Week 8 Assignment 8

1. Which of the following is the scheduling of threads in a GPU streaming multiprocessor?
   a. Blocks
   b. Grids
   c. Maximum number of threads in a block in a direction
   d. Warp

2. The total number of threads active in a GPU at any instance is fixed by its hardware and known as warp size.
   a. True
   b. False

3. Why will we ever see typically small speed-up to a GPU code if all the threads are launched in a single block?
   a. Most of the time will be taken for execution
   b. All threads will be executed serially
   c. There will be a high overhead in accessing memory
   d. Cache coherence issues

4. What does a GPU provide control switching to get better parallelism?
   a. Using better control over the hardware
   b. Through GPU kernel calls using streams
   c. By using large number of registers
   d. None of the above

5. Consider the following code snippet. Why will this become a low-speed-up?
   
   ```c
   __global__ void kernel(int *data)
   {
     int i, tid = threadIdx.x;
     // or blockIdx.x * blockDim.x + threadIdx.x
     int x = (tid + blockIdx.x) % dataSize;
   }
   ```
   a. Reduce number of threads
   b. High compute to global memory access
   c. Use of a small number of registers
   d. Thread divergence

6. Shared memory is small on-chip memory shared by all threads in a
   a. Warp
   b. Block
   c. Grid
   d. Thread

7. In a matrix-vector product using CUDA kernel, how a thread can be assigned to work on a particular row of the matrix?
   a. Program specifies a separate offset for the thread and map it to the matrix row
   b. Threading function can find local thread id and map it to the matrix row
   c. Global thread id using threads block and threadIdx is divided by the matrix row
   d. Looping code

8. Matrix multiplication using tiling can be helpful to CUDA optimization because
   a. Fill part of the matrix with fill line ordered memory
   b. Better communication to compute nodes
   c. More number of threads can be launched
   d. None of the above

9. OpenCL cannot be used for parallelization of a code for the following platforms
   a. Cell GPU
   b. Coppermen
   c. Cell's Seymour's multiprocessor
   d. Continuum cluster

10. In a hybrid multi-GPU program, MPI can be used along with CUDA for communication between the GPUs.
    a. True
    b. False