Large Number Arithmetic
Chris Whiting
3 May 2011

I.   Problem Statement

    A. General description

        The assignment is to develop a class, SuperInteger, for doing arithmetic with
        arbitrarily large integers, including addition, subtraction, multiplication, division,
        and remainder.

    B. Problem details

        1. Each number can be arbitrarily long.
        2. Each Super Integer is taken in as a String and outputted as a String prior to
            an operation.
        3. The program will be invoked with the command

            Java SuperSolver args[0] args[1] args[2]

            Where args[0] is the first integer, args[1] is the desired operation (+,-), and
            args[2] is the second integer. Any other argument specified after that will
            be treated as nothing.

II.   Solution

    A. Overall design

        Once the user enters the three arguments, super solver takes the inputs and
        invokes the desired method on the two super integer objects.

    B. Test methodology

        No unit tests were written. I simply spent to much time on the main part of the
        program, I couldn't write the tests.

    C. Key implementation Details

        1. All of the operations are methods in the SuperIntegers.java class.
        2. I treated a super integer as a String object, like for the changemaker
            project, I treated a tuple as an Array object.
        3. In doing so, I created multiple methods to get information from the super
            integers.
        4. Also, an important part to all of my arithmetic is my conversion from
            decimal to binary and then from binary to decimal. I created two methods
            for these ideas that all return and have SuperInteger type parameters.

5. NOTE: When invoking the multiplication method, you MUST use an 'x' instead of a '*'. That is how it worked when I was coding, so I kept it like that.

D. Known bugs

At this point, I believe, that my Addition, Subtraction, and Multiplication methods are bug-free. However, there are some problems with my Division and Remainder methods. They work for some types of numbers and I tried to figure out what, but they still came up with bugs. Also,

III. Conclusions

Overall, I was able to successfully integrate my add, subtract, and multiply methods. By far, the multiply method was the easiest to integrate, thanks to Russian peasants multiplication. I wish there was some algorithm, like Russian peasants division, it would have made my life way easier. Anyways, I tried my best on re-doing this project. I made some improvements, but ultimately, it is not exactly what you wanted.