Assignment 01 — 18.09.2019 – v1.0
Introduction to Concurrency

Exercise 1 (7 points)
Answer the following questions (1 point each):

a) Do recent central processing units (CPUs) of desktop PCs support concurrency? Why became concurrency for many software applications very important these days?
b) Why do we need synchronization mechanisms in concurrent programs?
c) What is safety? Give one concrete example of a safety violation.
d) What is liveness? Give a concrete example of a liveness violation.
e) Can a binary semaphore lead to a deadlock? Why? Can it lead to starvation? Why?
f) How do monitors differ from semaphores? Please provide a precise answer.
g) How are monitors and message passing similar? How are they different?

Exercise 2 (2 points)

\[ x := 1 \]
Thread 1 \(-\rightarrow x := x + 7.\]
Thread 2 \(-\rightarrow x := x * 5.\]

Considering the code above: Give all possible values of \(x\) at the end of the execution of both threads together with their corresponding execution traces.

Hint: You should be able to perceive 6 different execution flows in total, however, some of them could lead to the same \(x\).

Exercise 3 (1 point)
Implement a monitor using a binary semaphore, i.e., a semaphore that can contain at most one token. Use pseudo-code and comment it.