

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. $xa + yb = \gcd(a, b)$?

$\{x \mid \text{conditions} \dots\} \equiv$ set of all x
that satisfy
conditions.

$$\text{Ex: } \gcd(12, 18) = 6 = \underbrace{(-1)}_x \cdot 12 + \underbrace{1}_y \cdot 18$$

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

$$\text{if } b == 0, \quad \gcd = a = 1a + 0b$$

$$\text{Say we know } x', y' \text{ s.t. } \underbrace{\gcd(b, r)}_{\gcd(a, b)} = \underline{\underline{x'b + y'r.}}$$

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

$$\boxed{\text{Recall: } a = qb + r} \quad \text{So } a - qb = r.$$

$$\begin{aligned} \text{So, } \underbrace{\gcd(b, r)}_{\gcd(a, b)} &= x'b + y'r \\ &= x'b + y'(a - qb) \end{aligned}$$

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

How to write this in C/C++?

```

int output xgcd(int a, int b, outputs! int& x, int& y)
{
    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 $d = xx \cdot b + yy \cdot r$

```

x = yy;

```

```

y = xx - yy * q;

```

```

return d;

```

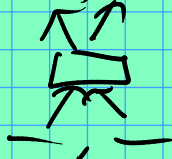
```

}

```

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

xgcd(18, 12, -, -)



$$\begin{array}{c} 1 \\ \times \gcd(6, 0, \overbrace{1, 0}^{\boxed{}}) \end{array}$$