Exercise: for input integer $n$, compute how many times it is divisible by two.

E.g. if $n = 40 = 2^3 \cdot 5$, then the answer would be 3.

Warm up: how to tell if $n$ is even or odd? (i.e., if answer $> 0$ or not).

Answer: use % operator, which gives the remainder: `if (n%2 == 0)` // n is even.

Idea: keep dividing $n$ by 2 until it's odd. the count will be the answer.

```
int n;
cin >> n;
int count = 0;
while (n%2 == 0) {
    n = n/2; // shorthand: n/=2;
    count++;
}
cout << count;
```

Question: if `n%2 == 0` was replaced by `n > 0`, what (approximately) would we set?

$\approx \log_2 n$.

---

Example: compute the gcd of two integers. ("greatest common divisor")

Let's "brute force" it.

Lets assume $a \le b$. What are the possible values for $\gcd(a,b)$?

$\{1, 2, \ldots, a\}$.

Some building blocks:

How to tell if $d$ is a common divisor (in C++)?

`(a%d==0 && b%d==0)`

Idea: run through candidates backwards & output the first one that passes the test.

```
int a, b; // inputs.
cin>> a >> b;
// make sure a ≤ b:
if(a>b) {
    int temp = a; //save a.
    a = b;
    b = temp;
}
// now for sure a ≤ b.
for(int d = a; !(a%d==0 && b%d==0); d--) {
}
cout << d;
```

once: `int d = a;` — boolean: `!(a%d==0 && b%d==0)` — update: `d--`

Note: this won't compile unless you declare d outside the for loop.