Finishing up gcd's.

Observation: if $a = qb + r$,
common divisors of a, b are the same
as those of b, r.

$\Rightarrow$ the following algorithm:

```
size_t gcd (size_t a, size_t b)
{
        if (b == 0) return a;
        return gcd(a, a%b);
}
```

Recall that $\gcd(a,b) = xa + yb$
for some $x, y \in \mathbb{Z}$.

How to find x, y?

E.g., $\gcd(7,3) = 1$

$$1 = \underset{\substack{\parallel \\ x}}{1} \cdot 7 - \underset{\substack{\parallel \\ y}}{2} \cdot 3 = 1$$

Application: modular inverses (useful in cryptography):

find x s.t. $x \cdot a \equiv 1 \mod n$

(in C++: $(x \cdot a) \% n = 1$)

$x \approx a^{-1}$

if $\gcd(a,n) = 1$, then $\exists\ x, y \in \mathbb{Z}$

s.t. $xa + yn = 1$.

But then $xa = 1 - yn$, so that

$$(xa \% n) = (1 - yn) \% n = 1$$

as desired!

(Generally,

$$(z + yn) \% n = z.)$$

Now for an algorithm.

Prototype: `int xgcd(int a, int b, int& x, int& y);`

Let $a = qb + r$, with $q = a/b$ and $r = a\%b$.

```
int x, y;
xgcd(a, b, x, y);
// now x, y set s.t.
//     xa + yb = gcd(a, b).
```

Suppose xgcd works on any smaller input (smaller value of b)

Then xgcd(b, r, x', y') will (with $r < b$)

set x', y' s.t. $\underset{\overset{\shortparallel}{(a,b)}}{(b,r)} = \underline{x'b + y'r}$.

Q: How are x', y' useful to find x, y for a, b?

A: Note that $a = qb + r$, so $r = a - qb$.

So, $\gcd(a,b) = x'b + y'r$

$$= x'b + y'(a - qb)$$

$$= y'a + x'b - y'qb$$

$$= \underbrace{y'}_{x} a + \underbrace{(x' - y'q)}_{y} b$$

```
int xgcd(int a, int b, int& x, int& y)
{
    if (b == 0) { // base case.
        x = 1;
        y = 0;
        // 1a + 0.0 = a = gcd. ✓
        return a;
    }
    int x', y';
    int q = a/b, r = a%b;
    int d = xgcd(b, r, x', y');
    // assuming xgcd worked, x'b + y'r = d.
    // as above:
    x = y';
    y = x' - y'q;
    return d;
}
```

---

xgcd(7, 3):

$(7, 3, \underset{x}{1}, \underset{y}{-2})$ $\qquad$ $\underline{q = 2, \ r = 1}$

$$\left(3, 1, \frac{0}{x'}, \frac{1}{y'}\right) \qquad q = 3, \; r = 0$$

$$\left(1, 0, \frac{1}{x''}, \frac{0}{y''}\right)$$