

COMP 193
Intro to GPU programming using CUDA
Exercise 1

In this exercise you will use code templates to generate your first program. The program will create two vectors of data and add those vectors together, either using the CPU or the GPU.

- 1) To get started, download and extract Exercise 1 package: exercise1.zip
included in the .zip are five files:

- interface.cpp
- interface.h
- gpu_main.cu
- gpu_main.h
- Makefile

Put them in a directory and type 'make' at the command line. Assuming everything goes well, this should generate an executable called ex1. Run the executable by typing:

```
> ./ex1 -r2 -v1
```

If you cannot get this to work, please notify me right away.

- 2) read through the code to see what is going on, and play with using different values for the flags when running the program.

3) A first priority is to be sure everyone knows how to work with passing args to the command line. So the first part of the exercise is to add a flag that specifies how big of a vector to use. Please modify the interface.cpp file to accept a new flag: -s where you can specify the size of the vector to be used from the command line. All the information you need to do this should be contained in the code. Just use the other flags as a template. The result is that you should be able to e.g. enter the command:

```
> ./ex1 -r1 -v0 -s100
```

to specify the vecSize to be of length 100 when the program executes.

Also please update the usage() function to include this new flag.

4) Your next task is to use code from add_loop_gpu of Chapter 4 to update the code of gpu_main.cu to run the program. Mainly you will be just copying and pasting code. But please try to understand what is going on, and read Chapter 4 of the book.

- 5) using a vector length of 100,000, was the CPU or GPU faster?

Please send me your modified code and your answer to question 5 over e-mail. Thanks.

(you may have to tinker with flags in the the makefile, depending on environment set-up)

(kudos given to anyone who catches any glitches in my template code!)