COSC 480 – Miniproject #2

1.) For this miniproject, you will implement matrix addition utilizing shared memory and CUDA atomic operations. Given two matrices A and B of the same dimensions, A + B =

$$\begin{bmatrix}1 & 2 & 3 & 4\\5 & 6 & 7 & 8\\4 & 2 & 1 & 1\end{bmatrix}+\begin{bmatrix}5 & 1 & 2 & 4\\4 & 5 & 2 & 1\\7 & 3 & 5 & 2\end{bmatrix}=\begin{bmatrix}6 & 3 & 5 & 8\\9 & 11 & 9 & 9\\11 & 5 & 6 & 3\end{bmatrix}$$

TASK 1: Create a CUDA kernel utilizing shared memory that will compute the addition of 2 matrices *A* and *B* of size *n x m*. These are to be integer matrices for reasons that will be clear later.

TASK 2: Create a second CUDA kernel utilizing shared memory and atomic operations that will do the same thing as task 1. There's a very nice substitution that can be made.

TASK 3: Create a main method that will run your two CUDA kernels sequentially on two matrices of sufficient size (> 300000 elements) and randomly initialized. I do not expect you to output to the screen the original and result matrices, though a nicely formatted file output would be nice (but not required). You will also use some timing code to time each kernel individually, and you will report the timing results before exiting the program. You should use the timing code available in section 6.3 of your book. Include as a comment at the bottom of your code your reasoning for why you see the performance difference between the two kernels.

2.) Submit your sufficiently commented source file via Blackboard by 11:59pm on 10/26. This project may be worked on in pairs and you may not work with your partner from the first project. If you choose to work in a pair, you must include that information as part of your comment block at the start of your source. I will be executing your code on AWS.