San José State University
Department of Computer Engineering

CMPE 180-92
Data Structures and Algorithms in C++
Fall 2017

Instructor: Ron Mak

Assignment #3.a

Assigned:	Thursday, September 7
Due:	Thursday, September 14 at 5:30 PM
CodeCheck:	http://codecheck.it/codecheck/files/17011623552sefot95xmzaj71sxktly25i9
Canvas:	Assignment 3.a. Prime Numbers
Points:	30

Prime numbers

This assignment will give you practice using a one-dimensional array that you must pass to functions. Write a C++ program that uses the Sieve of Eratosthenes to compute the prime numbers under 1000 and print then them to the standard output. At the above CodeCheck URL, complete the program **primes.cpp**.

The Sieve of Eratosthenes

See https://en.wikipedia.org/wiki/Sieve_of_Eratosthenes for a description of the sieve algorithm. Keep track of which numbers are prime or not in a one-dimensional array. In an outer loop, identify the next prime number, and then in an inner loop, mark all the prime's multiples as not prime. When you're done, all you will have left are prime numbers.

Your program must use the sieve algorithm and not repeated trial divisions.

Array and functions

Create your array in the main. Pass the array to a **`compute_primes`** function that uses the Sieve of Eratosthenes. Then pass the array to a **`print_primes`** function.

Expected output

Print the prime numbers in rows of twenty primes, followed by **`Done!`**

```
   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
 179 181 191 193 197 199 211 223 227 229 233 239 241 251 257 263 269 271 277 281
 283 293 307 311 313 317 331 337 347 349 353 359 367 373 379 383 389 397 401 409
 419 421 431 433 439 443 449 457 461 463 467 479 487 491 499 503 509 521 523 541
 547 557 563 569 571 577 587 593 599 601 607 613 617 619 631 641 643 647 653 659
 661 673 677 683 691 701 709 719 727 733 739 743 751 757 761 769 773 787 797 809
 811 821 823 827 829 839 853 857 859 863 877 881 883 887 907 911 919 929 937 941
 947 953 967 971 977 983 991 997

Done!
```

CodeCheck will verify that your program's results match this output exactly.

Submission into Canvas

When you're satisfied with your program in CodeCheck, click the "Download" link at the very bottom of the Report screen to download a signed zip file of your solution. Submit this zip file into Canvas. You can submit as many times as you want until the deadline, and the number of submissions will not affect your score. Only your last submission will be graded.

Submit the signed zip file from CodeCheck into Canvas:
Assignment 3.a. Prime Numbers

Note: You must submit the signed zip file that you download from CodeCheck, or your submission will not be graded. Do not rename the zip file.

Rubric

Your program will be graded according to these criteria:

Criteria	Maximum points
Good output (as determined by CodeCheck) • Correct output values. • Correct report format.	**6** • 3 • 3
Good program design • Good use of a one-dimensional array. • Array properly passed to functions **`compute_primes`** and **`print_primes`**. • Good implementation of the Sieve of Eratosthenes.	**18** • 5 • 5 • 8
Good program style • Descriptive variable names. • Meaningful comments. • Follow the coding style (formatting, braces, indentation, function declarations before the main, etc.) of the Savitch textbook.	**6** • 2 • 2 • 2

Academic integrity

You may study together and discuss the assignments, but what you turn in must be your individual work. Assignment submissions will be checked for plagiarism using Moss (http://theory.stanford.edu/~aiken/moss/). **Copying another student's program or sharing your program is a violation of academic integrity.** Moss is not fooled by renaming variables, reformatting source code, or re-ordering functions.

Violators of academic integrity will suffer severe sanctions, including academic probation. Students who are on academic probation are not eligible for work as instructional assistants in the university or for internships at local companies.