In this exercise we familiarize ourselves to threads.

Exercise 13 (More about thread synchronization, 1p)

Phase 1. (Demonstrate the problem)

Download the program counter_several_threads_pohja.c from the Tuubi. In the program five threads increment the common counter until the value becomes 5 000 000. The original requirement was that it must be guaranteed that no thread never can increment the counter so that it exceeds the upper limit 5 000 000. The second requirement was that when the main thread returns from the wait loop, the counter value must be exactly the upper limit. Neither requirement is met as you can see when you run the program. The program also uses busy loop to wait until counter becomes, what was required. This means wasting processor resources.

Fix the problems of this program. The both outputs at the end of the program must be 5 000 000 and the program cannot use busy loops or delays for the synchronization.

Exercise 13 (Extra, more about thread synchronization, 0.25p)

Write a program which has the main thread and one sub thread. The main thread displays continuously digits 1 and the sub thread displays continuously digits 2 to the screen in the synchronous way so that the digits alternate in the display (they appear one after another). The output looks as follows

1212121212...

It is not allowed to use delays or busy loops in the program. The output starts with the digit 1 and is finished when both threads have displayed their digits 100 000 times. When the main thread has done its output it waits for (joins to) the sub thread displays the message that all is done and exits.

Remark. The solution must be portable, in other words, it must be based on the POSIX standard. For that reason it is not allowed for a thread that has not locked the mutex to unlock the mutex.