Problem 1. Consider an anonymous system consisting of $N$ processes $0 \ldots N-1$. The topology is a completely connected network, and the links are bidirectional. Propose an algorithm using which processes can assign unique identifiers to themselves. Thus you are asked to convert an anonymous system into a named one. Justify that your algorithm will not lead to the duplication of names. (Hint: use coin flipping to break symmetry, and organize the computation in rounds.)

Problem 2. The CSP language proposed by Hoare [H78] uses a form of synchronous message passing: a process sending a message is delayed or blocked until the receiver is ready to receive the message, and vice versa. The symbols $!$ and $?$ are used to designate output and input actions respectively. To output a value $e$ to a process $Q$, the sending process $P$ executes the statement $Q! e$, and to input this value and assign it to a local variable $x$, process $Q$ executes the action $P? x$. Write a program with three processes $P$, $Q$, move, so that process move receives the values of an array $x$ from $P$ one after another and sends them to $Q$, which assigns these values to a local array $y$.

Problem 3. Alice and Bob enter into an agreement: whenever one falls sick, he will call the other person. Since making the agreement, no one called the other person, so both concluded that they are in good health. Assume that the clocks are synchronized, communication links are perfect, and a telephone call requires zero time to reach. What kind of interprocess communication model is this?
**Problem 4.** In a distributed system, processes are called *synchronous*, when they take actions are in lock step synchrony. As an example of synchronous behavior, consider the problem of *phase synchronization*, where a set of processes execute their actions in phases, and no process is allowed to execute phase $k+1$ unless every process have completed phase $k$, $k \geq 0$. Sometimes this is useful in parallelizing loop computations. Implement phase synchronization using the shared memory model. In your program, you can block and unblock a process using semaphores or any other kinds of primitives that you are familiar with.

**Problem 5.** In a network of mobile nodes, each node supports wireless broadcast only. These broadcasts have a limited range, so every node may not receive all the messages, and thus the network may not be connected. How will a node $P$ find out if it can directly or indirectly communicate with another node $Q$? What kind of computation models are you using for your solution? List all relevant properties.