6.1.

Write a Pthreads program that implements the trapezoidal rule. Use a shared variable for the sum of all the threads' computations. Use busy-waiting, mutexes, and semaphores to enforce mutual exclusion in the critical section.

6.2.

Use linked list operations, to write a Pthreads program that implements a "task queue". The main thread begins by generating a random sorted linked list *L*. The user inputs *a* -number of tasks, and *b* - number of threads then. A task can be member, delete, or insert operation on the list *L*. Each thread will do a number of tasks (each task is randomly choosing from member, delete, and insert) according to *a* and *b*. Your program must provide the reader-writer (RW) lock to protect *L* from incorrect status.