SHERLOCK Debug Report

Error: RecursionError

Root Cause

The `factorial` function is missing a base case. For a recursive function to terminate, there must be a condition under which the function stops calling itself and returns a direct value. In this implementation, `factorial(n)` always calls `factorial(n-1)`, leading to an infinite sequence of calls (e.g., `factorial(5)` -> `factorial(4)` -> ... -> `factorial(0)` -> `factorial(-1)` -> `factorial(-2)` ...). This continues until the Python interpreter's maximum recursion depth is exceeded.

Fix

To fix this, a base case needs to be added for `factorial(0)`. By mathematical definition, the factorial of 0 is 1. When `n` reaches 0, the function should return 1 directly, without making another recursive call. Additionally, it's good practice to handle negative inputs, as factorial is typically defined only for non-negative integers.

Corrected Code

```
def factorial(n): if n < 0: raise ValueError("Factorial is not defined for negative numbers") elif n == 0: return 1 else: return n * factorial(n-1)
```

Confidence: 100%