Assignment #2

We have a 1,000,000 bank accounts (represented by a number) initialized with $1,000, implement a program to do 100,000 transfers between two randomly chosen accounts (in each iteration). The amount of money transferred is $50 and no negative accounts' balances are allowed.

You need to run the program using 1, 2, 4 threads. The number of transfers should be split on the number of threads. For example, 25,000 transfers per thread when we are running 4 threads.

The total amount of money before and after the transfers must be the same. You program should print the sum before and after executing all transfers.

Implement the program using coarse-grained (one lock) and fine-grained locking (multiple locks). You should come up with a technique to implement fine-grained locking.

Measure the total execution time for each case and prepare a plot with 6 bars (3 threads settings * 2 implementations [coarse & fine])

Notes:

You must use barriers to ensure all threads started at the same time. A skeleton file is provided which you can use to start (it has barriers and time calculations).

Deliverables:

1. Program implementation (C/C++ file(s))
2. Bars Graph + Tabular Results Data (any format, but I suggest Excel or Google Sheets for simplicity)

Dead line:

Jan 31st, 2018 at 12:00 PM