1. The code would not work because every thread is running the same code. Therefore, it will not find a lock, and every thread will enter the critical section at the same time.
2. One way to do this would be to have two or more binary semaphores. Set semaphore 1 to 0 and semaphore 2 to 1. Set a count variable the to the initial value that you want the counting semaphore to have. Send a wait signal to semaphore 1. Set your counter to -1. If count is less than 1 then send a signal to semaphore 1 and tell semaphore 2 to wait. After that, wait on semaphore 1 and then increment the counter. If count is 0 then signal semaphore 2. Otherwise, signal semaphore 1.
3. One way to implement a barrier is to use a variable that records a pass and stop rate and another to that tracks how many threads are in the barrier. When a thread enters the barrier, it is put into a stop state. The threads that are trapped in the barrier constantly testing the state until the last thread arrives. After this, the barrier is released. This could be implemented as a linked list.