LSU EE 7722 **Homework 1** **Due: 24 February 2017**

Problem 0: Read the following information about the assignment package, and follow instructions on course procedures page, `http://www.ece.lsu.edu/gp//proc.html`, for account setup and Programming Homework Workflow. Try compiling and running the code and familiarize yourself with the command line arguments described below.

Problem 1: The code in `hw01.cc` is similar to the code in `.../cuda/intro-simple/thds.cc` except that threads with an even value of `tid` initialize array elements and threads with an odd value of `tid` perform the computation. However, this was achieved only by guarding the initialization and computation loops with an `if` statement, no other changes were made to the work assigned to each thread and so the wrong answer will be computed.

The `main` routine spawns `thread_main` threads in two rounds, a *parent-syncs round* and a *child-threads-sync round*. In parent-syncs round, even-numbered threads are spawned first. When all the even-numbered threads finish odd-numbered threads are spawned. When all the odd-numbered threads finish results are checked, and the child-threads-sync round starts.

In the child-threads-sync round all threads are spawned (odd and even) in one step. After all of these threads finish results are checked.

There is a structure `App` which has information for the threads to use. Member `App::thds_sync` is set to `false` during the parent-syncs round and is set to `true` during the child-threads-sync round.

(*a*) Modify the code so that correct answers are obtained during the parent-syncs round. This can be accomplished by changes to `thread_main`.

(*b*) Modify the code so that correct answers are obtained during the child-threads-sync round. This will require changes to `thread_main`, the `main` routine, and the `App` structure, and will require the use of synchronization mechanisms not covered in class. See the C++11 thread support library. A solution must be reasonably efficient, no spin waiting.