CSCD 439/539 GPU Computing Lab4
Matrix Multiplication

No Late Submissions are accepted. **Rules:** Your code must use C and CUDA C Language. If your program shows a compilation error, you get a zero for this lab assignment.

Submission: Wrap up all your **source files and other data files** into a single zip file. Name your zip file as *FirstInitialYourLastName*Lab4.zip. For example, if your legal name is Will Smith, you should name your zip file as wsmithlab4.zip. A simple makefile has been provided in the zip file.

Before you leave the laboratory, please show the TA or the instructor how your program works, they will give you a score for this Lab assignment.

For archive purpose, please also submit your single zip file on EWU Canvas by following CSCD439-01 Course → Assignments → Lab4 → Submit Assignment to upload your single zip file.

Problem Description:

Based on the lecture about simple matrix multiplication on CUDA device, you are required to implement the following features and answer the questions.

In the provided lab package, you have three subfolders, **data**, **src** and **sampleCode**. The **data** folder contains all 2D matrices you will play with. The **src** folder has most of the source code you need to perform experiments. The sampleCode provides a single demo code to generate random integer numbers in C.

1, Read the provided code in **src** folder, specifically read the main function in the source file matrix_multiplication.cu. Answer the questions below,

a) What tasks the program performs?

b) What are the dependency C files and header files? How to call functions that is defined in another source file?

2, Based upon the lecture notes, please write the simple kernel function to perform matrix multiplication on GPU, on top of the source file **matrix_multiplication.cu**.

3, Explore the Makefile, how shall we jointly compile .cu and .c files in a single project?

4, Check the APIs Docs and find out what cudaEventRecord() and cudaEventSynchronize() do?

5, In the main function, what is the equation that the program uses to compute the throughput (in unit of GLOPS)? Please interpret the equation.

6, Run your program after you finish your kernel on the dataset 1024.mat and 2048.mat, how much speedups do you obtain compared with CPU time cost? What are the GPU throughput and CPU throughput you observed in these cases?

7, Can you find some specification data about the peak performance (GFLOPS) for the GPU that we are using in the Lab (NVIDIA GTX 660 ti) ? Is the GPU device throughput that you observed in step 6 close to their Peak performance? Guess the reason why they are close or why they are far away?

8, Refer to the sampleCode provided, you have to add another function in arrayUtils.c and arrayUtils.h, float * fillArrayRandom(int row, int col), which populates a 2D matrix with random **float** number ranging 0 to 1. Redirect your **input** from data files to these randomly generated matrices, record only the GPU throughput and time cost for matrix multiplication with size 10000 * 10000, 20000 * 20000. (FYI: please comment out the CPU sequential code test because they takes forever to finish. Also please make sure the thread block size is within limit.) If you can use **long long integers** for the size of the array, you can test arrays with much more bigger size.