Problem Set 2

Problem 1.
The goal of this problem is to realize a CUDA implementation of the PRAM algorithm seen in class for univariate polynomial multiplication. See http://www.csd.uwo.ca/~moreno/cs3101_Winter_2015/PRAMs.pdf.

To simplify the implementation, we will make the same assumptions as in our PRAM algorithm:
1. The two input polynomials have the same degree, namely $n - 1$
2. The integer $n$ is a power of 2.

Your CUDA program should use multiple thread-blocks and allow $n$ to reach (at least) $2^{10}$. Note that the PRAM model does not have a concept of thread-block. So the $n^2$ processors of the PRAM algorithm can translate to $n$ thread-blocks with $n$ threads, or, $n^2/t$ thread-blocks with $t$ threads.

The input polynomials should be randomly generated on the CPU, then transferred to the GPU. The product computed on the GPU should be transferred back to the CPU. Similarly to the simple examples seen in class, your CUDA program should include a C function which verifies that the result of the computations performed on the GPU are correct.

The input polynomials should have random coefficients in the range $0 \cdots p - 1$ where $p$ is a small prime number, say $p = 103$. Then, all products and sums of coefficients should be computed modulo $p$. This will prevent from overflow and will make the verification feasible.

Question 1. [60 points] Realize a first version of your CUDA program using $n$ thread-blocks with $n$ threads.

Question 2. [40 points] Modify your CUDA program such that it can work with $n^2/t$ thread-blocks with $t$ threads, where $t \in \{64, 128, 256, 512\}$. 


Submission instructions.

**Format:** *Input tests* and a **Makefile** (for compiling and running) are required. Please provide a **README** describing how to compile and test your code. Please submit source code only! All the files should be archived using the UNIX **tar** command.

**Submission:** The assignment should be submitted via OWL.

**Collaboration.** You are expected to do this assignment *on your own* without assistance from anyone else in the class. However, you can use literature and if you do so, briefly list your references in the assignment. Be careful! You might find on the web solutions to our problems which are not appropriate. For instance, because the parallelism model is different. So please, avoid those traps and work out the solutions by yourself. You should not hesitate to contact me if you have any questions regarding this assignment. I will be more than happy to help.

**Marking.** This assignment will be marked out of 100. A 10 % bonus will be given if your submission is clearly organized. Messy assignments (no Makefile, no README, no tests) will receive a 10 % malus in addition to possible deductions (e.g. if the code cannot be compiled).