A Divisive Issue

This is a **pair** exercise and must be competed in your **tutorial** or **lab** with your partner.

this activity was derived from *the Structure and Interpretation of Computer Programs*, by Abelson, Sussman, and Sussman; if you're interested in a very different take on computer science and on programming, you will enjoy this book!

The greatest common divisor, or GCD, of two integers a and b is the largest integer that divides both a and b with no remainder; for example, the GCD of 16 and 28 is 4 (as $16 = 4 \times 4$, and $28 = 4 \times 7$).

One way to calculate the GCD would be to totally factor both numbers, and find common factors, but there's a much faster and easier way to do it.

If r is the remainder when I divide a by b, then the common divisors of a and b are precisely the same as the common divisors of b and r, so the following identity holds:

$$\mathrm{gcd}(a, b) = \mathrm{gcd}(b, r)$$

If we start with any two positive integers, and applied this identity repeatedly, r will eventually become zero, and the other number in the pair is the greatest common divisor.

This is an amazing method known as Euclid's Algorithm, and is probably the oldest known non-trivial algorithm; it was first described in Euclid's *Elements* in around 300 BC.

For this exercise, you should create a file `gcdRec.c`, and implement a function

```
int gcdRec (int a, int b);
```

which implements Euclid's algorithm as described above.

To run some simple automated tests:

```
$ 1511 autotest gcdRec
```

To run Styl-o-matic:

```
$ 1511 stylomatic gcdRec.c
Looks good!
```

You'll get advice if you need to make changes to your code.

Submit your work with the *give* command, like so:

```
$ give cs1511 wk12_gcdRec
```

Or, if you are working from home, upload the relevant file(s) to the `wk12_gcdRec` activity on Give Online.