In this assignment, you have to implement a concurrency control algorithm. The system consists of a transaction threads, concurrency control manager and queue objects. Each transaction thread (TT) should have the following structure:

repeat
    Select a transaction type randomly
    Select object(s) for transaction randomly
    Invoke transaction method.

The possible transactions are:
- Enqueue(Q, i): enqueue integer i into Q, where i > 0; if Q is full, then perform no action
- Dequeue(Q): returns the integer dequeued from Q; returns -1 if Q is empty
- Transfer(Q1,Q2): dequeue an item from Q1 and enqueue into Q2. If either Q1 is empty or Q2 is full, then the transaction should have no impact
- Max(Q1,Q2): reads the items at the front of both queues Q1 and Q2 (without dequeuing) and returns the maximum of the two. If both are empty, then it must return -1; if either one of them is empty, then it returns the items in the non-empty queue.

The queue object must implement the Enqueue, Dequeue and ReadFront operations. The queue object must contain no synchronization code. The Concurrency Control Manager (CCM) must implement the mechanism to implement the control access to the queues. At the minimum, it should include locking and unlocking operations. The Transaction object must implement the Enqueue, Dequeue, Transfer and Max transactions. Each transaction must invoke the necessary lock and unlock operations.

The goal is to study performance improvements based on the granularity of locking. The first version of the program must allow serial schedules (locking the entire database). You must then compare the performance to your second version which does two-phase locking. Finally, implement a third version which allows more fine-grained level (e.g., if Queuesize is greater than 1, then it may allow Enqueue and Dequeue at the same time). The performance must be measured in terms of transaction throughput (number of transactions completed over a specific interval of time). You must have a sleep statement inside each queue method to simulate time required to access the queue. You must plot the following data for each version: (a) Impact of number of TT threads on throughput, (b) Impact of the transaction mix on throughput.