York University COSC6117 January 8, 2008

Homework Exercise #1
Due: January 22, 2008

1. Recall Dijkstra's mutual exclusion algorithm. Suppose we want to solve the same problem, but now instead of just read/write registers, we can use a stronger type of shared object in the memory: a peekable queue. The peekable queue stores a sequence of values and provides three atomic operations:

- ENQUEUE(x): adds x to the end of the sequence and returns *ack*.
- PEEK: returns the first element in the sequence without changing the sequence.
- DEQUEUE: removes and returns the first element of the sequence. (If the queue is empty when this operation is done, the operation just returns *empty* instead.)

(a) Give a **simple** mutual exclusion algorithm that uses a single queue.

(b) Prove that it satisfies the exclusion and progress properties.

(c) Can processes starve in your algorithm? If so, give an execution where somebody starves. If not, explain why not.