CS/EE 217 Lab 1
Simple Matrix Multiplication
Due Wed. Oct 14 at 8pm

1) Unzip lab1-starter into your sdk projects directory

2) Edit the basicsgemm() and the mysgemm(...) function in kernel.cu as well as main in main.cu to complete the functionality of the matrix multiplication on the device. Do not change the source code elsewhere. The size of the matrix is defined such that one thread block will be sufficient to compute the entire solution matrix.

3) There are several modes of operation for the application. Check main() for a description of the modes (repeated below).

a) No arguments: The application will create two randomly initialized matrices to multiply size (1000x1000). After the device multiplication is invoked, it will compute the correct solution matrix using the CPU, and compare that solution with the device-computed solution. If it matches (within a certain tolerance), if will print out "Test PASSED" to the screen before exiting.

b) One argument: The application will use the random initialization to create the input matrices (size mxm, where m is the argument. Start your testing with small matrices.

c) Three arguments m, k, and n: The application will initialize the two input matrices with random values. A matrix will be of size m x k while the B matrix will be of size k x n, producing a C matrix of size m x n

Note that if you wish, you may add a mode to accept input matrices from files, or to dump input and output matrices to files to facilitate testing.

4) Answer the following questions:
1. How many times is each element of the input matrices loaded during the execution of the kernel?

2. What is the memory-access to floating-point computation ratio in each thread? Consider a multiply and addition as separate operations, and ignore the storing of the result. Only global memory loads should be counted towards your off-chip bandwidth

3. Describe two approaches to speedup the computation. You may assume large matrices.

Submission:

Upload a zip file with your updated code and report to iLearn. Please remove all executable files or other unnecessary files from the directory before creating the zip. Please name your report report.txt , report.pdf or report.doc

Grading:

Your submission will be graded on the following parameters.
Correctness: 35%

- Test passes in all three modes.

Functionality: 35%
- Correct usage of CUDA library calls and C extensions.
- Correct usage of thread id's in matrix computation.

Report: 30%
- Answer to question 1: 10%, answer to question 2: 12%, answer to quester 3: 8%