University of Arkansas
Department of Computer Science & Computer Engineering

Lab 1: Vector Addition

Due Date: February 6 (Thursday), 2020 at 11:59pm

1. Objective

The purpose of this lab is to get you familiar with using the CUDA API by implementing a simple vector addition kernel and its associated setup code.

2. Procedure

Step 1: Download the Lab 1 materials from blackboard to your home directory on the Karpinski GPU cluster. Unzip the file.

```
unzip  lab1-vector-add.zip
```

Step 2: Edit the file `lab1-vector-add/main.cu` to implement the following where indicated:

a) Allocate device memory

b) Copy host memory to device

c) Initialize thread block and kernel grid dimensions and invoke CUDA kernel

d) Copy results from device to host

e) Free device memory

Step 3: Edit the file `lab1-vector-add/kernel.cu` to implement the vector addition kernel code.

Step 4: Compile and test your code.

```
cd lab1-vector-add

make

./vecadd          # Uses the default vector size

./vecadd m        # Uses vectors of size m
```

Step 5: Submit your assignment. You only need to submit the following files:

- `main.cu`
- `kernel.cu`

Compress the files and name them after your last name as follows:

```
tar -cvf lab1_<lastname>.tar main.cu kernel.cu
```

Submit the compressed file to blackboard before the deadline.

3. Grading:

The full grade of this lab is 100 points, which are broken down as follows.

- Correctness (60 points): both default vector size and user-input vector size will be tested.
    - If code is not functioning correctly, grade will be zero.
- Completeness (30 points): all the missing parts are completed.
- Test on the return values of APIs (10 points): check if the calling of APIs is successful.
    - See the last slide of Lecture 2 for example.

Note: This is a simple but essential exercise. Please write out the code and do not copy it from other examples or lecture slides. That process is most important.