Distributed Systems 20/07/2017
Corso di Laurea Magistrale in Ingegneria Informatica

☐ **5 Credits**   ☐ **6 out of 12 Credits (not passed CNS yet)**

☐ **6 Credits**   ☐ **6 out of 12 Credits (passed CNS)**

(tick the appropriate box above – write clear below)

Family Name_____Name_____Student ID_____

Ex 1: Discuss why passing from a synchronous system to an eventually synchronous one, it is no longer possible for failure detectors to ensure the property of strong accuracy while it is possible to ensure the property of eventual strong accuracy.

Ex 2: Describe the basic approaches presented in the course for information dissemination inside *publish/subscribe systems* discussing characteristics and limitations (e.g. "*the approach X is the best option when ……. because…….., while it is the worst option when ……because……*").

Ex 3: Consider the run depicted in the figure:
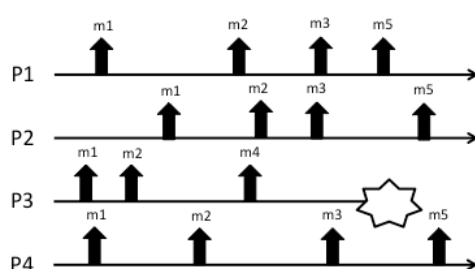
1. Which type of total ordering is satisfied by the run? Specify both the agreement and the ordering properties.
2. Modify the run in order to satisfy TO(UA, WUTO) but not TO (UA SUTO)
3. Modify the run in order to satisfy TO(NUA, WNUTO) but not TO(NUA, WUTO)

Ex 4: Let us consider a reliable broadcast primitive. Explain how the algorithm implementing the primitive must be changed while moving from the assumption of crash failures to the assumption of Byzantine failures. Discuss, in particular, the relation between the number of processes in the system n and the number of failures f.

Ex 5: Consider a distributed system composed by n processes $\{p_1,p_2,…,p_n\}$ that communicate by exchanging messages on top of a line topology, where p_1 and p_n are respectively the first and the last process of the network.
Initially, each process knows only its left neighbour and its right neighbour (if they exist) and stores the respective identifiers in two local variables LEFT and RIGHT.

Processes may fail by crashing, but they are equipped with a perfect failure detection system that notifies at each process the new neighbour (when one of the two fails) through the following primitives:

- **Left_neighbour(p$_x$):** process p$_x$ is the new left neighbour of p$_i$
- **Right_neighbour(p$_x$):** process p$_x$ is the new right neighbour of p$_i$

Both the events may return a NULL value in case p$_i$ becomes the first or the last process of the line. Each process can communicate only with its neighbours.

Write the pseudo-code of an algorithm implementing a FIFO Reliable Broadcast primitive assuming that channels connecting two neighbour processes are fair-loss links.