

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 common algorithms to find the GCD is the **Euclidean algorithm**.

Euclidean Algorithm

The Euclidean algorithm is based on the principle that the GCD of two numbers also divides their difference. The steps are:

1. If B is 0, then the GCD is A.
2. Otherwise, replace A with B and B with the remainder of A divided by B.
3. Repeat the process until B becomes 0.

Implementation in R

Here's how you can implement the Euclidean algorithm in R:

```
# Function to find GCD of two numbers
gcd <- function(A, B) {
  while (B != 0) {
    temp <- B
    B <- A %% B
    A <- temp
  }
  return(A)
}

# Example usage
A <- 56
B <- 98
result <- gcd(A, B)
print(paste("The GCD of", A, "and", B, "is", result))
```

Explanation

1. **Function Definition:** We define a function gcd that takes two arguments, A and B.
2. **While Loop:** The loop continues until B becomes 0.
3. **Temporary Variable:** We store the value of B in a temporary variable temp.
4. **Update B:** We update B to be the remainder of A divided by B ($A \% B$).
5. **Update A:** We update A to be the value of temp.
6. **Return GCD:** Once B is 0, the value of A is the GCD.

Example

In the example provided, we find the GCD of 56 and 98. The function will output:

```
[1] "The GCD of 56 and 98 is 14"
```

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