Question 2 (10 points)

An algorithm that solves the 2-Generals Problem must satisfy the following three properties:

- **Agreement**: No two processes decide on different values.
- **Validity**:
  1. If all processes have input 0, then 0 is the only decision value.
  2. If all processes have input 1 and all messages are received, then 1 is the only decision value.
- **Termination**: All processes decide after a finite number of messages.

In class, we learned that there is no algorithm that solves the 2-Generals Problem in a system with unreliable messaging (and no process failures).

Consider a different messaging model where both generals know exactly how many messages will be lost, in total, but they don’t know which messages will be lost. Is it possible to solve the 2-Generals Problem under this system model? Answer YES or NO. If your answer is YES, give a solution and explain why your algorithm solves the problem. If your answer is NO, explain why not.

Question 3 (10 points)

A solution to the Atomic Commit Problem must satisfy the following three properties:

- **Agreement**: No two processes decide on different values.
- **Validity**:
  1. If any process votes *abort*, then *abort* is the only decision value.
  2. If all processes vote *commit* and there are no failures, then *commit* is the only decision value.
- **Termination**: All non-failed processes eventually decide.

We studied the Two Phase Commit algorithm (2PC) for this problem in a synchronous system model with reliable messaging and crash failures (but no recovery). In this model, if a process does not receive an expected message by a “timeout”, it can conclude that the sender has failed. We showed that 2PC guarantees agreement and validity in this system model, but not termination.

Consider an execution of the 2PC algorithm (in the same system model) in which the timeout is accidentally set to be too small. Thus, a process may not receive a message within the timeout and conclude that the sender has failed. However, it may be that the sender has not failed, but the message took too long. Assume that any message received after its corresponding timeout is ignored.

For each question below, answer YES or NO and justify your answer.

1. Does 2PC guarantee safety if the timeout is too small?
2. Does 2PC guarantee validity if the timeout is too small?

Question 4 (10 points)

The Paxons want to pass a decree to determine what type of candy to hand out on Halloween. To reach consensus, they decide to use the Synod algorithm with 5 acceptors. Suppose proposer $p$’s proposed value is ‘Butterfinger’. $p$ sends a prepare message with proposal number 6, and it receives a set $S$ of responses (promise messages), where each response is of the form $(accNum, accVal)$.

For each set $S$ below, is this set one that $p$ could receive if all acceptors follow the Synod algorithm? If the answer is NO, explain why the set $S$ could not have been received by $p$.

If the answer is YES, describe a correct execution of the Synod algorithm in which $p$ receives the promise messages corresponding to the set $S$. Also, state which proposal is chosen (when $p$ receives $S$) in your execution, if any.

1. $S = \{(1, "Skittles"), (2, "Skittles"), (null, null)\}$
2. $S = \{(3, "Raisins"), (null, null), (7, "Snickers")\}$
3. $S = \{(1, "Twix"), (3, "Twizzlers"), (5, "Starburst")\}$