CS 315
Homework 4, Fall 2017
Due: Thursday, October 12, 11:59 pm, on **Canvas**

In this assignment you must **not** use your programming language support (nor any libraries) for arbitrary length arithmetic.

Some procedures that you created while solving Homework 2 may be useful in this assignment too.

Problem 1. Design and implement an algorithm that takes two non-negative binary integers x and y represented as in Homework 2 and returns the quotient and the remainder of the integer division of x by y. Assume that $y > 0$. Represent both the quotient and the remainder as vectors of binary integers. (You can implement the division algorithm from the textbook).
Test your program for the following binary integers x and y:

1. $x = [0, 1, 1, 0, 0, 0, 1, 1, 1]$ and $y = [1, 1, 1, 1, 0, 1, 1, 0, 1]$
2. $x = [1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1]$ and $y = [0, 1, 1, 0, 1]$
3. $x = [1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1]$ and
   $y = [1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1]$.

Problem 2. Design and implement an algorithm that takes a positive integer N and two non-negative integers x and y, and returns $x^y \bmod N$. Assume that $x, y \leq N - 1$, and that, N, x, and y are represented as vectors of binary digits (like in Homework 2 and Problem 1).
Test your program for

1. $N = [0, 0, 1, 0, 1]$
   $x = [0, 1]$, $y = [0, 1, 0, 1]$
2. $N = [1, 0, 0, 0, 1, 1, 0, 0, 0, 1]$
   $x = [1, 0, 1, 1]$, $y = [0, 0, 0, 0, 1, 1, 0, 0, 0, 1]$
3. $N = [1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1]$,
   $x = [1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1]$,
   $y = [1, 1, 0, 0, 0, 1, 1]$

Problem 3. Test the performance (time in seconds) of your modular exponentiation program on integers of different sizes. Comment on how the performance of the program

depends on the number of digits in the input vectors representing binary integers. Use the following test inputs:

$$N = [1, 0, 1, n\ 0's, n\ 1's, 0, 1], x = [1, 1, n\ 0's, n\ 1's], y = [1, 0, n\ 0's, n\ 1's],$$

where $n = 1, 2, 4, 8, 16, 32, 64, 128$ etc. Continue doubling n but stop when the time of a test exceeds 600 seconds. You may want to write a procedure that generates these vectors rather than have the user to type them in.

In your program you must not convert the input binary vectors representing numbers to decimals and use the built-in arithmetic operations on decimals. The operations must be performed on the binary vectors.

To write your programs you have to use one of the following programming languages: C/C++, Java, or Python 3.0 or higher.

Submit two files. One of them must contain the report in pdf format. The other one, submitted as a single .zip file, should contain the source codes (with adequate and clear comments) and, possibly, data sets.

In your report include a brief description of the algorithms you implemented, a list of test cases you tried and the results. The report should also contain compilation and execution instructions for your programs.