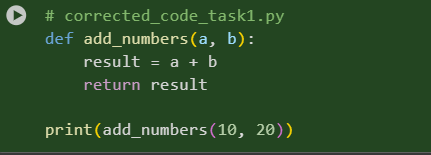
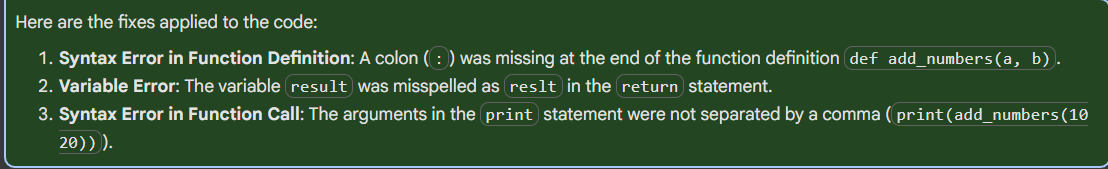
Assignment 10.4

Task-1:

Code:



Explanation:



Result:

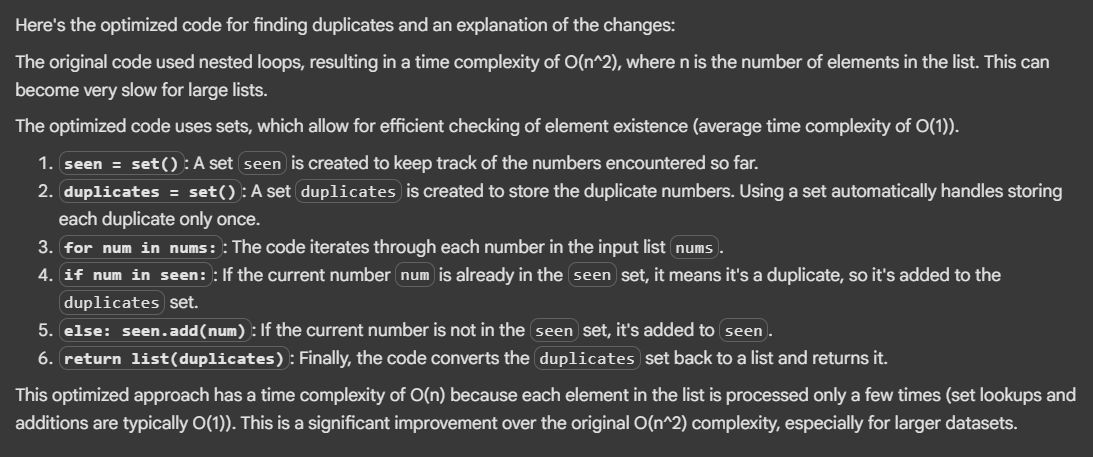


Task-2:

Code:



Explanation:

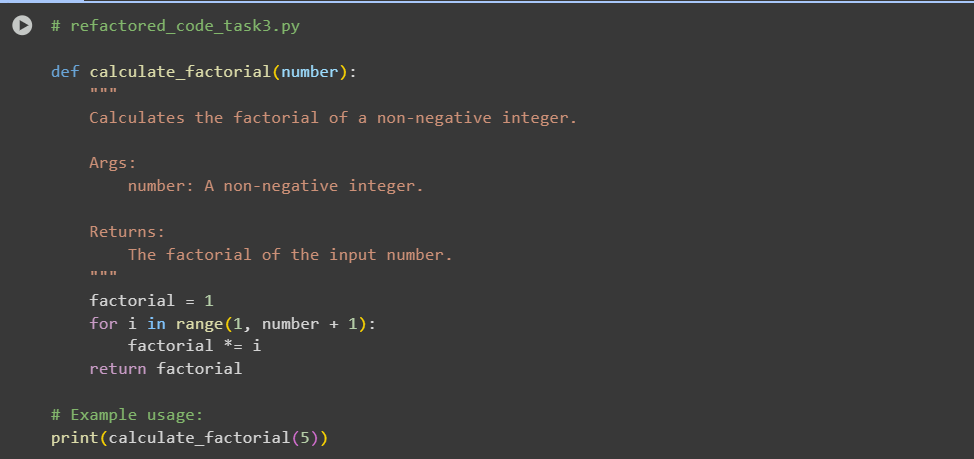


Output:

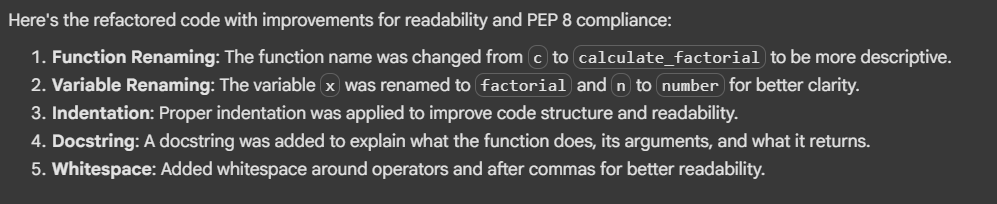


Task-3:

Code:



Explanation:

