Conceptual Questions

1. OpenMP is an API for writing multithreaded programs. Some of the benefits are that it is easy to use and can do parallelism in just a few statements, and it standardizes a variety of shared memory architectures.

2. The #pragma definitions are compiler directives. They specify how OpenMP should compile sections of the code.

3. #pragma omp parallel for

4. Reduction is used for combining values in a single accumulation variable. An example would be adding a value to a variable on each iteration of the loop.

5. The **critical** and **private** declarations are both used for synchronization. The **critical** declaration marks the critical section of code, which no two threads can run at the same time. The **private** declaration marks a variable that is shared by the threads and creates a private copy for each thread.