CSCI320: Homework 2

Due: Monday, February 5th

Most of these questions have many possible answers. Please provide some justification for your answer.
Each question is worth 4 points.

1. What is one difference between concurrency and parallelism? Give an example of a concurrent task and a parallel task.

2. If a program runs *more than* p times faster on p processors than on a single processor, then it is said to be a program that achieves **superlinear speedup**.

   In *most* circumstances this is impossible to achieve. Why do you think this the case?

3. Consider the first program we examined in the course which involved the summation of a lot of numbers. How might parallelizing this task be different on a SIMD, MIMD, or Shared-Memory computer?

4. Suppose you have written a long, coherent essay about English literature that spans many paragraphs. You are trying to parallelize the task of displaying this long essay to the screen. What issues could there be?

5. In real CPU hardware, caches do not store individual variables but *cache lines* or *cache blocks*, which consists of enough storage for multiple variables.

   Common cache line sizes are 32, 64, or 128 bytes. Cache lines can only be fetched/written a whole line at a time and CPUs will only keep track of entire cache lines at a time.

   The goal of such a design is to improve performance since it allows one operation to bring in a whole block of memory to the cache.

   However, this means:

   - When loading a variable into cache, the entire line must be loaded
   - When writing a variable from cache into main memory, the entire line will be considered updated.

   In a shared-memory system, how might this affect the problem of cache coherence? What impact does it have on performance?