Worksheet: Bounded Blocking Concurrent List using Semaphores

Use the semaphore acquire() and release() to ensure that all threads are able to fairly access the BoundBlocking Concurrent List in addFirst() and remove().

```java
1. public class BoundedBlockingList {
2.   final int capacity;
3.   final ConcurrentLinkedLIst list = new ConcurrentLinkedLIst();
4.   final Semaphore sem;
5.   public BoundedBlockingList(int capacity) {
6.     this.capacity = capacity;
7.     sem = new Semaphore(capacity);
8.   }
9.   public void addFirst(Object x) throws InterruptedException {
10.      try { list.addFirst(x); }
11.      catch (Throwable t) { rethrow(t); } // only performed on exception
12.   }
13.   public boolean remove(Object x) {
14.      if (list.remove(x)) { return true; }
15.      return false;
16.   }
17. } // BoundedBlockingList
```