Laboratory Programming Exercise 07
Computer Engineering 160
Fall 2018
C. Paolini

A well-known sorting algorithm performs C_N comparisons for a random permutation of N elements, where C_N is given by the recurrence relation

$$C_N = N + 1 + \frac{1}{N} \sum_{1 \le k \le N} \left(C_{k-1} + C_{N-k} \right)$$

for $N \geq 2$ with $C_1 = C_0 = 0$.

Develop a code that computes C_N for values of N from 1 to 20 in an iterative structure that implements the recurrence relation. Observe C_N is a function of C_{k-1} and C_{N-k}. Thus, define a recursive function

```
double C(int N)
```

that invokes a call to itself via `C(k-1)` and `C(N-k)`. An approximation for C_N is

$$C_N \approx 2N \ln N$$

Show how close C_N is to $2N \ln N$ by computing an error. Print the values of N, the number of function calls made to `C(int N)` to compute C_N, C_N, $2N \ln N$, and the absolute value of the difference between C_N and $2N \ln N$. For example,

```
N:  1, calls:            1, C(N):  0, aprx:   0.00, err: 0.00
N:  2, calls:            5, C(N):  3, aprx:   2.77, err: 0.23
N:  3, calls:           15, C(N):  6, aprx:   6.59, err: 0.59
.
.
.
```

Use a global variable declared outside the scope of `main()` to count the number of function calls made to `C(int N)` to compute C_N.

Generate a screen capture of your Eclipse IDE workspace, showing your code and the output of your program in a terminal. Show all 20 rows of output to reveal the number of recursive function calls made to compute C_{20}. Make sure you reset the global variable used to count function calls in your iterative structure each time you compute a new C_N. Also note that $C_1 = C_0 = 0$.

Please name your Eclipse IDE project *`Lastname_REDID_Lab_07`*. Create a ZIP file of your project folder and submit the ZIP file through Blackboard.