EECE 528: Parallel and Reconfigurable Computing
ASSIGNMENT 0

DUE DATE: SEPT 22

SUBMISSION (DO BOTH!):

1. **DROPBOX (make folder 'yourid', save yourid-report.pdf, yourid-src.zip)**
2. **HYDRA2 (eg: /data/home/eece528/2017/submit/yourid/assign0/)**

1. Write a "vanilla" matrix multiply program in C
   - Compute C[] = A[] * B[]
   - Parameterize the data type to support one of {int, float, double}
   - Try different ways of allocating arrays in C
     i. int a[m*n];
     ii. int a[m][n];
     iii. int *a
     iv. int **a;
   - Parameterize the size of the matrix, support up to 4096 x 4096; this will require about 64MB of memory per matrix.
   - Initialize with random data
   - Measure and print the run-time of matrix multiply only (not init)
2. Compile each of the 3 data types (int/float/double) against 5 possible compiler optimization settings (none, **-O**, **-O1**, **-O2**, **-O3**, and one more flag setting, see below) to produce 15 different executable programs. The extra flag setting should be something like the following (but feel free to adapt these flags)
   - For hydra2, use "**-O2 –ftree-vectorize –m64 –march=broadwell –mtune=intel**")
   - Record version of gcc (gcc --version). Also check /proc/cpuinfo (Intel(R) Xeon(R) CPU E5-2699 v4 @ 2.20GHz) and the gcc manual starting here:

   https://gcc.gnu.org/onlinedocs/gcc-4.9.3/gcc/i386-and-x86-64-Options.html#i386-and-x86-64-Options
   https://gcc.gnu.org/onlinedocs/gcc-4.9.3/gcc/Optimize-Options.html#Optimize-Options
3. Run your programs on hydra2.ece, using 'htop' to ensure you are not competing for CPU or memory resources. Plot your runtime results for a variety of matrix sizes: 64, 256, 1024, 2048, 4096.
4. Rewrite your "vanilla" code by applying as many typical compiler transformations as you can; document the ones that you applied in a list.
5. Repeat execution in step 3.
6. Write a 1-page report, including a table and a graph showing results of Steps 3 and 5. Discuss the differences in performance for different data types, compiler settings, and matrix sizes. Do not present raw data for all possible combinations; show interesting trends only. However, you must present a full set of results for "float" using "-O3" with all matrix sizes
7. Write a 1-page report and hand it in.

PREPARATION FOR ASSIGNMENT 1 (not a part of this assignment, do not hand in)

1. Repeat, but using Gaussian Elimination instead of matrix multiply.
2. Implement and optimize **only** the forward-elimination. Do not perform back-substitution. Focus only on single-precision "float" performance.