HW3: Pthread Interactions

Due Midnight 2/10/2020

For this assignment you will modify your code from HW2 to have the threads interact with each other and guarantee a specific number of random primes.

Specify a compile-time constant `NUM_VALUES` that indicates the total number of random primes that the team of threads should generate. Collect these values in a global array that's protected by a mutex.

The master thread should wait on a condition variable, and be woken up when `NUM_VALUES` is reached, at which point it should cancel the threads and print out values. Print out the total time taken for your program, as well as the number of threads used.

Rubric

Program compiles with no warnings	30
Code launches a configurable number of threads and collects results	30
Use of condition variable and mutex are correct	30
Code style is professional* and the submission contains the correct files	10

Please note: file names should not contain spaces. Code that fails to compile will not be graded.

Scalability testing

We can't predict scalability for this program because of the random element. Nevertheless, I want you to see how the times compare when running different numbers of threads. What can you say about the strong scaling of your program when run on 1 thread, 2 threads, one thread for each physical core on your system (not hyperthreads), and twice that amount.

Turn in . . .

A zip file containing your single source code file, a text file showing the contents of your Bash prompt, compiling and running the code (remember to use `-Wall` and `-O3` flags), and a text file describing the performance results described previously.