Lab 2 Common Errors:

1) Lots of students haven't experimented with the threads. This resulted in so much overhead in their programs.

2) Several students didn't understand the meaning of parallelizing the original program, some just dispatched several threads, with each thread doing the "whole" serial program (for example: initialize A, B, C arrays) repeatedly.

3) Several students ignored racing problems in their pthread (using global min, max or sum without mutexes) or openMP programs (using shared min, max, sum without "critical section" annotation)

4) Many students wrote programs that can only run with fixed number of threads (for example, 100, each thread create/modify one array index), which cannot be used to compare performance between 2, 4, 8...128 threads.

5) Some students didn't consider the "uneven" work distribution of worker threads, for example, they let each worker thread to create/modify 30 array elements, then the last worker should only work on the last 10 array elements (on the contrary, use 4 threads with 30 elements work each, then ended up with out of array boundary error), or didn't finish the whole 100 elements work (exp: just dispatch 8 threads, each work for 12 elements)

6) Many student's didn't consider doing "functional parallel" for min/max/sum of array C (No points are deducted).