

Consider the following programming problem:

Given two positive integers A and B,
find GCD of A and B.

Please provide a solution for this problem in the R language.

Vorschläge anzeigen ▾

Finding GCD in R using the Euclidean Algorithm

Here's an R function that calculates the greatest common divisor (GCD) of two positive integers using the Euclidean algorithm:

Code-Snippet

```
gcd <- function(a, b) {
  while (b != 0) {
    temp <- a % b
    a <- b
    b <- temp
  }
  return(a)
}
```

Verwende den Code [mit Vorsicht](#).



Explanation:

1. `gcd <- function(a, b)`: Defines a function named `gcd` that takes two arguments, `a` and `b`.
2. `while (b != 0)`: The loop continues as long as `b` is not zero.
3. `temp <- a % b`: Calculates the remainder of `a` divided by `b` and stores it in `temp`.
4. `a <- b`: Assigns the value of `b` to `a`.
5. `b <- temp`: Assigns the value of `temp` (the previous remainder) to `b`.
6. `return(a)`: Once the loop terminates, the function returns the value of `a`, which is the GCD.

Example usage:

Code-Snippet

```
# Find the GCD of 12 and 18
result <- gcd(12, 18)
print(result) # Output: 6
```

Verwende den Code [mit Vorsicht](#).



This function effectively implements the Euclidean algorithm to find the GCD of two positive integers in R.

Prompt hier eingeben



Gemini zeigt möglicherweise fehlerhafte Informationen an, z. B. zu Personen. Überprüfe daher die Antworten. [Datenschutz & Gemini-Apps](#)