Assignment\_4 Written Portion

The system that I used in order to complete Assignment\_4 was macOS Mojave. The compiler that built/ran/executed my program was Xcode, which is an integrated development environment (IDE) for macOS containing various development tools. Within the algorithm I was able to implement multi-threading by being able to calculate the multiplication of a number in a certain index of a vector to every number within another vector in parallel. Each thread had a certain task to perform within the algorithm and the calculations of each thread was being performed in parallel. Threads are amazingly helpful for the calculations/computations of a program but become a hassle when trying to output. So, for this algorithm I had to use mutex and its lock/unlock functions to help the outputting portion. Using multiple threads in this program allowed the process of multiplying vectors together to be calculated in parallel which essentially sped up the execution time of the algorithm. Adding threads does not always cause an increase in program speed, the speed is dependent on the amount of threads the system’s hardware can handle. The speed of a program will increase if the amount of threads being spawned are within the systems hardware capacity for threads and will decrease if a program was to spawn more threads than the system’s hardware can handle. Within my code the part that was being parallelized by the threads were the multiplication of the two vectors. The portion of the code that was not parallelized was the output of the code. Since threads randomly finish tasks at any given time, then it is quite difficult to manipulate and format outputs in a certain way. One way to output threads in a more formattable way is to use mutex locking and unlocking functions. These functions allow the user to handle threads and their random completion times. Mutex lock and unlock functions only allow a single thread to go through a certain portion of code at one given time while the other threads are waiting their turn to go through.