Goals

- Compare recursion versus iteration algorithms efficiency
- Code reuse & Running time measurement

In this lab, you will **measure and compare** the running time of **recursive and iterative** implementations of **Fibonacci Numbers Calculation**. You do not need to implement the algorithms. Feel free to find the source online, or in a book. Just ensure you **cite the source**! Here is one site that discusses both versions and you can take this as a starting point:

https://www.codeproject.com/tips/109443/fibonacci-recursive-and-non-recursive-c (Links to an external site.)Links to an external site.
You must modify the program to measure the running time of these two algorithms. You can use this reference to measure execution time: http://www.cplusplus.com/reference/ctime/clock/ (Links to an external site.)Links to an external site.. You can use any other timing mechanism you choose (other than wall clock time).

Before you start, try to understand how the algorithms work and write down your expectations. When you measure the running time, start with small values of N and increase N until you can see clear difference of running time (eg: 10, 50, 100…). Record your measurements results, then write your analysis. You can list the representative data in a table with three columns: N, recursive time, iterative time. Briefly explain the time difference in these two algorithms.

Tip：You might need a large number of N to see the actual time difference. If you do not want to do that, you can run the same N for multiple times (100 for example) in one measurement (you can use a for loop to do that) and record the start and end time of this loop. Present the results in the form of “N (run for 100 times)”.

What to submit

You will submit the following files in a zip file to Canvas:

- All the program files
- A makefile
- A short report (.pdf) to summarize the measurement results, comparison and analysis