CSCD 439/539 GPU Computing Lab2
Diff operation on GPU

No Late Submissions are accepted.

Rules: Your code must use C and CUDA Language. If your program shows a compilation error, you get a zero for this lab assignment.

Submission: Wrap up all your **source files** into a single zip file. Name your zip file as *FirstInitialYourLastName*Lab2.zip. For example, if your legal name is Will Smith, you should name your zip file as wsmithlab2.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→Lab2→ Submit Assignment to upload your single zip file.

Problem Description:

In science and engineering, **diff** is used to calculate Differences and Approximate Derivatives. The definition of **diff** is as follows.

Y = diff(**X**) calculates differences between adjacent elements of X, where **X** and **Y** are 1D array in this lab. If **X** is an array of length **m**, then Y = diff(X) returns a vector of length **m-1**. The elements of **Y** are the differences between adjacent elements of X.

Y = [X(1)-X(0), X(2)-X(1), ..., X(m-1)-X(m-2)]

I have provided most of the code in the zip file. You are required to implement the following features and answer the questions.

1, Read the provided code and understand the input, the output, and how data is transferred between host and device.

2, According to the definition of diff above, please write the kernel to perform diff on GPU device.

3, The input array contains 18 float numbers. I set the block size to 4, defined on the top of source code as a Macro. Please set up the gridSize, blockSize (**formally called execution configuration**) and launch the kernel you implemented. You are supposed to get the results like the following after your run your program.

```
The original array is:   4,   5,   6,   7,  19,  10,   0,   4,   2,   3,   1,   7,    9,  11,  45,   23,  99,  29,

The diff      array is:   1,   1,   1,  12,  -9, -10,   4,  -2,   1,  -2,   6,   2,   2,  34,  -22,  76, -70,
```

4, Did you notice any duplicated memory access in the kernel?

5, Please explain how each thread is mapped to an unique spot(index) of the **output array**? Can you draw some conclusions when mapping thread to data with regards to whether you should think the input array first or the result array first?

6, Is there any data read/write conflict in the kernel for each thread? That is, in you kernel, are there any two threads reading a same element at the same, or is it possible to happen?

7, In you kernel, are there any two threads writing a same spot at the same, or is that case possible to happen?

8, how many thread blocks are generated by the kernel?