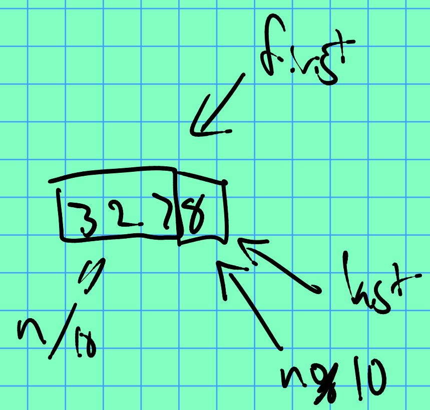
Exercise: without using loops, write a function (recursive) that prints an integer "vertically" to stdout. E.g., f(3278) would print

3
2
7
8



More interesting example: <u>Euclidean algorithm</u> for GCD's (greatest common divisor). Let's try to do this without exhaustive search.

Fact: for $a, b \in \mathbb{Z}$, then $d \mid a$ & $d \mid b$

$\Longleftrightarrow$ $d \mid b$ & $d \mid r$, where r is the remainder of a/b.

(Recall that we can write $a = qb + r$ where $\underline{r < b}$)

Proof of :

($\Longrightarrow$) Suppose $d \mid a$ & $d \mid b$. From the div. alg., know $a = qb + r$. ①

Recall: $x \mid y \equiv \exists z \in \mathbb{Z}$ with $y = zx$.

$d \mid a \Rightarrow \exists z \in \mathbb{Z}$ s.t. $a = zd$.

$d \mid b \Rightarrow \exists z' \in \mathbb{Z}$ s.t. $b = z'd$.

So ① becomes

$$zd = q(z'd) + r$$

$$\Rightarrow r = zd - qz'd$$
$$= d(z - qz')$$
$$\Rightarrow d \mid r. \quad \checkmark$$

($\Leftarrow$) Sp $d \mid b$ & $d \mid r$. Then

$$a = q(zd) + z'd$$
$$= (qz + z')d$$
$$\Rightarrow d \mid a.$$

The point: the common divisors of $\underline{a, b}$ are precisely the common divisors of $\underline{b, r}$.

Thus $\gcd(a,b) = \gcd(b,\underline{r})$.

$$r \lneq b.$$

So, if we define the "size" of the problem $\gcd(a,b)$ as $|b|$, then our recursive call to $\gcd(b,r)$ is indeed on a **strictly smaller input** as required.

Base case: second parameter $== 0$

Since everything divides 0, the answer for the base case will be the first parameter.

Now in C++:

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

"Euclidean Algorithm"

What if $a < b$?

Then $a/b = 0$

$a\%b = a$.

So, next call will be gcd(b,a).

"Extended" Euclidean Algorithm.

Note/recall that gcd(a,b) is an <u>integer</u> <u>linear</u> <u>combination</u> of a & b.

That is, $\gcd(a,b) = ua + vb$ where $u,v \in \mathbb{Z}$.

Example: $\gcd(2,5) = 1 = \boxed{-2}\,2 + \boxed{1}\cdot 5$

Exercise: write a function that computes $\gcd(a,b)$ & $u,v \in \mathbb{Z}$ s.t. $\gcd(a,b) = ua + vb$.

Prototype in C++:

```
int xgcd(int a, int b, int& u, int& v);
```

$$\underset{\gcd(a,b)}{\texttt{int}}\ \texttt{xgcd}(\underbrace{\texttt{int a, int b}}_{\text{inputs}},\ \underbrace{\texttt{int\& u, int\& v}}_{\text{Out-puts}});$$

How to solve it? Again we'll use recursion.

Base case: $b == 0$. $\gcd(a,b) = ua + vb$.

Then $u = 1$, $v = 0$.

(Aside: could choose something other than 0 for v. Answer will still be correct, but different. See "Bezout Identity")

Time for recursive magic!

Suppose that on every smaller input, our xgcd gets the right answer. How to use this to find u, v for a, b?

Let $a = qb + r$ & say $d = \gcd(a,b)$.

q is a/b, r is $a \% b$

Say we set u', v' from $\text{xgcd}(b, r, u', v')$

Then what are the u, v for a, b?

$$d = u'b + v'r.$$

But note that $r = a - qb$.

So, $d = u'b + v'(a - qb)$

$$= \underbrace{v'}_{u = v'}a + \underbrace{(u' - qv')}_{v = u' - qv'}b$$