EE 5370
Programming Assignment #3
Due October 8, 2020

Modify your multi-threaded program that computes the Catalan number sequence in the following ways:

1. Each thread should sweep through the Catalan numbers from low to high (e.g., with two threads: thread 1 computes Catalan number 1, 3, 5, 7, … and thread 2 computes Catalan number 2, 4, 6, 8, …
2. Any number of threads the user selects should be supported.
3. Each thread writes Catalan numbers to memory. Specifically, to an appropriately sized array on the **heap**. Be sure that race conditions and deadlocks are avoided. That array should not be any larger than what is necessary to store the Catalan numbers the user has requested.
4. When all threads have completed, write the contents of the array containing the computed Catalan numbers to a file called "*catalan.dat*". Each of those numbers should be in fixed point format.

Remember, your program takes two command line arguments: the first specifies the *number of Catalan numbers* to generate and the second specifies the *number of threads* to use to compute the Catalan number sequence. This time you cannot assume any particular maximum value for number of threads; you will have to dynamically allocate any data structure to coordinate thread activities.

You must use the "***long double***" type to get the largest range possible; the Catalan numbers grow very quickly. Output must be in ascending order (fixed point format) in the file "*catalan.dat*".

You can name the source file whatever you like. You will submit this assignment via Blackboard on October 29th. The instructor will contact you to schedule a time for you to demonstrate its operation.

Grading rubric:

Submission	30
Correct thread creation/coordination	20
Use of dynamically-allocated data structures	20
Avoid race conditions/deadlocks (including verbal explanation)	15
Output format	15