CSCI 1320 (Principles of Computer Science I), Fall 2014
Homework 7

Credit: 10 points.

1 Reading

Be sure you have read chapter 10.

2 Programming Problems

Do the following programming problems. You will end up with at least one code file per problem. Submit your program source (and any other needed files) by sending mail to `bmassing@cs.trinity.edu`, with each file as an attachment. Please use a subject line that mentions the course and the assignment (e.g., "csci 1320 homework 7" or "CS1 hw7"). You can develop your programs on any system that provides the needed functionality, but I will test them on one of the department's Linux machines, so you should probably make sure they work in that environment before turning them in.

1. (10 points) Your mission for this problem is to write a Scala program that defines and tests a case class for a simple implementation of rational numbers. The program should include a definition for a case class `Rational` and the following functions:

   - `stringFor`, taking a `Rational` as input and returning a `String`.
   - `add`, `subtract`, `multiply`, and `divide`, each taking two `Rational`s as input and returning a `Rational` with the obvious(?) value.

   It can also include, for extra credit, either or both of the following functions:

   - `lowestTerms`, taking a `Rational` as input and returning a `Rational` with the obvious(?) value.
   - `compare`, taking two `Rational`s `r1` and `r2` as input and returning an `Int` with a value of -1 if `r1 < r2`, 1 if `r1 > r2`, and 0 if `r1 == r2`.

   The program should prompt for numerators and denominators for two rational numbers and print the result of calling the four arithmetic-operation functions and any optional functions. (It should just print an error message if either denominator is zero.) Sample output for inputs 1, 2, 6, and 10:

   ```
   r1 = 1/2
   r2 = 6/10
   r1 in lowest terms = 1/2
   r2 in lowest terms = 3/5
   r1 + r2 (lowest terms) 11/10
   r1 - r2 (lowest terms) -1/10
   r1 * r2 (lowest terms) 3/10
   r1 / r2 (lowest terms) 5/6
   r1 < r2
   ```

Notes/hints:

- To get full credit the program must not use floating-point numbers.
- Be sure to think about what should happen if numerator or denominator or both are negative. To get full credit `stringFor` should print at most one minus sign.
- The obvious(?) way to reduce a rational number to lowest terms is to find the greatest common divisor (GCD) of the numerator and denominator. Euclid's algorithm is one way of doing that, and it can be implemented using recursion based on the following definition:

  For non-negative integers a and b, not both zero, with $a \geq b$,

  $$\gcd(a, b) = \begin{cases} a & \text{if } b = 0 \\ \gcd(b, a \bmod b) & \text{otherwise} \end{cases}$$

  where $a \bmod b$ is the remainder when a is divided by b.