I started off reading the text file line by line using the getline() c function into the InpArray integer array of 1200 integers. After the 1200 integer array was filled up, I closed the file using fstream(). I allocated memory for the three global arrays for the results of the computations for their 12 partitions in InpArray.

I made 3 computation functions that find the sums, sum of product squares and geometric averages.

The sums computation function just loops through 1000 index values in their designated part of the InpArray, and sums them up, saving their values to the global array float sums[12]. It was the simplest function to do.

The sqrt_sum_squares() function just squares each value, looping while adding them subsequently and then takes the value of the square root of the entire thing.

The geometric_avg() function was a bit trickier, at first I was confused why I was getting zero values at the end, but I realized if I had a zero being multiplied, it basically zeros out the entire geometric average despite the values that may follow subsequently. So I implemented a condition that if a zero was encountered while looping in the array, to break out of the entire process and zero out the value of that particular value of geometric average to equal 0. I also broke down the computation bit by finding the thousandth root for every 10 values and multiplying those values with each other.

For each of the 12 worker threads, they execute a worker computation thread passing one of those computations. So in my create_grandchilds() function I created 3 threads and attempted that.

In the main thread, I create the 12 worker threads from the struct child_thds which contain information about the threads and nested another struct for the worker computation threads named computations[3].

I was running into a couple of issues so I compiled with the –fsanitize=address flag and was getting a SEGV error. I believe something wasn't being implemented properly in the pthread_create() function at this point. I almost submitted a project that was incomplete but I decided to sleep on it. Luckily on Sunday morning on December 11 I woke up from a dream that hinted me to what was going wrong in my code. Silly enough I was basically passing in an incorrect index value in the computation threads in my create_grandchilds() function. So instead of passing a unique computation function manually 3 times, I decided to create a pointer to the function pointers and passed a function pointer that will be one of the functions for the computations. It not only made my code look more elegant but I could have avoided that manual indexing bug I was dealing with.

I also had a memory leak if I allocated memory for the value of line when reading through the textfile, but setting that value to NULL resolves it (at least on my end when compiling it).

The memory leaks were nerve wracking and I probably spent most of my time trying to find to find them all and freeing them. At this point I'm not sure what I could do more to improve my code but I think I did ok. I ran the code a couple of times and got the same results each time. I compiled using with an –lm at the end for the math library function pow().