

Big number library

Description

In this challenge you will be using arrays to extend the integer data type to work with numbers of at most 1000 digits. There are many ways to solve these problems and some are far more efficient than others. You must decide how you are going to store and operate on the numbers.

Assume that you have two integers, N and M , that are at most 1000 digits stored in the program. You may wish to generate them for testing purposes according to simple rules. For example, every digit is 1 for both numbers. Solve as many of the following problems as you can. We suggest solving them in order.

- 1) Add N and M . You may assume that N and M are both positive
- 2) Subtract M from N (that is, compute the difference $D = N - M$)
 - a. Assume that $M \leq N$ and both are positive
 - b. Handle the case where $M > N$ but both are still positive
 - c. Handle the case where either N or M is negative, or both are
- 3) Multiply N and M
 - a. Assume that both N and M are positive
 - b. Handle the case where either N or M is negative, or both are
- 4) Divide N by M (that is, compute the product $Q = N/M$). You should follow the rules for integer division here: all decimals are truncated.
 - a. Assume that both N and M are positive and that $M \leq N$
 - b. Handle the case where $M > N$ (Think about it, is this one hard?)
 - c. Handle the case where either N or M is negative, or both are

A few notes

- Think very carefully about what all of these operations are, really. For instance, if you are adding two numbers, what is the best way to handle a carry?
- Division is actually pretty difficult to do efficiently. You may want to revert to the grade school method that you learned until it works, and then work on making it more efficient...
- In general, these programs will be fairly difficult to test, since you'd have to know what the answers are for sets of large input integers. Part of the fun of this challenge is coming up with ways to test the programs without having to type in 2000 digits but in such a way that you still know that you are right.