Multithreading

1. (5p) Running tasks in parallel.
   The file **predictors.txt** contains various information about participants of a trial, separated by commas. The last number in each row is the gender of the participants (1=male, 0=female). Write a program to read the gender data into a container. Write a function to count the number of rows corresponding to male participants in the container. Then write a function to count the number of male participants between given indices of the container, and use that function to launch at least two threads (or other asynchronous operations) to count the male participants. You don't have to return the number of participants to the main function, but you can instead just print the number to console.
2. (10p) Sharing data between threads.
   Write a program that has two "types" of threads: "producer" threads that write to a shared "buffer" container of strings and "consumer" threads that read the oldest string from the shared container, print the string, and remove the string from the container. All the threads should be able to operate concurrently (that is, the producers and consumers work at the same time) with the exception that the buffer container should only be accessed by one thread at a time. Test the program with at least 3 producer threads and 3 consumer threads (a total of 6 threads running at the same time).
   You may find some of or a combination of the following useful: std::condition_variable, std::mutex, std::atomic.