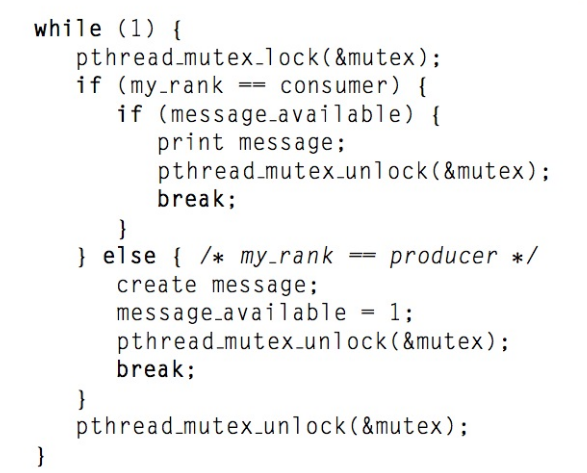
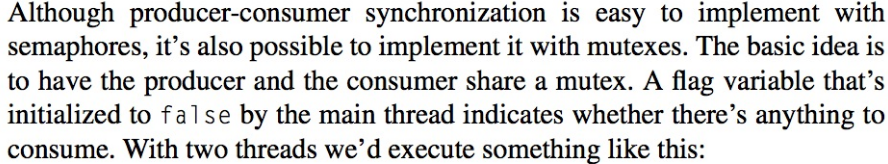
4.1. Modify the mutex version of the calculation program so that the critical section is in the for loop. Compare the performance of your version with the serial version.

4.2. Modify the mutex version of the calculation program so that it uses a semaphore instead of a mutex. Compare the performance of your version with the serial version.

4.3.



a. Write a Pthreads program that implements this version of producer-consumer synchronization with two threads.

b. Generalize this so that it works with 2*k* threads – odd-ranked threads are consumers and even-ranked threads are producers.

c. Generalize this so that each thread is both a producer and a consumer. For example, suppose that thread q “sends” a message to thread (*q* + 1) mod *t* and “receives” a message from thread (*q* − 1 + *t*) mod *t*, where *t* is the total number of threads.

Templates for 4.1 and 4.2 are provided, they both use timer.h to obtain the system time, and this head file is also provided.

**Note: All programs have to be presented to the instructor on the class of due date.**