

Copilot

Sure, I'd be happy to help! The code you provided is almost correct, but there's a small issue. The problem is that the values of A and B are swapped during the calculation, which means the final value of A is not the original value of A.

To fix this, we need to store the original values of A and B before the loop starts and then use these stored values when printing the result. Here's the corrected code with minimal changes:

```
find_gcd <- function(A, B) {  
  original_A <- A  
  original_B <- B  
  while (B != 0) {  
    temp <- B  
    B <- A %% B  
    A <- temp  
  }  
  cat("GCD of", original_A, "and", original_B, "is:", A, "\n")  
}  
  
A <- 56  
B <- 98  
gcd <- find_gcd(A, B)
```

Explanation:

1. **Storing Original Values:** We store the original values of A and B in `original_A` and `original_B` before the loop starts.
2. **Loop:** The loop continues until B becomes 0. Inside the loop, we use the Euclidean algorithm to find the GCD.
3. **Printing the Result:** After the loop, we print the GCD using the original values of A and B.

This way, the function correctly calculates and prints the GCD of the original values of A and B. Let me know if you have any questions or need further clarification!