LSU EE 4702-1 Homework 4 Due: 20 November 2012

For the problems below refer to Chapter 2, Programming Model, and Chapter 4, Hardware Implementation, in the NVIDIA CUDA Compute Unified Device Architecture Programming Guide available locally via
`http://www.ece.lsu.edu/gp/refs/CUDA_C_Programming_Guide.pdf`.

Problem 1: The kernel below is launched in a configuration of grid size of $(1024, 1, 1)$ and block size of $(256, 1, 1)$. (The components are (x, y, z).)

```
__global__ void dots_iterate1() {
  int thread_count = /* Omitted so things aren't too obvious. */;
  int idx_start = threadIdx.x + blockIdx.x * blockDim.x;
  for ( int idx = idx_start; idx < array_size; idx += thread_count )
    b[idx] = v0 + v1 * a[idx]; }
```

(*a*) How many threads are there? How many blocks? How many warps?

(*b*) Suppose the array size is $5 \times 10^5 = 500\,000_{10} = 7\,a120_{16}$ elements. How many threads will perform two iterations? One iteration? Zero iterations?

(*c*) A GPU has four multiprocessors. Explain why launch configurations of four and eight blocks are each better than a launch configuration of six blocks. Use the code above as an example.

Problem 2: Consider the kernel code below. It is launched with a block size of 512 threads and a grid size of 64 blocks. The array has $2^{20} = 1\,048\,576$ elements.

```
__global__ void dots_iterate15()  {
  int thread_count = /* Omitted so things aren't too obvious. */;
  int tid = threadIdx.x + blockIdx.x * blockDim.x;  // Not used in code.
  int idx_start = blockIdx.x + threadIdx.x * gridDim.x;

  for ( int idx = idx_start; idx < array_size; idx += thread_count )
    b[idx] = v0 + v1 * a[idx];
}
```

(*a*) The table below shows various information about selected threads in the launch described above. The first three columns should be self-explanatory. The three columns headed `idx` show the element number (value of `idx` in the code above) accessed by the respective thread in the first, second, and third iteration of the `for` loop. Each row has at least one column filled. Fill the remaining columns.

			-- idx ---	-- idx ---	-- idx ---
tid	blockIdx.x	threadIdx.x	First Iter	Second Iter	Third Iter
0					
1					
2					
			0		
			1		
			2		
	1	1			