1. **[15 points]** Suppose that the full Paxos algorithm (with the leader optimization) is used to maintain a replicated system with a single object $x$. The system consists of a set of clients, a set of 3 acceptors, and a set of 3 learners. The clients, acceptors, and learners each operate on different processes (machines). Each learner stores a copy of the object $x$.

   To perform a write operation, a client acts as a proposer and sends the write operation to the distinguished proposer (one of the acceptors). The acceptors then perform the part of the Paxos algorithm that allows them to reach consensus on whether to accept the write operation. If consensus is reached, the distinguished proposer sends commit messages to the learners, who record the write operation in their logs and apply the write operation to their local copies of $x$. The write operation is considered completed after the leader sends the commit messages. Assume the client retries its proposal until the write is successful.

   To perform a read operation, a client sends a read request to one of the learners. The learner returns the value of its local copy of $x$.

   Answer each question below, and justify your answer (give a counter-example, if applicable). Assume the system model is the same as studied in class for the Paxos algorithm.

   (a) Does this system implement sequential consistency?

   (b) Does this system implement eventual consistency?

2. **[15 points]** A range query is a database operation that returns all data objects whose value is contained in an interval with a specified lower and upper bound. An example of a range query is range($A, C$), which should return all objects with names that start with the letter ‘A’, the letter ‘B’, or the letter ‘C’.

   Consider a distributed storage system based on the Chord DHT, i.e., the object locations are determined by the Chord lookup function.

   (a) Devise an algorithm for executing range queries on top of the Chord DHT. Your algorithm can call the lookup function as many times as needed.

   (b) What is the message complexity of a single range query in your algorithm?

   (c) Is your implementation guaranteed to retrieve all files in the query range even when there is churn? If so, explain why. If not, describe an example execution where the response to a query does not contain all files in the specified range.