

Copilot

Sure! Let's solve this problem step-by-step.

The **GCD (Greatest Common Divisor)** of two positive integers A and B is the largest positive integer that divides both A and B without leaving a remainder.

One of the most efficient ways to find the GCD of two numbers is by using the **Euclidean algorithm**. Here's how it works:

1. **Divide A by B** and get the remainder, R.
2. **Replace A with B** and B with R.
3. **Repeat the process** until B becomes 0. The GCD is the last non-zero remainder.

Let's write a Python function to implement this algorithm:

```
def gcd(a, b):  
    while b != 0:  
        a, b = b, a % b  
    return a
```

Here's a breakdown of the code:

1. **Define the function** gcd(a, b) which takes two arguments, a and b.
2. **Use a while loop** to continue the process as long as b is not zero.
3. **Inside the loop**, update a to be b, and b to be the remainder of a divided by b (a % b).
4. **When the loop ends**, return a as the GCD.

Let's see an example of how this function works:

```
A = 48  
B = 18  
print("The GCD of", A, "and", B, "is", gcd(A, B))
```

When you run this code, it will output:

The GCD of 48 and 18 is 6

Feel free to ask if you have any questions or need further clarification!