Question 1. (10 points) A sender process $P$ sends a sequence of messages to a receiver process $Q$. Each message $m$ is stamped with a bounded sequence number $seq$. If channels can reorder messages, then is it possible to design a protocol for FIFO message delivery that will only tolerate message loss? Do not worry if the receiver accepts duplicate copies of the same message. Clearly explain your answer.

Question 2. (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. 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 3.** (20 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 would you use for broadcasting, so that messages will reach every process even if one process fails? Suggest a distributed algorithm for constructing such a subgraph.

**Extra credit questions** (You can earn up to 30 extra points)

**Question 4.** (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 possible cases.

**Question 5.** (20 points) Present an example to show that in a synchronous distributed system consisting of six processes and at most two faulty processes, consensus may not be reached using OM(2) algorithm. You must clearly draw a tree to show the exchange of messages at different levels, Assume that the domain of the input and output values is {0,1}. 