1. **[15 points]** Consider a distributed system with 3 processes, where events are timestamped using one of the three methods we studied in class.

   (a) (Lamport timestamps): Suppose event $a$ has timestamp 3 and event $b$ has timestamp 2. Which one of the following four options apply? Justify your answer.

   i) Event $a$ happens before event $b$.

   ii) Event $b$ happens before event $a$.

   iii) Events $a$ and $b$ are concurrent.

   iv) The timestamps do not provide enough information to conclude which of the above options (i, ii, or iii) apply.

   (b) (Vector timestamps): Event $c$ has timestamp (3,2,0) and event $d$ has timestamp (1,2,1). Which one of the following four options apply? Justify your answer.

   i) Event $c$ happens before event $d$.

   ii) Event $d$ happens before event $c$.

   iii) Events $c$ and $d$ are concurrent.

   iv) The timestamps do not provide enough information to conclude which of the above options (i, ii, or iii) apply.

   (c) (Matrix timestamps): Event $e$ has timestamp \[
   \begin{bmatrix}
   1 & 1 & 1 \\
   1 & 2 & 1 \\
   0 & 0 & 3
   \end{bmatrix}
   \] and event $f$ has timestamp \[
   \begin{bmatrix}
   1 & 1 & 1 \\
   1 & 3 & 1 \\
   1 & 1 & 3
   \end{bmatrix}
   \]

   Corrected timestamps: Event $e$ has timestamp \[
   \begin{bmatrix}
   1 & 1 & 1 \\
   1 & 2 & 3 \\
   0 & 0 & 3
   \end{bmatrix}
   \] and event $f$ has timestamp \[
   \begin{bmatrix}
   1 & 1 & 1 \\
   1 & 3 & 3 \\
   1 & 1 & 3
   \end{bmatrix}
   \]

   Which one of the following four options apply? Justify your answer.

   i) Event $e$ happens before event $f$.

   ii) Event $f$ happens before event $e$.

   iii) Events $e$ and $f$ are concurrent.

   iv) The timestamps do not provide enough information to conclude which of the above options (i, ii, or iii) apply.

2. **[10 points]** Consider an asynchronous distributed system with three processes $p_1$, $p_2$, and $p_3$. The network is fully connected (every process can send messages to every other process), messaging is reliable, and all communication channels are FIFO. The processes use Lamport’s mutual exclusion algorithm to grant access to a resource. Suppose that $p_1$ requests the resource and is granted access. While $p_1$ is accessing the resource, $p_2$ requests access to the resource, and then shortly after that, $p_3$ requests access to the resource.

After waiting a while, $p_3$ wants to check on its request for the resource, so it initiates the Chandy-Lamport global snapshot algorithm to learn the state of the system, i.e., the contents of the request queue at each process and which process is accessing the resource, if any. At the same time, $p_1$ releases the resource and notifies the other processes, triggering additional steps in the mutual exclusion algorithm. Describe the execution of the Chandy-Lamport algorithm in this setting. What messages are exchanged, and what process and channel states are recorded?
3. **[10 points]** For each of the following statements, either argue why it is correct or give a counter-example. Use space-time diagrams to illustrate your examples where appropriate.

   (a) Every Atomic Broadcast algorithm ensures the causal order property.
   
   (b) Every Causal Reliable Broadcast algorithm ensures the FIFO order property.

4. **[10 points]** Consider the Byzantine Agreement Algorithm (Oral Messages Algorithm) in a system with 1 commander and 6 lieutenants, where $m = 2$, and where there are at most two faulty processes. Assume the system model is the same one studied in class. What is the maximum number of messages that may be sent in a single execution of the algorithm? Justify your answer.

5. **[10 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 leader (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. After consensus is reached, the leader 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.

   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?

6. **[10 points]** An algorithm that solves the Two Generals Problem must satisfy the following properties:

   - **Agreement:** No two processes decide on different values.
   - **Validity:** If all processes start with 0 (RETREAT), then 0 is the only possible decision value. If all processes start with 1 (ATTACK) and all messages are delivered, then 1 is the only possible decision value.
   - **Termination:** All processes eventually decide.

   In class, we learned that in a system with two nodes (generals) that never fail, but where messages may be lost, there is no algorithm that solves the Two Generals Problem. Suppose that we strengthen the communication model so that it is guaranteed that either all messages will be delivered or all messages will be lost, and both generals know this guarantee exists. Further, we assume the system is synchronous so missing messages can be detected. Is it possible to solve the Two Generals Problem under this model? If so, give an algorithm that solves the problem. If not, explain why not?

7. **[10 points]** Consider Amazon Dynamo’s eventually consistent replication scheme. Let the replication factor be $k = 3$, i.e., each object is replicated at three nodes. Let the read quorum and write quorum sizes each be 2. Give an example execution of reads (gets) and writes (puts) on a single object that is not sequentially consistent. Explain which replicas are accessed in each operation.