There are five questions. Each question is worth 10 points

Q1. A distributed system is charged with the responsibility of deciding whether a given integer $N$ is a prime number. The system has a fixed number of processes, initially, only a designated process called the initiator knows $N$, and the final answer must be available to the initiator. Informally describe (1) how many processes will you use, (2) what each process will do, and (3) what inter-process messages will be exchanged.

Q2. 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 send a value $e$ to a process $Q$, the sending process $P$ executes the statement $Q! e$, and to receive this value from $P$ 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 $x$ of size $N$ from $P$ one after another and sends them to $Q$, which assigns these values to a local array $y$.

Q3. Consider an anonymous distributed system consisting of $N$ processes. The topology is a completely connected network, and the links are bidirectional. Propose an algorithm using which processes can acquire unique identifiers. (Hint: use coin flipping, and organize the computation in rounds). Justify why your algorithm will work.
Q4. Consider an array of \( n \) (\( n > 3 \)) processes. Starting from a terminal process, mark the processes alternately as even and odd. Assume that the even processes have states \( \in \{0, 2\} \), and the odd processes have states \( \in \{1, 3\} \). The system uses the state-reading model, and distributed scheduling of actions. From an unknown starting state, each process executes the following program:

\[
\text{program} \quad \text{alternator \{for process } i \}\}
\]
\[
\text{define} \quad s \in \{0, 1, 2, 3\}. \{\text{state of a process}\}
\]

\[
\text{do \forall j \in \text{Neighbor } (i): s[j] = s[i] + 1 \mod 4 \rightarrow s[i] := s[i] + 2 \mod 4 \text{ od}
\]

(a) Observe and summarize the steady state behavior of the above system of processes.

(b) What is the maximum number of processes that can eventually execute their actions concurrently in the steady state? Show your analysis.

Q5. There are two processes \( P \) and \( Q \). \( P \) has a set of integers \( A \), and \( Q \) has another set of integers \( B \). Using message passing model, develop a program by which processes \( P \) and \( Q \) exchange integers, so that eventually every element of \( A \) is greater than every element of \( B \).

Using a variant function, prove that the computation terminates in the desired goal state.