1. General Remarks

This assignment is an introduction to CUDA programming. You will be using the ece cluster. Nodes equipped with NVIDIA CUDA GPUs are

```
en-openmpiXX.ece.cornell.edu
```

where `XX = 04, 05, 06, 07`. You should be able to login to any of the nodes using your Cornell netid.

For more info about CUDA consult

```
http://docs.nvidia.com/cuda/cuda-c-programming-guide/
```

or any other convenient source.

2. Compile

Your codes should be compiled with

```
/usr/local/cuda-8.0/bin/nvcc -arch=compute_XX -o foo.out foo.cu
```

(and any other flags that are required) where

- `XX = 52` for nodes 05 and 07 (GeForce GTX 980), and
- `XX = 35` for nodes 04 and 06 (Tesla K40C).

3. Assignment

You are asked to implement (square) matrix-matrix multiplication on CUDA. There are four versions of multiplications (see lecture notes for Lecture 18 and 19),

1. the "naive" version which computes $c_{i,j} = \sum_{0=1}^{N-1} a_{i,k} b_{k,j}$ by traversing the entire row i and column j,
2. the "tiled" version where tiles of appropriate dimensions $K \times K$ are first loaded to shared memory and then used to accumulate partial sums $c_{i,j}^{(m)} = \sum_{k=mT}^{(m+1)T-1} a_{i,k} b_{k,j} + c_{i,j}^{(m-1)}$ where m is the index of the current tile,
3. same as (2) but where tiles of matrix B are transposed when loading the shared memory so bank conflicts are avoided,

4. a call to the CUBLAS library.

All versions should be executed for different sizes of matrices, different sizes of (Block, Grid) thread arrangements, and execution times should be measured. (The execution times should include host-to-device and device-to-host transfers).

4. Format

- Your codes must be written in standard C language with CUDA extensions, and compiled with `nvcc`.
- Your code must be well documented.
- The first line in the code must show how the code should be compiled.
- For a fixed matrix size $N \times N$ you are asked to run your code for a number of (Grid,Block) combinations.
- For a fixed (Grid,Block) combination you are asked to run your code for progressively larger matrices.

For each case report speed-up (or slow down) over a sequential algorithm. If there is a slow down, identify the parts of the code that contribute to the slow down.

1. Your code(s) must be saved in separate file(s) named `your_net_id_hw4_p_to_r.c` where `p` and `r` refere to a particular code or combined codes. For example if you write a single code for codes 1 to 3 then set `p = 1` and `r = 3`. For code 4 set `p = r = 4`.
2. Your codes must be described in a single file `your_net_id_hw4.pdf`. Please include your NAME and net ID on all pages of your write up. Please DO NOT submit `*.docx` files as I have difficulties with printing them.
3. All files need to be packed with the `tar` or `gzip` facilities. The packed file must have the name `your_net_id_hw_4.suffix` where `suffix` is either `tar` or `zip`.
4. If you relay on resources outside lecture notes but publically available, you need to cite sources in your write-up(s).