The idea is simple -- turn your program from Assignment 1 into a multi-threaded parallel program! Here's the requirements:

1. Each block will be summed in its own thread, so that the block summations still proceed "simultaneously", but now in parallel rather than interleaved in one thread of execution.

2. Your program should create numBlock threads, and each block should be summed in its own thread.

3. Since pthread\_create() needs to be given a function to start the thread at, you will need to separate your block summation code into a separate function that is suitable to be passed to pthread\_create(). Note that this constrains it's interface, so you may need to get clever!

4. Since you will create numBlock threads, one for each block, the main program thread will NOT do any block summation, but will rather use pthread\_join() to wait for all the block threads to finish.

5. You must find a way to combine the individual block sums into a final total sum. If you use a global variable that all the threads can access, then be careful -- you MUST use a mutex to protect access to the variable! (google pthread mutex)

6. You should still be able to leave your timing calls in the main program -- just get the start time before creating the threads and get the end time after pthread\_join() returns. Since you are using real (wall-clock) time, this will take into account the time spent waiting for the child threads. If you use thread time, then the main thread will be really short since it doesn't do any of the actual summation anymore.

That should be enough to get started. I won't give much extra detail UNLESS you ask me questions, so the ball is in your court. Have fun!

### Submission

**Turned In!**

Feb 26 at 6:07pm

[Submission Details](https://nmsu.instructure.com/courses/1049933/assignments/5200980/submissions/3367007)

[Download sumprogramthread.c](https://nmsu.instructure.com/courses/1049933/assignments/5200980/submissions/3367007?download=104639701)

Grade: 10 (10 pts possible)

Graded Anonymously: no

#### **Comments:**

No Comments