**Lab 10 – Code Review and Quality: Using AI to Improve Code Quality and Readability**

NAME – A.SHIVANI

HTNO – 2403A51400

BATCH NO – 16

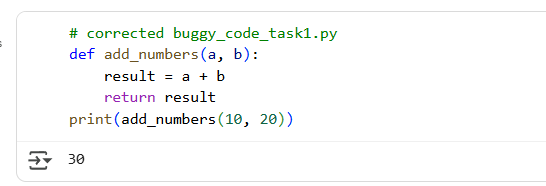
**Task 1: Syntax and Error Detection**

CODE –

bugged



Ai corrected –



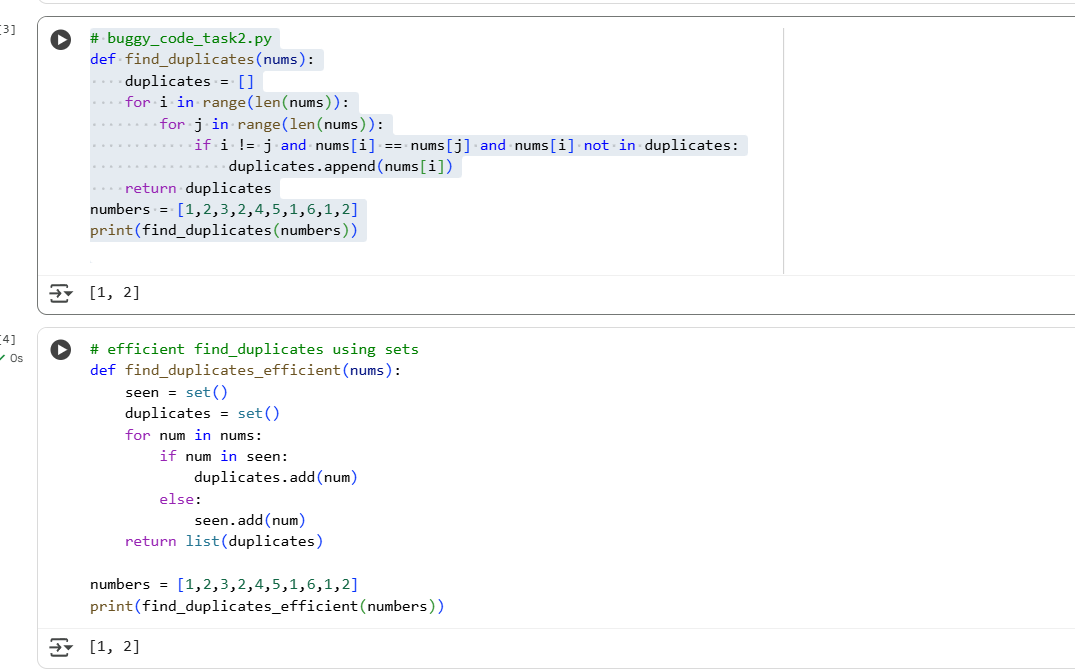
EXPLANATION –

The code defines a Python function called add\_numbers that takes two arguments, a and b. Inside the function, it calculates the sum of a and b and stores it in a variable named result. The function then returns the value of result. The last line of the code calls the add\_numbers function with the arguments 10 and 20, and the result of this function call (which is 30) is printed to the console.

