6.1.

Write a Pthreads program that implements the trapezoidal rule. Use a shared varaible for the sum of all the thread's computations, and choose one of busy-waiting, mutexes or semaphores to enforce mutual exclusion in its 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.

A task can be member, delete, or insert operation on the list *L*.

Each thread will do several 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.