**22C: 166 Distributed Systems and Algorithms**

**Homework 5**

**Total points = 50**

Assigned 11/13/07, due 11/27/07 in class

*Please submit typewritten solutions.*

**Question 1.** (20 points) A sender process $P$ sends a sequence of messages to a receiver process $Q$. Each message $m$ is stamped with a sequence number $seq$ that increases monotonically. The program for $P$ can be specified as follows:

```plaintext
define seq : integer
initially seq = 0

do true
  send m[seq] to Q; seq := seq + 1
od
```

In the absence of failures, $Q$ receives the messages in the same order in which they are sent. Channel failures may cause messages to reach $Q$ out of order, but messages are never lost, and it is important that $Q$ accepts the message in the same order in which they were transmitted.

(a) Describe the program for $Q$. Calculate its buffer requirement.

(b) Now assume that $Q$ has a buffer that can hold at most one message. Rewrite the programs of $P$ and $Q$, so that $Q$ accepts the messages in the same order in which $P$ sent them. Argue why your solution will work.

**Question 2.** (10 points) The Byzantine generals algorithm helps reach a consensus when less than one-third of the processes undergo Byzantine failure. However, it does not suggest how to diagnose such failures. Assuming that at most one process can be faulty, examine if the faulty process can be identified without any ambiguity. Investigate all relevant cases.
**Question 3** (10 points) Chapter 13, exercise 3

**Question 4.** (10 points) In a spanning tree of a graph, there is exactly one path between any pair of nodes. If a spanning tree is used for broadcasting a message, and a process crashes, some nodes will not be able to receive the broadcast. Our goal is to improve the connectivity of the subgraph used for broadcast, so that it can tolerate the crash of one process.

What kind of minimal subgraph will you use for broadcasting, so that messages will reach every process even if one process fails? In the grid topology, identify such a minimal subgraph.