**Task 2: Logical and Performance Issue Review**

CODE –

Bugged and Ai corrected –



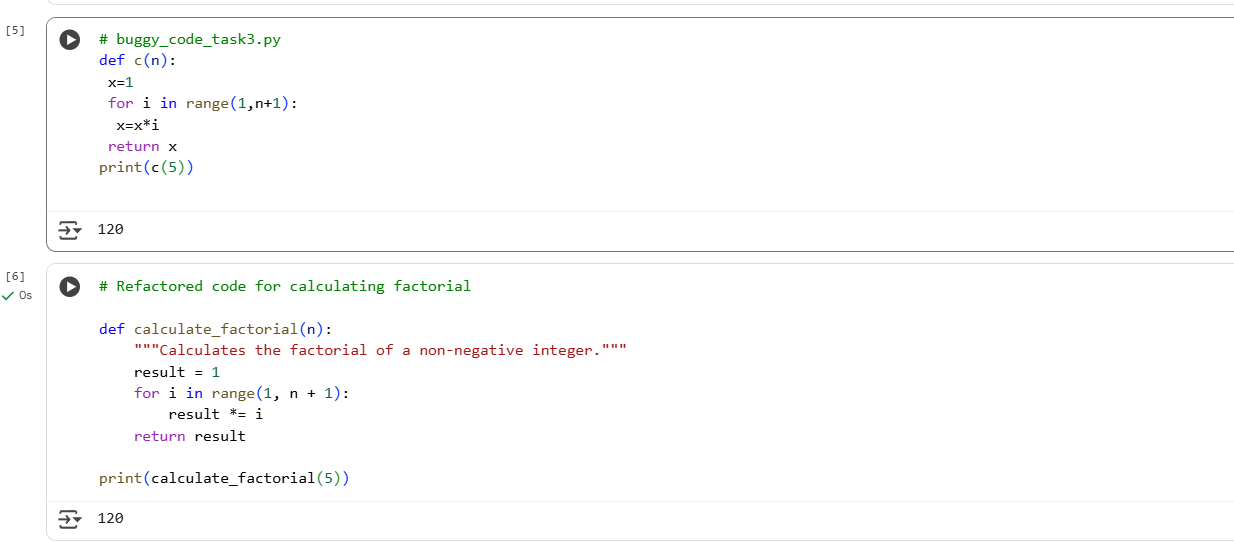
EXPLANATION –

This code defines a function called find\_duplicates\_efficient that takes a list of numbers as input. It uses two sets: seen to keep track of numbers encountered so far and duplicates to store the numbers that appear more than once. It iterates through the input list, and for each number, it checks if the number is already in the seen set. If it is, the number is added to the duplicates set because it's a duplicate. If the number is not in the seen set, it's added to the seen set. Finally, the function returns the elements of the duplicates set converted back into a list. This method is more efficient than nested loops because checking for an element's existence in a set is typically a very fast operation.

