Programming Assignment 1

Note 1:

Please, submit your work via Webcourses.

Submissions by e-mail will not be accepted.

Due date: Friday, January 26th by 11:59 PM

Late submissions are not accepted.

Note 2:

This assignment is individual.

You can use a programming language of your choice for this assignment (limited to C++, Java, and Rust).

If you do not have a preference for a programming language, I would recommend C++.

Problem 1 (100 points)

Your non-technical manager assigns you the task to find all primes between 1 and 10^8. The assumption is that your company is going to use a parallel machine that supports eight concurrent threads. Thus, in your design you should plan to spawn 8 threads that will perform the necessary computation. Your boss does not have a strong technical background but she is a reasonable person. Therefore, she expects to see that the work is distributed such that the computational execution time is approximately equivalent among the threads. Finally, you need to provide a brief summary of your approach and an informal statement reasoning about the correctness and efficiency of your design. Provide a summary of the experimental evaluation of your approach. Remember, that your company cannot afford a supercomputer and rents a machine by the minute, so the longer your program takes, the more it costs. Feel free to use any programming language of your choice *that supports multi-threading* as long as you provide a ReadMe file with instructions for your manager explaining how to compile and run your program from the command prompt.

Required Output:

Please print the following output to a file named primes.txt:

<execution time> <total number of primes found> <sum of all primes found>

<top ten maximum primes, listed in order from lowest to highest>

Notes on Output:

1. Zero and one are neither prime nor composite, so please don't include them in the total number of primes found and the sum of all primes found.
2. The execution time should start prior to spawning the threads and end after all threads complete.

Grading policy:

General program design and correctness: 50%

Efficiency: 30%

Documentation including statements and proof of correctness, efficiency, and experimental evaluation: 20%

Additional Instructions:

Cheating in any form will not be tolerated. Please, submit your work via webcourses.

In addition to being parallel, your design should also make use of an efficient algorithm for finding prime numbers.

Submission:

You will submit a link to your GitHub page. Your repositories **must be private** until the next morning. **You must still push your code before the deadline, because Github will record this time. No code pushes will be accepted after the deadline.** However, we wont start grading until the next morning.

If we cannot access your repositories, or if you provide an invalid link, you will receive a 0 - *please double check your submission once it has been made*.

Late submissions will receive a 0 as per the syllabus.

This GitHub management does two good things:

1. Removes the temptation to look at other's work, because they will all be private until after the deadline.

2. Makes your life easier, as you don't have to send us invites

Also, we cannot accept any late work as all the repos will be potentially public shortly after the deadline.

Happy coding!