1. Why does locking via disabling interrupts not work on multi-processor architectures?

2. Write pseudo code to implement a “Too Much Milk” solution with two threads (robots) using the `test_and_set()` operation to implement mutual exclusion. Remember to define the `Lock::Acquire()` and `Lock::Release()` functions.

3. The following is a modified solution for the consumer-producer problem, with bounded buffer. Does the solution work? If so, please defend your position. If not, please show a sequence of operations that leads to a failure of the solution.

```cpp
semaphore nLoadedBuffers = 0;  // consumer waits on 0
semaphore nFreeBuffers = N;    // producer waits on 0
                             // N >= 2
semaphore mutex = 1;           // one thread waits when
                             // another thread is
                             // modifying the
                             // buffer

Producer() {
1. P(nFreeBuffers);
2. P(mutex);
3. // put 1 item in the buffer
4. V(nLoadedBuffers);
5. V(mutex);
}

Consumer() {
6. P(nLoadedBuffers);
7. P(mutex);
8. // take 1 item from the buffer
9. V(mutex);
10. V(nFreeBuffers);
}
```