**Task 3: Code Refactoring for Readability**

CODE –

Bugged and Ai corrected –



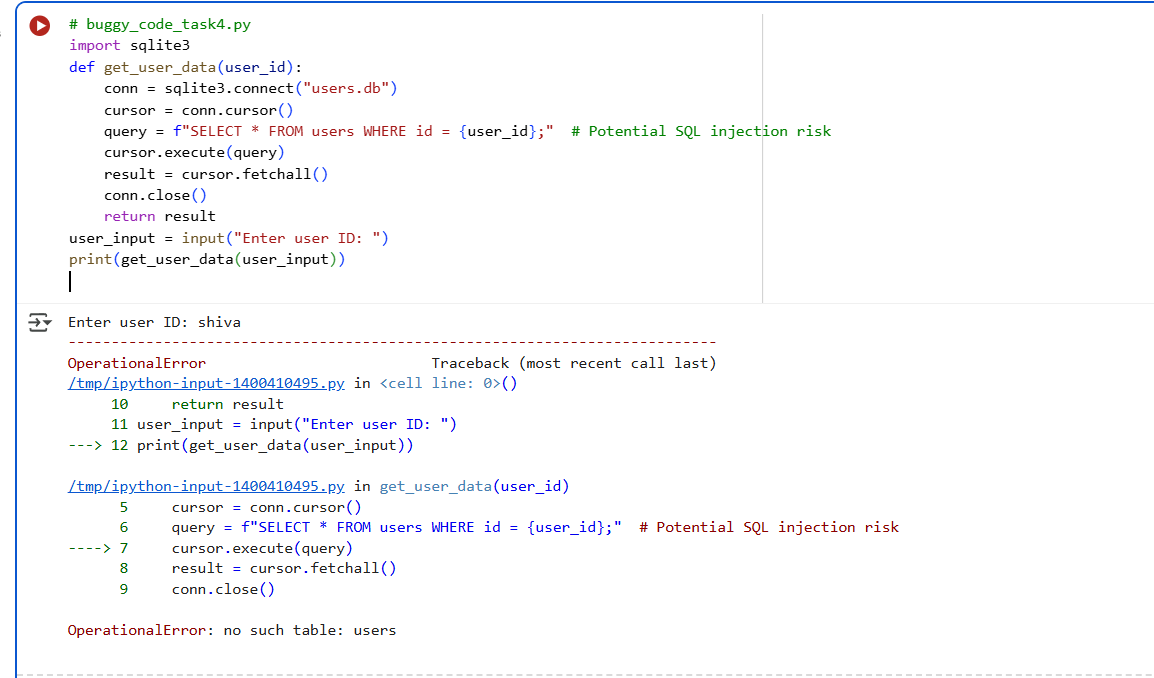
EXPLANATION –

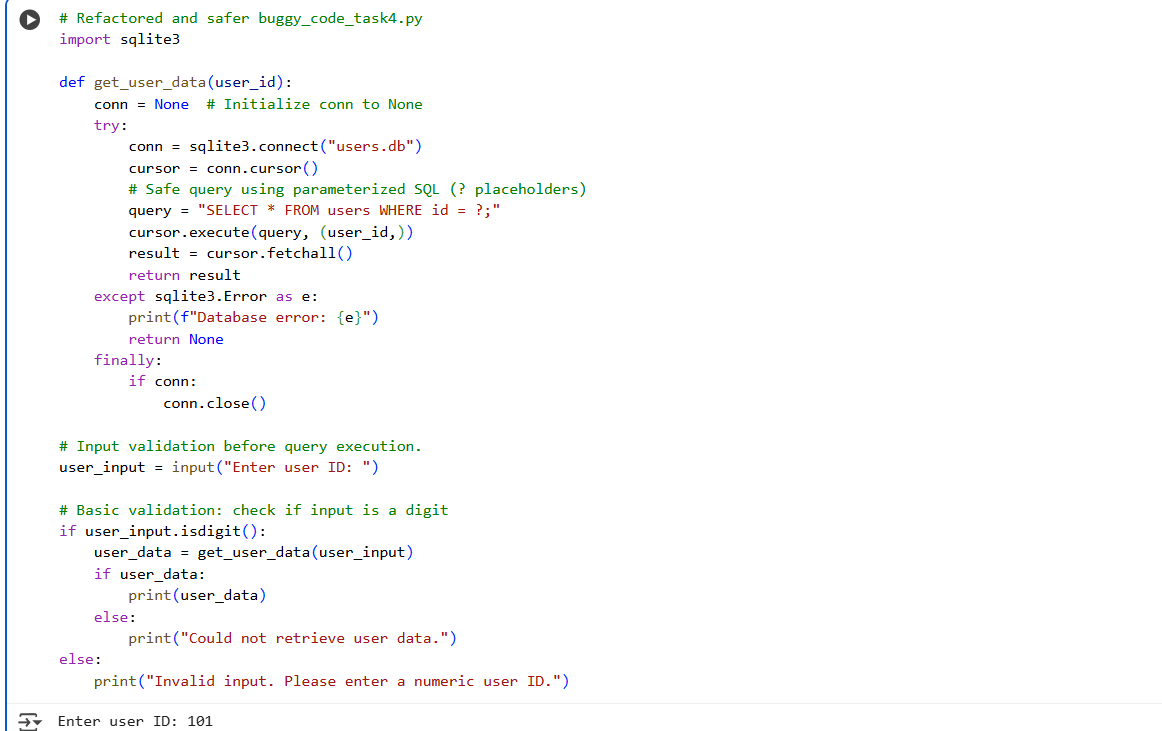
code calculates the factorial of a non-negative integer. The function is now appropriately named calculate\_factorial for better clarity. It includes a docstring that explains what the function does. The variable names x and c have been replaced with more descriptive names result and calculate\_factorial respectively. The code also follows standard Python indentation for improved readability. The core logic remains the same: it initializes a result to 1 and then multiplies it by each integer from 1 up to the input number n. Finally, it returns the calculated result. The print statement at the end calls this function with 5 and displays the result (which is 120).

