Remember gcd from last time:

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

Note: alternate characterization of gcd:

$$\gcd(a,b) = \min\{|xa+yb| \mid x,y \in \mathbb{Z}\}.$$

How to find such x,y s.t. $xa+yb = \gcd(a,b)$?

$\{x \mid \text{conditions}\ldots\}$ ≡ set of all x that satisfy conditions.

Ex: $\gcd(12,18) = 6 = \underset{x}{(-1)}\cdot 12 + \underset{y}{1}\cdot 18$

Let's see if we can modify our original gcd algo to find these values.

if $b==0$, $\gcd = a = 1a + 0\cdot b$

Say we know x', y' s.t. $\gcd(b,r) = x'b + y'r$.

$\gcd(b,r) = \gcd(a,b)$

How to find x,y s.t. $xa+yb = \gcd(a,b)$?

Recall: $a = qb + r$ So $a - qb = r$.

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

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

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

How to write this in C/C++?

output

```
int xgcd(int a, int b, int& x, int& y)
```

outputs!

```
{   if(b==0) {
      x = 1;
      y = 0;   <---
      return a;
    }

    // Now assume this thing works on
    // all smaller inputs (smaller values of b)

    int xx, yy;
    int r = a%b, q = a/b;

    int d = xgcd(b, r, xx, yy);
    // now  d = xx·b + yy·r
    x = yy;
    y = xx - yy*q;
    return d;
}
```

Ex: xgcd(12, 18, _, _)

|

xgcd(18, 12, _, _)

|

_, _

$\times gcd(6, 0, \underline{1}, \underline{0})$