1) **What is the difference between the terms concurrency and parallelism?**
    a) Concurrency is concerned with managing access to a shared state from different threads. Concurrency does not imply parallelism as the order in which multiple, concurrent threads are running on a single processor can also determine a shared state. Parallelism is concerned with utilizing multiple processors and cores to improve the performance of a computation. Parallelism requires verification that the state of the system is valid for whatever application is running on it.
2) **What is the difference between shared memory and message-passing parallel computer models?**
    a) In the shared memory model, multiple tasks share a single memory area. These tasks can read and write to the shared memory region independently and asynchronously with the help of locks and semaphores that specify access control to shared variables. These shared variables make software design easier as the developer does not have to clarify communications between tasks; however, they make managing data locality more difficult. In the message-passing model, each processor has its own memory. Multiple tasks inhabit one processor or are distributed on an arbitrary number of interconnected processors. Developers must clarify the communications between processors as well as the amount of parallelism and number of processors.
3) **What is the difference between multi-threaded control parallelism and data parallelism?**
    a) Multi-threaded control parallelism is concerned with processes and threads. A process can have multiple flows of execution, which are known as threads. Threads generally interact with shared resources, so the multi-threaded control model is typically used with shared memory architectures. As such, multi-threaded programming is error-prone and requires the use of semaphores and locks. Data parallelism is concerned with tasks operating on the same data structure. Each task operates on a different portion of the data structure, typically an array, which greatly simplifies control needed to execute. As a result, the data-parallel model helps developers avoid race conditions.
4) **What is the difference between speedup and efficiency of parallel computation?**
    a) The speedup of a parallel computation is defined as the ratio of the sequential speed (T_S) to the parallel speed (T_P). Speedup measures the benefit of decreased execution time when solving a problem in parallel. The efficiency of a parallel computation is defined as the ratio of speedup (T_S/T_P) to the number of processors (p). Efficiency measures how much of the execution time is spent doing useful work. Efficiency is always measured to be less than or equal to one, and algorithms with linear speedup have an efficiency equal to one.
5) **You have to run a program that performs 4000 GigaFlop computation, and your core computes at a speed 30 TeraFlop/sec. How long will the program run for in seconds?**
    a) $\text{Execution Time} = \frac{\text{Work}}{\text{Speed}} = \frac{4000\times10^{9}\text{ Flop}}{30\times10^{12}\,\frac{\text{Flop}}{\text{sec}}} = \frac{4000}{30}\times10^{-3}\text{ sec}$

6) **A floating point program just completed in 0.32 seconds on a core with speed 2 TeraFlop/sec, how many GigaFlop does the program perform?**
   a) $\text{Work} = \text{Execution Time} \times \text{Speed} = (0.32 \text{ sec}) \times \left(2 \times 10^{12} \frac{\text{Flop}}{\text{sec}}\right) =$
      640 GigaFlop
7) **You have to run a program that performs 2000 TeraFlops, and your core computes at speed 450 GigaFlop/sec. How long will the program run?**
   a) $\text{Execution Time} = \frac{\text{Work}}{\text{Speed}} = \frac{2000 \times 10^{12} \text{ Flop}}{450 \times 10^{9} \frac{\text{Flop}}{\text{sec}}} = \frac{2000}{450} \times 10^{3} \text{ sec}$
8) **A program that performs 3000 GigaFlop just ran in 1.5 minutes on one single core. What is the core speed in TeraFlops per second?**
   a) $\text{Speed} = \frac{\text{Work}}{\text{Execution Time}} = \frac{3000 \times 10^{9} \text{ Flop}}{90 \text{ sec}} = \frac{3}{90} \text{ TeraFlops per sec}$