Programming Assignment 3A Hints

To define a lock: pthread_mutex_t lock;
To initialize a lock: pthread_mutex_init(&lock, NULL);
To get a lock, (i.e., lock): pthread_mutex_lock(&lock);
To release a lock, (i.e., unlock): pthread_mutex_unlock(&lock);

To make Table2 a monitor:

```
pickup( int i ) {
        // lock immediately
        // your textbook logic
        // unlock at the very last moment.
}

putdown( int i ) {
        // lock immediately
        // your textbook logic
        // unlock at the very last moment.
}
```

To initialize self[i]: pthread_cond_init(&self[i], NULL);
To convert Textbook/slide's self[i].wait( ) into pthread:
        pthread_cond_wait(&self[i], &lock)
To convert Textbook/slide's self[i].signal( ) into pthread:
        pthread_cond_signal(&self[i])

Your ./a.out 0, a.out 1, and a.out 2's execution outputs do not have to be exactly the same as the homework specification.

Of importance is to show that a.out 1 guarantees at most one thread at a table, whereas a.out 2 accepts two threads at a table simultaneously, so that the execution performance will be almost two times faster than a.out.