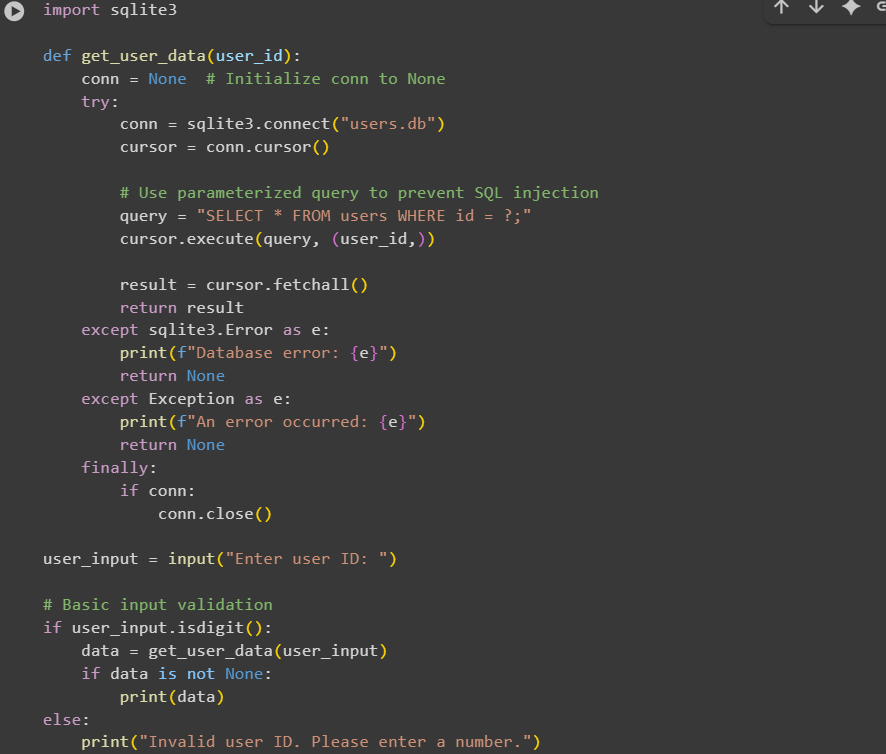


Output:

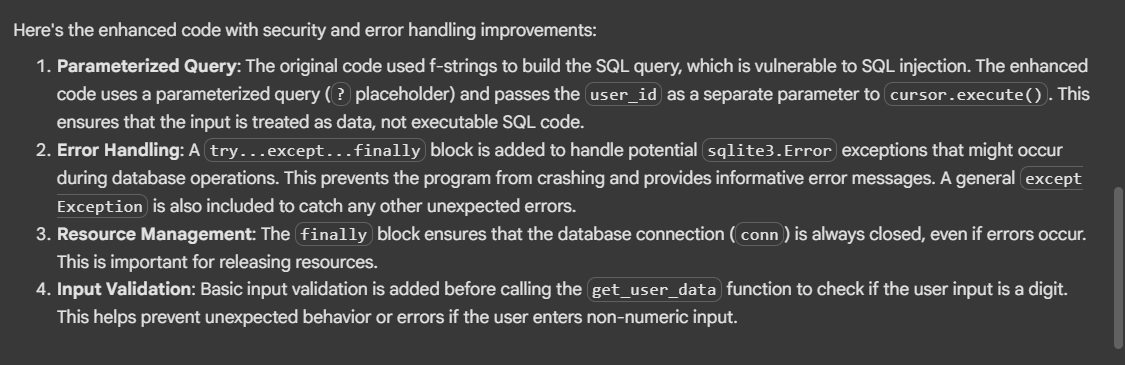


Task-4:

Code:



Explanation:

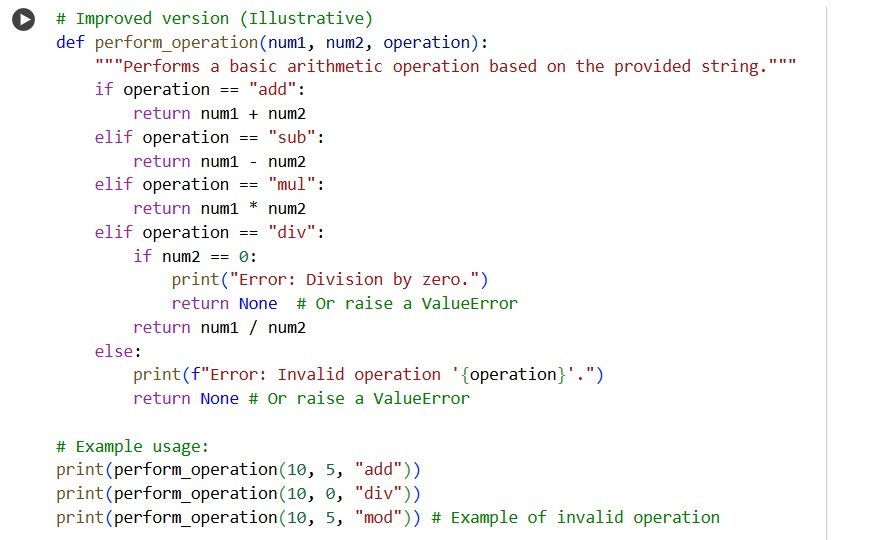


Result:



Task-5:

Code:



Explanation:

The original code had a few areas that could be improved for readability, maintainability, and robustness:

Poor Naming: The function name calc and the parameter names x, y, and z are not descriptive. It's hard to tell at a glance what the function does or what the parameters represent.

Inconsistent Formatting and Indentation: The indentation is inconsistent, especially in the elif z=="sub": return x-y line, which makes the code harder to read and understand the code structure. While Python is flexible with single-line statements, consistent indentation is crucial for readability and following PEP 8.

Lack of Error Handling for Division by Zero: The code does not handle the case where the operation is "div" and the second number (y) is zero. This would result in a ZeroDivisionError and crash the program.

Handling of Invalid Operations: For invalid operations, the code just prints "wrong". It would be better to provide a more informative error message and potentially return a specific value (like None) or raise an exception to indicate that the operation failed.

No Docstring: The function lacks a docstring, which is a good practice to explain what the function does, its parameters, and what it returns.

Output:  
