CS315, Fall 2016
Homework 5 - Bonus
due: October 22

There are four pages and seven problems. Read carefully all specifications

You must **not** rely on your programming language support for arbitrary length arithmetic nor on any libraries that might provide it.

In this assignment you will design and implement procedures for arithmetic operations (adding numbers, converting numbers from one base to another). These procedures must assume that numbers are represented as sequences (lists, vectors) of digits (in a given base). For instance, the binary number 1101 (the rightmost 1 being the least significant digit) should be represented as the list $[1, 0, 1, 1]$ (position 0 in the list containing the least significant digit). Similarly, the decimal number 806732006 should be represented as the list $[6, 0, 0, 2, 3, 7, 6, 0, 8]$.

When testing your procedures, you will assume that the numbers are given as files. Each line in the file should contain a single digit, the first line should contain the least significant digit and the last line — the most significant one. Thus, the file with these lines:

1
0
1
1

represents the integer 1101. To represent 0 you will use a file with no content, that is, no lines. Your main program will access the files with the digits of the input numbers, read them in and store them as a list. You will need to implement a procedure designed for that task. In particular, after reading the file above, this procedure should return the list $[1, 0, 1, 1]$.

The files can be created manually or by a designated program (the latter recommended for those test numbers that have many digits). The files (and programs you used to generate them) should be included with your submission.

You will test the procedures you will decing for this assignment by calling them from your main program.

Problem 1: Design an algorithm that takes two numbers in base $b \geq 2$ (b is an input parameter) and adds them up generating the result also in base b. For instance, if $b = 3$, and the two inputs numbers (already in their list representations) are $[1, 0, 2, 2]$ and $[2, 2, 1, 2, 1]$, the result should be $[0, 0, 1, 2, 2]$.

Assume that b (and so also each digit in base b) can be stored in a single machine word.

Test your program by adding these numbers the numbers here are given in the standard notation, they will

be "reversed" in the list notation that your programs will operate on):

1. 0 and 430020104 ($b = 5$)
2. 430020104 and 430020104 ($b = 5$)
3. 202210101212 and 2201220 ($b = 3$)
4. 1010111001011110 and 10011110101 ($b = 2$)
5. 1010111001011 and 1111111010110 ($b = 2$)

Problem 2: Design and implement an algorithm to convert numbers from binary to decimal (binary numbers given as lists of binary digits and decimal numbers given as lists of decimal digits). For instance, the input might be $[1, 0, 0, 0, 1, 1, 1, 0, 1]$ (representing the binary number 101110011). The corresponding output should be $[1, 7, 3]$ representing the integer 371 (in the standard decimal notation).

Test your program by converting into decimal:

1. the number consisting of 16 1's (in the most significant places) followed by 6 0's followed by 3 1's (that is, 11111111111111111000000111)
2. the binary number starting with three 1's (in the two most significant places) followed by 150 0's, then by one 1 and then 149 0's.

Problem 3: Design and implement an algorithm to convert numbers from decimal to binary (as before, binary numbers given as lists of binary digits and decimal numbers given as lists of decimal digits). For instance, the input might be $[9, 8, 3]$ (representing the integer 389). The procedure should return $[1, 0, 1, 0, 0, 0, 0, 1, 1]$ (which, in the standard notation is 110000101, a binary representation of 389.

Test your program by converting into binary:

1. the decimal integer consisting of four 7's (in the most significant places) followed by four 3's followed by two 1's (that is, 7777333311)
2. the decimal intger starting with two 5's (in the two most significant places) followed by twenty 2's, then by one 9 and then twenty 6's.

Problem 4 (comparison): Design and implement an algorithm that takes two non-negative integers x and y (in base 2, represented as lists of their binary digits), returns **true** when $x \geq y$ and **false**, otherwise.

Test your program on these pairs of numbers (given already as lists):

(a) $x = []$ and $y = []$

(b) $x = []$ and $y = [1, 1, 0]$

(c) $x = [1, 1]$ and $y = []$

(d) $x = [1, 1, 0, 1, 0, 1, 1, 0, 1, 1]$ and $y = [1, 1, 0, 1, 1, 1, 1, 0, 1, 1]$

(e) $x = [1, 0, 0, 1, 1, 1, 0, 1, 1]$ and $y = [1, 0, 0, 1, 1, 1, 0, 1, 1]$

(f) $x = [0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1]$ and $y = [1, 1, 0, 0, 1, 1, 1, 1, 0, 1]$

(g) $x = [1, 1, 0, 1]$ and $y = [1, 1, 1, 1, 0, 1]$

Problem 5 (subtraction): Design and implement an algorithm that takes two non-negative integers x and y (in base 2, represented as vectors of binary digits, assume that $x \geq y$), and returns $x - y$ (in base 2; represented as a vector of binary digits).

Test your program by subtracting

(a) $y = [1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1]$ from $x = [0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1]$

(b) $y = [1, 1, 0, , 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1]$ from
$x = [1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1]$

(c) $y = []$ from $x = [1, 0, 1, 0, 1, 1, 1, 1, 0, 1]$

(all numbers in vector representation, with the least significant digit as the first element in the sequence).

Problem 6 (multiplication and division): Design and implement algorithms that take two non-negative integers x and y (in base 2, represented as vectors of binary digits) and return

(a) the product xy (in base 2, represented as a vector of binary digits)

(b) the quotient and the remainder of the integer division of x by y (you may assume that $y > 0$, for this procedure; both the quotient and the remainder represented as vectors of binary digits)).

Test your program for x and y as in Problem 5 (a) and (b).

Problem 7: Design an algorithm that takes a positive integer N (modulus) and two non-negative integers x and y, and returns $x^y (mod\ N)$. (Assume that $x, y \leq N - 1$, and that N, x and y are in base 2 and are represented as vectors of binary digits.)

Test your program for

(a) $N = [1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1]$,
$x = [1, 1, 0, 1, 1, 1, 1, 1, 1]$, $y = [0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1]$

(b) $N = [1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1]$,
$x = [1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1]$,
$y = [1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1]$

Submit the source codes (with adequate and clear comments) and the files with the test cases, instructions on how to run the programs, and a report with a brief description of the algorithms you implemented, a list of test cases you tried and the results.

Reports must be pdf documents. All files (reports, data files, source code, compilation and execution instructions, etc. must be submitted as a single .zip file)