# Computing the GCD

**Objective**

In this challenge, we learn how to compute GCD using the Euclidean algorithm.

**Resources**

Here's a helpful video on the topic:

Given two integers, $x$ and $y$, a recursive technique to find their GCD is the Euclidean Algorithm.

The algorithm states that, for computing the GCD of two positive integers $x$ and $y$, if $x$ and $y$ are equal, $GCD(x, y) = x$. Otherwise $GCD(x, y) = GCD(x - y, y)$ if $x > y$. There are a few optimizations that can be made to the above logic to arrive at a more efficient implementation.

**Task**

Given the starter code, you need to complete a function body that returns the GCD of two given integers $x$ and $y$.
The task of reading in input and printing the output will be handled by us.

**Programming Language Support**

At this point of time, we have a template for Scala. This means that we provide the code required to accept the input and display the output.

**Input Format**

One line of input containing $2$ space separated integers.

**Constraints**

$1 \leq a, b \leq 10^6$

**Output Format**

Output one integer, the GCD of the two given numbers.

**Sample Input**

```
1 5
```

**Sample Output**

```
1
```

### Explanation

### Sample Return Values:

```
GCD(1,5) = 1
GCD(10,100) = 10
GCD(22,131) = 1
```