Exercise 1
(Consensus & Perfect failure detector)

Consider our fail-stop consensus algorithms (Consensus Algorithm I and Consensus Algorithm II). Explain why none of those algorithms would be correct if the failure detector turned out not to be perfect.
Exercise 2  
(Consensus & Eventually perfect failure detector)

*Explain why any fail-noisy consensus algorithm (one that uses an eventually perfect failure detector P) actually solves uniform consensus (and not only the non-uniform variant).*
Exercise 3
(Consensus & Correct majority)

Explain why any fail-noisy consensus algorithm (one that uses an eventually perfect failure detector $\Diamond P$) requires a majority of the processes to be correct. More precisely, provide a “bad run” in the case where the majority of processes is faulty.
Sequential Objects

A sequential object is a tuple $T = (Q, q_0, O, R, \Delta)$, where:

- $Q$ is a set of states.
- $q_0 \in Q$ is an initial state.
- $O$ is a set of operations.
- $R$ is a set of responses.
- $\Delta \subseteq (Q \times \mathcal{P} \times O) \times (Q \times R)$ is a relation that associates a state, a process, and an operation to a set of possible new states and responses.

Processes invoke operations on the object. As a result, they get responses back, and the state of the object is updated to a new value, following from $\Delta$. 
Define a sequential object representing Asset Transfer, i.e., an object that allows processes to exchange units of currency.
Bonus Exercise 5
(Total Order & Asset Transfer)

Use Total Order Broadcast to implement an Asset Transfer sequential object.