Parallel Computing Second Lab Report

This code used the same method as the first lab, which was creating a method to multiply matrices, but the difference between this one and the first one was that this lab is using parallel methods by using the pymp file. I was supposed to run it using 1 thread, 2 threads, 4 threads, 6 threads, and 8 threads. When I ran it with 1 thread 795.961 seconds, the one with 2 threads was 475.014 seconds, the one with 4 threads took about 329.548 seconds, the one with 6 threads took 325.750 seconds, and the one with 8 threads took 319.309 seconds.

Instruction to Run File

For this code, you can change the number of threads by going into the code and changing the number in the “with pymp.Parallel(6) as p:” part and changing the numbers in there to any number of threads that you want. This lab ended up taking the matrix from the MatrixUtils file, and prints it using the printSubarray method from the MatrixUtils file as well.