**Problem 1.** You are given the algorithm in Figure 1 for constructing a single-reader single-writer (SRSW) $M$-valued atomic register using single-reader single-writer (SRSW) Boolean atomic registers. Does this proposal work? Either prove the correctness or present a counterexample.

**Problem 2.** Does Peterson’s two-thread mutual exclusion algorithm work if the shared atomic flag registers are replaced by regular registers? Explain.

**Problem 3.** Show that with sufficiently many $n$-thread binary consensus objects and atomic registers, one can implement $n$-thread consensus over $n$ values.

**Problem 4.** An A2Cas object represents two locations for values that can be read individually and be modified by an atomically executing method a2cas(). If both locations have the corresponding expected values $e_0$ and $e_1$, then a call to a2cas$(e_0, e_1, v)$ will write $v$ to **exactly one** of the two locations, chosen nondeterministically.

What is the consensus number of the a2cas() object? Prove your claim.
public class AtomicSRSWRegister implements Register<int> {
    private static int RANGE = M;
    boolean[] r_bit = new boolean[RANGE]; // atomic boolean SRSW
    public AtomicSRSWRegister(int capacity) {
        for (int i = 1; i <= RANGE; i++)
            r_bit[i] = false;
        r_bit[0] = true;
    }
    public void write(int x) {
        r_bit[x] = true;
        for (int i = x - 1; i >= 0; i--)
            r_bit[i] = false;
    }
    public int read() {
        for (int i = 0; i <= RANGE; i++)
            if (r_bit[i])
                return i;
        return -1; // impossible
    }
}

Figure 1: Boolean to $M$-valued SRSW Atomic Register Algorithm