First Homework of Concurrent Systems

Exercise 1 Consider Peterson algorithm for n (> 2) processes, Fig. 2.8 of Rynal's book. Fix a suitable value of n and show an execution of the algorithm where no upper bound can be given on the number of bypasses that process p_1 can undergo.

Exercise 2 Consider the *conc_abort* primitive of Fig. 2.12 of Rynal's book and assume that there are only two processes, p_1 and p_2, with p_1 that receives *commit*. Then, prove that p_2 can never receive *commit*.

Exercise 3 Consider the *conc_abort* primitive of Fig. 2.12 of Rynal's book and assume that there are only two processes, p_1 and p_2. Is it true that in every possible run either p_1 or p_2 will receive *commit*? If yes, provide a formal proof; if not, provide a counterexample.

Exercise 4 Consider the following simplification of the *lock* primitive in Lamport's Fast MUTEX algorithm:

(1) $FLAG[i] \leftarrow up$
(2) $X \leftarrow i$
(3) **if** $Y \neq \perp$
(3) **then** $FLAG[i] \leftarrow down$
(4)   **wait** $Y = \perp$
(5)   **goto** (1)
(6) **else** $Y \leftarrow i$
(7)   **if** $X = i$
(8)   **then return**
(9)   **else** $FLAG[i] \leftarrow down$
(10)   **wait** $Y = \perp$
(11)   **goto** (1)

Show that it is not deadlock-free.