Day 9: Recursion!

Problem Statement

Welcome to Day 9! Check out this video on recursion, or jump right into the problem.

Euclid's Algorithm for Computing the GCD of two integers

Given two integers, x and y, their GCD (greatest common divisor) can be calculated recursively using Euclid's Algorithm, which essentially says that if x equals y, then $GCD(x,y) = x$; otherwise, $GCD(x,y) = GCD(x-y, y)$ if $x \gt y$. Note that this logic can be further optimized for a more efficient implementation.

Given the starter code in your editor, complete the function body so it returns the GCD of two input integers, x and y.

Input Format

Two space-separated integers, x and y.

Constraints
$1 \le x,y \le 10^{6}$

Output Format

Print the GCD of x and y as an integer.

Sample Input

```
1 5
```

Sample Output

```
1
```

Explanation

We are given $x=1$ and $y=5$. This explanation uses the subtraction implementation mentioned in the problem description, and is outlined in pseudocode below:

```
int GCD(x,y):
    If x equals y, return x;
    Else, return GCD(x',y'), where x' = MAX(x,y) - MIN(x,y) and y' = MIN(x,y).
```

GCD(1,5): $1 \neq 5$, so return a call to $GCD(5-1, 1)$.
GCD(4,1): $4 \neq 1$, so return a call to $GCD(4-1, 1)$.
GCD(3,1): $3 \neq 1$, so return a call to $GCD(3-1, 1)$.
GCD(2,1): $2 \neq 1$, so return a call to $GCD(2-1, 1)$.
GCD(1,1): $1 = 1$, so we return x (which is 1).

The final return is passed back through the call stack as the return value for the original call. That is to say, $GCD(1,1)$ returns 1 to $GCD(2,1)$, the function that originally called it. $GCD(2,1)$ then returns it to $GCD(3,1)$, which returns it to $GCD(4,1)$, which returns it to $GCD(1,5)$. Thus $GCD(1,5)$ returns a value of 1, which we print as our answer.

Note: The algorithm used here is merely demonstrative and can be further optimized.