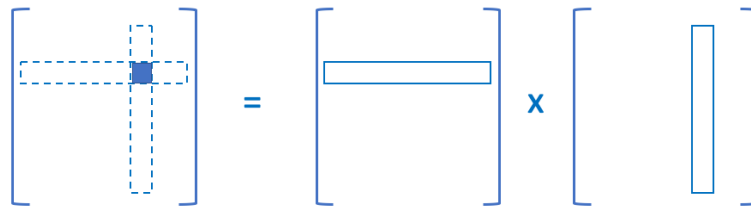For this assignment, use threads to multiply two square matrices. The number of threads to use is determined by the size of the matrices to multiply and a constant factor. For example, if the matrices have size 20 and the factor is 2, then the number of threads is $20/2 = 10$.

In case you have forgotten how to multiply two-dimensional matrices, here is a brief explanation. Let A and B be the operands and C be the result. Then:

$$C_{i,j} = \sum_{k=0}^{n-1} A_{i,k} * B_{k,j}$$

Because you are using threads to do the multiplication, each thread will be responsible for calculating some number of rows of the result. The number of rows is equal to the factor mentioned above. Continuing that example, if the matrices have size 20×20 and there are 10 threads, each thread is responsible for 2 rows of the result. Think carefully about which rows each thread should calculate.

You should do the following:

- Look at the code that is provided. Use that to help guide your problem solving. You might also want to revisit the `multi_thread_test.py` example from class on Monday.

- Complete the `main` method. This is where you will create and `start` the threads.

- Write the body of the `multiply` method.

- Test your program. Since the input is random, it will be difficult to determine if your result is correct. Change `m1` and `m2` so that all values in each are 1. Then `m3` should have all values equal to size. This is an easy way to determine if your program is working correctly.

- Bask in the warm glow of satisfaction from a job well-done and the undying admiration of your peers.