CSCI/CMPE 1370

Lab 06

The Problem

Write a program to do fractional arithmetic. To keep it a little shorter, we'll just do the subtraction and multiplication operations. It should also reduce the answer to its lowest form (e.g. 3/9 should be 1/3). It should not print silly things like 0/5 (should just be 0) or 7/1 (should just be 7). The user should be allowed to enter any number of problems.

A sample dialogue follows. The user responses are in gray:

```
Fractional Arithmetic Program

This program will perform arithmetic on fractions.
Problems should be entered like this: 2/5 - 4/7

The allowed operations are subtraction (-) and multiplication (*).

Please enter your problem => 1/2 - 1/4
The answer is: 1/4
Would you like to do another one? (y/n): y

Please enter your problem => 2/9 * 3/4
The answer is: 1/6
Would you like to do another one? (y/n): n

Goodbye.
```

Some Rules

1. You must accept fraction equations from the user as above! No prompting for each number or operator individually. You can assume that the user always enters two valid fractions in the format `x/y` (where x and y are whole numbers and y is not zero), with a single character operator between them (- or *).

2. **This program deals exclusively with fractions, and each fraction must be stored as 2 whole numbers. Therefore, it should use integer variables (no float or doubles!).**

3. Your program must be made using functions. You must have at least these eight functions:

   - A function to read in an equation
   - Functions to perform the two operations:
     - Subtracting one fraction from another
     - Multiplying two fractions
   - A function to reduce a fraction
     - ...which uses another function to find the GCD (see below)
   - A function to print a fraction (the answer)

A Little Help Here

To reduce a fraction, you will need to find the greatest common denominator (GCD) of the numerator and denominator. The GCD of two integers is the largest integer that will divide both numbers evenly. For example, the GCD of 18 and 27 is 9, since 9 is the largest integer to divide evenly into both 18 and 27. Since 9 goes into 18 twice and into 27 three times, the fraction 18/27 can be reduced to 2/3.

The Greek mathematician Euclid (ca. 300 BC) is credited with coming up with a simple, efficient algorithm for finding the GCD of two integers. Implement this algorithm as a function with the heading:

```
int gcd( int x, int y )
```

The Euclidean Algorithm in "pseudocode":

1. let A and B be two positive integers
2. let R = A % B
3. while R != 0
   - A = B
   - B = R
   - R = A % B
4. B is the gcd

For example, consider finding the gcd of 24 and 60:

- A = 24, B = 60 — 24/60 is 0 with remainder R = 24
- set A = 60, B = 24 — 60/24 is 2 with remainder R = 12
- set A = 24, B = 12 — 24/12 is 2 with remainder R = 0
- So, the gcd is 12

Some Tips

Keeping a running program helps debugging tremendously. You will only be turning in your final version, but I recommend you start with a simpler problem and work your way up. Maybe start with the subtract and multiply functions (on hardcoded numbers) and worry about reducing and the user interface later.

Checklist

(You used all integers, no floats or doubles, right?)

The required functions:
Read in equation
Subtract
Multiply
Reduce
GCD
Print fraction

The user interface:
Properly responds to the different operators
Does problems until the user says to stop