1. Suppose you want to write a kernel that operates on an image of size 400x900 pixels. You also like to assign one thread to each pixel. Your thread blocks are square and to use the maximum number of threads per block possible on the device. The maximum number of threads per block is 1024. How would you select the grid dimensions and block dimensions of your kernel? and why?

2. Consider a block with 8 threads executing a section of code before reaching a barrier. The threads require the following amount of time (in micro seconds) to execute the sections: 2.0, 2.3, 3.0, 2.8, 2.4, 1.9, 2.6, 2.9 and spend the rest of their time waiting for the barrier. What percentage of the threads' summed up execution times (i.e. cumulative time for all threads) is spent waiting for the barrier?

3. Can the scenario mentioned in the above problem 2 happen to threads of the same warp? why or why not?

4. What factors can make two threads corresponding to two different warps but of the same block take different amount of time to finish?

5. Can memory be coalesced for threads in a warp yet not-coalesced for threads in a different warp of the same block?

6. What is the difference between shared memory and L1 cache?