Computer Science 2221	**Homework 1**
University of Toronto	**Due: Friday, September 26, 2002**

This problem set is about the *Coordinated Attack* problem. Every process starts with an initial input of 0 or 1. Each process can output a decision value of 0 or 1 (once a process decides, it cannot "change its mind"). Processes must satisfy the following properties:

- *Uniform Agreement:* No two processes decide differently.
- *Validity:*
  (a) If all processes start with 0, then no process decides 1.
  (b) If all processes start with 1, and there are no failures, then no process decides 0.
- *Termination:* All nonfaulty processes eventually decide.

We consider the above problem in a completely connected network, where processes proceed in synchronous rounds. Processes do not fail, but any number of messages can be lost. Since all processes are correct, in this context Termination is simply:

- *Termination:* All processes eventually decide.

In class we proved that the Coordinated Attack problem cannot be solved in such a system consisting of two processes. In the following questions consider only systems consisting of two processes p and q.

1. Consider the following validity requirement:

   - *Strong Validity:*
     (a) If all processes start with 0, then no process decides 1.
     (b) If all processes start with 1, then no process decides 0.

   Give a *simple* proof that the problem consisting of *Agreement*, *Strong Validity*, and *Termination* cannot be solved. Do not use the result that we gave in class (your proof must be "from scratch").

2. Consider the following termination requirement:

   - *Weak Termination:* If there are no failures, then all processes eventually decide.

   Is the problem consisting of *Agreement*, *Validity*, and *Weak Termination* unsolvable? If yes, then give an impossibility proof; otherwise give an algorithm that solves it.

3. Consider the following requirement:

   - *Unanimous Termination:* If any process decides, then all processes eventually decide.

   Is the problem consisting of *Agreement*, *Validity*, *Weak Termination*, and *Unanimous Termination* unsolvable? If yes, then give an impossibility proof; otherwise give an algorithm that solves it.

4. In the previous questions, there were no restrictions on message losses. We now assume that the lossy link connecting p and q is *fair*: informally, every message sent repeatedly by one process is eventually received by the other process. So messages can be lost, but if a message is retransmitted repeatedly, then eventually at least one copy reaches its destination.

   (a) An algorithm is called *quiescent* if eventually no messages are sent. Give a quiescent algorithm that solves the original Coordinated Attack problem (i.e., *Agreement*, *Validity*, and *Termination*).

   (b) Explain where the impossibility proof given in class breaks down when we assume that lossy links are *fair*.

   (c) An algorithm is called *halting* if eventually all processes reach a "halting" state from which they cannot execute any steps. Obviously, a halting algorithm is quiescent. The converse is not necessarily true. A quiescent algorithm can reach a state where each process is idle but not halted: The process is ready to respond to a message if it ever gets one, but no messages are in transit.

   Assuming links are fair, give a halting algorithm for the Coordinated Attack problem or prove that none exists.