1. A stack object supports two types of operations:

**Push**  Inserts a value to the head of the object;

**Pop**  Removes a value from the head and returns it, or returns an empty indication if the object is empty.

(a) Give a two-process wait-free consensus algorithm using *only* stack objects. Prove your answer.

(b) Show that the consensus problem for three processes cannot be wait-free solved in an asynchronous system using only stack objects and read/write objects.

2. Consider a collection of atomic read/write registers with additional *move* operation. A move operation atomically copies the value of one shared register to another shared register. We use the expression $A \leftarrow B$ to denote a move operation copying the contents of $B$ to $A$. Algorithm 1 presents an n-process consensus protocol using the move operation. Prove its correctness.

**Algorithm 1** An n-process consensus protocol using the move operation. code for $p_i$.

1. CONSENSUS($x_i$)
   
   global array $Pref[1..n]$ initially ⊥.
   global array $R[1..n, 1..2]$ where $R[i, 1]$ initially 1 and $R[i, 2]$ initially 0.
2. $Pref[i] := x_i$
3. $R[i, 2] \leftarrow R[i, 1]$
4. **for** $j = i + 1$ **to** $n$ **do**
5.   $R[j, 1] := 0$
6. **for** $j = n$ **down to** 1 **do**
7.   if $R[j, 2] = 1$ then return $Pref[j]$
Algorithm 2 A polynomial consensus algorithm in the presence of Byzantine failures

Initially $\text{pref}[i] = x$ \quad $\triangleright$ initial preference is input
Initially $\text{pref}[j] = v_\perp$, for any $j \neq i$ \quad $\triangleright$ default for others

1: round $2k - 1$, $1 \leq k \leq f + 1$: \quad $\triangleright$ first round of phase $k$
2: send \langle $\text{pref}[i]$ \rangle to all processors
3: receive \langle $v_j$ \rangle from $p_j$ and assign to $\text{pref}[j]$, for all $0 \leq j \leq n - 1$, $j \neq i$
4: let $maj$ be the majority value of $\text{pref}[0], \ldots, \text{pref}[n-1]$ ($v_\perp$ if none)
5: let $mult$ be the multiplicity of $maj$

6: round $2k$, $1 \leq k \leq f + 1$: \quad $\triangleright$ second round of phase $k$
7: if $i = k$ then send \langle $maj$ \rangle to all processors \quad $\triangleright$ king of this phase
8: receive \langle $king - maj$ \rangle from $p_k$ ($v_\perp$ if none)
9: if $mult > n/2 + f$
10: \quad then $\text{pref}[i] = maj$
11: else $\text{pref}[i] = king - maj$
12: if $k = f + 1$ then $y = \text{pref}[i]$

3. Consider the Byzantine consensus algorithm with constant size messages (Algorithm 2).

(a) Show that for an input set $\{0, 1, 2\}$ the algorithm does not satisfy the strong validity condition (the output must be the input of some non-faulty process).

(b) Modify the algorithm so that the strong validity condition holds, for $n > 6f$. Prove your answer. Hint: add another condition in Line 9 that will verify that a value can be adopted from the king only if it is the input of a non-faulty process. This might also require saving some more information from the first round in the phase.