true.  
  
for  $i = 2, 3, 4, \dots$ , while  $i \leq \sqrt{n}$ :      eliminate all multiples of  $i$  up until (and including)  $\sqrt{n}$   
    if is_prime[ $i$ ] is true:  
        for  $j = i^2, i^2 + i, i^2 + 2i, \dots$ , while  $j \leq n$ :      remove multiples of  $i$  up to and including  $n$   
            is_prime[ $j$ ] = false      mark them as not prime  
  
Now all  $i$  such that is_prime[ $i$ ] is true are prime.
```

$\leq \sqrt{n}$ because any non prime number has one factor less than its square root