Problem 1: (10 points): For physical clock synchronization, two protocols are presented in the book: NTP and Berkeley algorithm (pages 240-241). Briefly outline the differences between these two approaches.

Problem 2: (20 points): The state machine model of total-ordering of all operations in a group of replicated processes is presented in the book (pages 247-248). In this model each process executes the same sequence of operations. The textbook describes an approach based on logical clocks for implementing this model. Outline a scheme for total-ordering of operations based on physical clock values assuming that the maximum skew between any two processes’ clocks is no more than $\Delta$ and the maximum delay in any point-to-point message communication is bounded by $d$. A process will multicast its operation request to all others with a timestamp equal to the local physical clock value at the time of multicasting.
- Give the rules for imposing a unique total order on the requests at processes in the system.
- When can a process safely execute the operation at the head of its local request queue?

Problem 3: (20 points): In Figure 6.21 there are two ELECTION messages circulating simultaneously. It would be more efficient to suppress one of these messages. Describe an approach for doing this without affecting the correctness of the leader election protocol.

Problem 4: (15 points): Given are vector clock values $T_a$ and $T_b$ for two events $a$ and $b$, respectively. For each of the following cases indicate whether the happened-before relationship holds between these two events or if the events are concurrent. In this example the vector clock size is three.
1. $T_a = <5, 3, 2>$ and $T_b = <4, 4, 1>$
2. $T_a = <4, 3, 2>$ and $T_b = <4, 4, 3>$
3. $T_a = <5, 3, 2>$ and $T_b = <4, 3, 1>$
4. $T_a = <5, 3, 2>$ and $T_b = <4, 3, 3>$
5. $T_a = <3, 5, 2>$ and $T_b = <4, 4, 1>$

Problem 5: (15 points): As discussed in the textbook, Ricart and Agrawala’s algorithm for mutual exclusion is prone to process failures. If a requesting process sends a request message to a process that has crashed, it will wait for forever. The book suggests a scheme in which all requests are immediately answered by a process with a reply as either grant to deny (see page 257). This makes it easier to detect a crashed process. On receiving a deny reply, the requester waits for a grant from that process. Is this method sufficient to deal with process crashes? Explain your answer.

Problem 6: (20 points): What is the purpose of the INQUIRE messages in the mutual exclusion protocol by Maekawa? What actions are taken by a process receiving a message of this kind?