**Task 4: Security and Error Handling Enhancement**

CODE –

Bugged and Ai corrected –





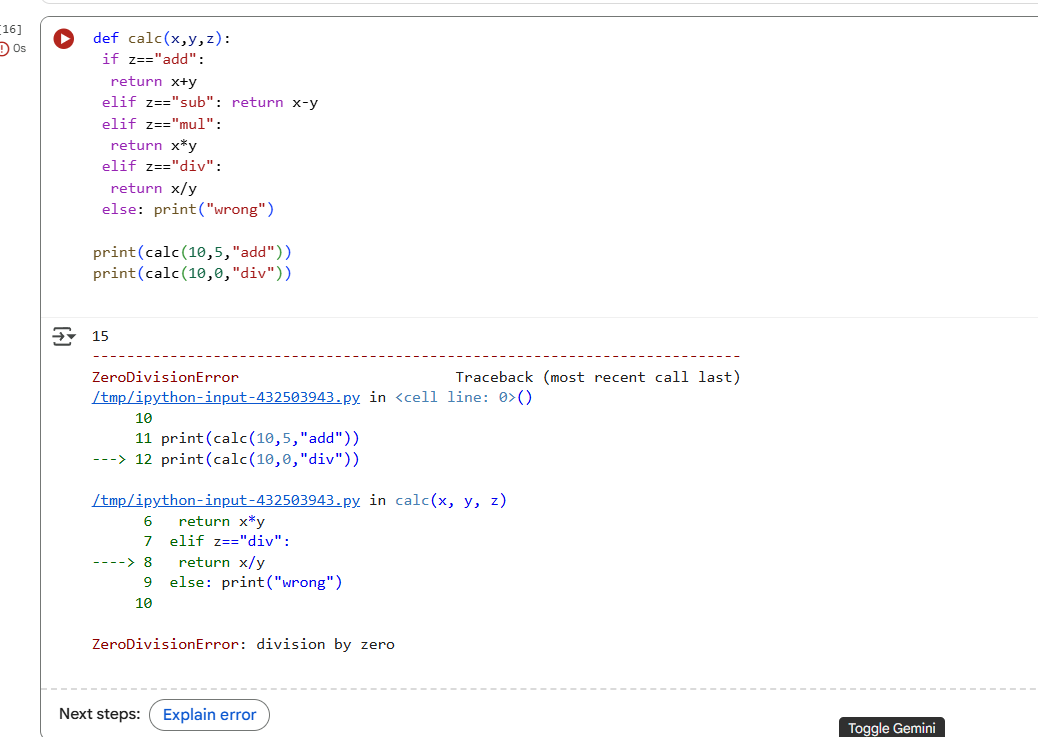
EXPLANATION –

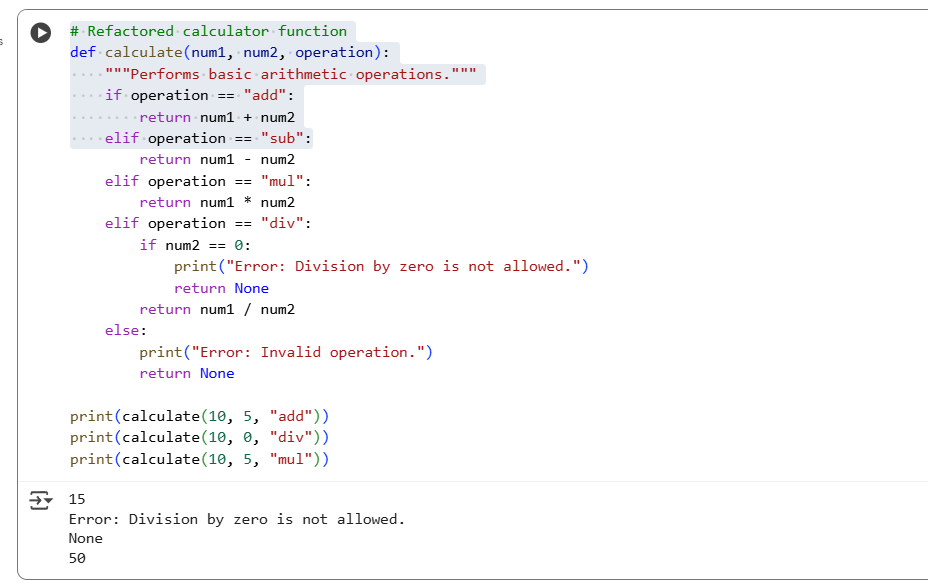
This code defines a Python function get\_user\_data that safely retrieves user data from a SQLite database named "users.db" based on a provided user ID. It uses a try-except-finally block to handle potential sqlite3.Error exceptions during database operations, ensuring that the database connection is always closed in the finally block. Crucially, it employs parameterized SQL queries (WHERE id = ?) instead of f-strings to prevent SQL injection vulnerabilities. Before calling the function, it prompts the user for an ID and performs basic validation to check if the input is a digit. Based on the validation and the result of the database query, it either prints the retrieved user data or an appropriate message indicating invalid input or failure to retrieve data.

**Task 5: Automated Code Review Report Generation**

CODE –

Bugged and Ai corrected –





EXPLANATION –

This code defines a function called calculate that takes three arguments: num1, num2, and operation. The function performs basic arithmetic operations based on the value of the operation argument. It includes conditional statements to check if the operation is "add", "sub", "mul", or "div". For division, it specifically checks if the second number (num2) is zero to prevent a division by zero error and prints an error message if it is. If the operation is not one of the supported operations, it prints an "Invalid operation" error message. The function returns the result of the calculation or None in case of an error. The print statements demonstrate how to use the function with different numbers and operations, including a case that triggers the division by zero error handling.