Exercise 1. ***Wait-free implies lock free***

Because if there was a lock, it would be possible for one thread to starve (by never succeeding to acquire the lock). Thus, it couldn't be wait free. Therefore wait-free implies lock-free.

Exercise 2. ***Consensus among prisoners***

We divide the block days into blocks of 100. We propose the following strategy:

- If it's the first day of a 100-day block
  - If the light bulb is turned off, turn it on.
  - If the light bulb is turned on, state "by now every prisoner was in the room at least once"
- On any other day:
  - If we are in the room for the fist time during the current 100-day block, do nothing.
  - If we are in the room for the second time during the current 100-day block, turn the light bulb off.

Eventually we will have a 100-day block where each prisoner is in the room just once. In that case, the light bulb will be turend on at the beginning of the next 100-day block.

Our strategy is neither wait-free, nor non-blocking.

Non-blocking: No two prisoners can be in the room at the same time. **Wait-free**: In theory it is possible that one prisoner will have to wait forever. It is thus not wait-free.

Exercise 3. ***Implementing two thread consensus***

Algorithm 1 consensus

procedure CONSENSUS(val)
  $result[mythreadid] = val$
  if $TAS() == 1$ **then return** $result[mythreadid]$
  return $result[otherthreadid]$

Exercise 4. ***Linearizability***

(a) The history is invalid. Thread A calls s.push(1), but gets void only after s.pop().

(b) The history is linearizable. s.pop() are the linearization points for both threads.

(c) The history is not linearizable, but it is sequentially constistent.