Complete the following and upload it using Canvas (online.utk.edu) by midnight on the due date. You should create a tar or zip file for your code, and it should include a makefile and documentation on usage. Don't include binaries; please just include the code.

Create a program in C or C++ using pthreads to implement a thread pool with a fixed number of threads. The thread pool should not create new threads when new work is submitted, and threads should remain in the pool after they complete their work (do not create new threads or delete them at the start/end of new work).

The thread pool should include a task queue for managing arriving tasks/work. (Each thread pool instantiated will have a set of threads for performing work along with an associated queue for managing arriving tasks/work. Hint: if using C++ the STL Queue can help here.)

Your thread pool should include a function/method that will allow you to `pause` execution of tasks/work (while allowing any threads currently executing tasks to complete first). You should provide another function/method that will allow the thread pool to `resume`, assuming that the thread pool previously did a `pause`.

You should also provide a function/method to `stop` the thread pool and delete the threads. For threads currently executing tasks/work, these should all finish their execution before deleting each thread.

Provide test code to exercise your thread pool, e.g., having test code with task functions for determining if an input integer is prime.