**Assignment:10.2**

**Name: Bollam Sathvika**

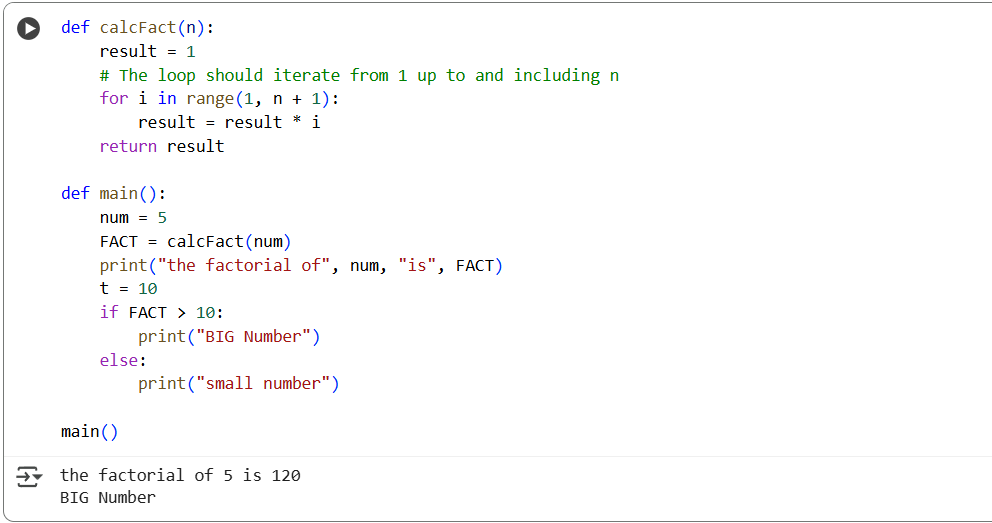
**Htno:2403A51344**

**Batch:14**

**#Task1**

**Prompt:** I have the following Python program with some basic errors. Please review the code, suggest corrections, and provide the corrected code with comments.

**Code and Output:**

****

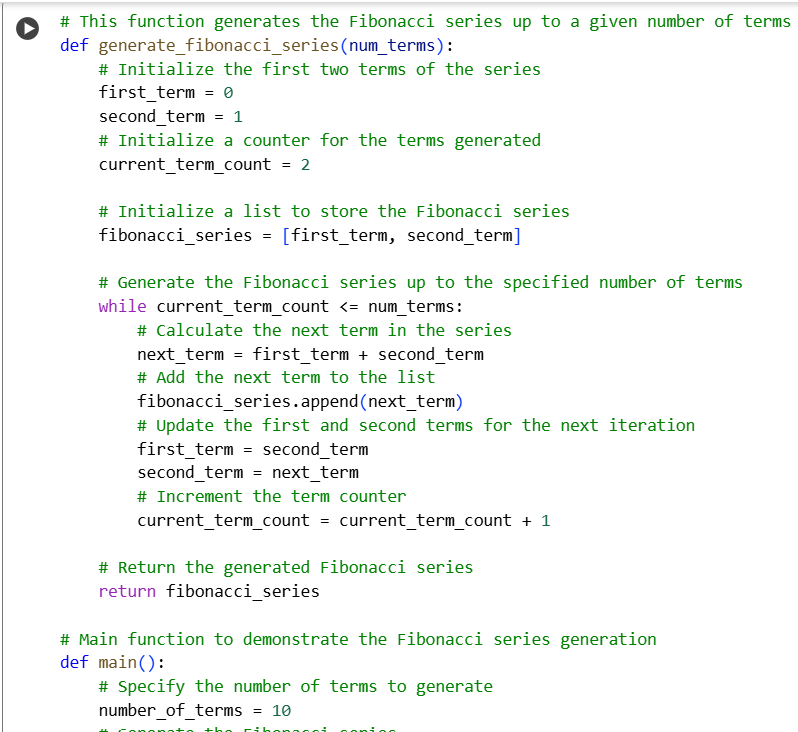
**Observation:**

In this program I observed that the loop range(1,n) was wrong because it did not include the last number, so the factorial result was incorrect. There was also an unused variable x and the variable name FACT was not in proper convention. After correcting these mistakes the program is working correctly and gives the output “The factorial of 5 is 120” and then prints “BIG Number” because the factorial is greater than 10.

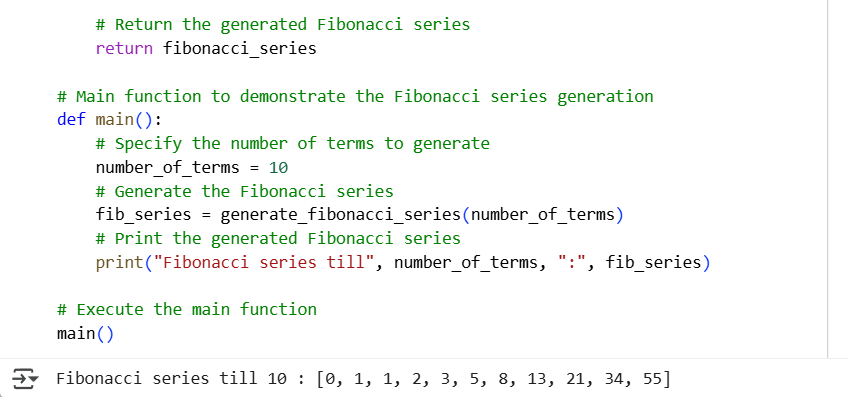
**#Task2**

**Prompt:** I have written a Python program to generate the Fibonacci series but my code has unclear variable names, no comments, and poor formatting. Please review my code, improve the variable names, add inline comments to explain each step, and apply proper PEP8 formatting so that the program looks clean and readable

**Code:**

****

**Output:**

****

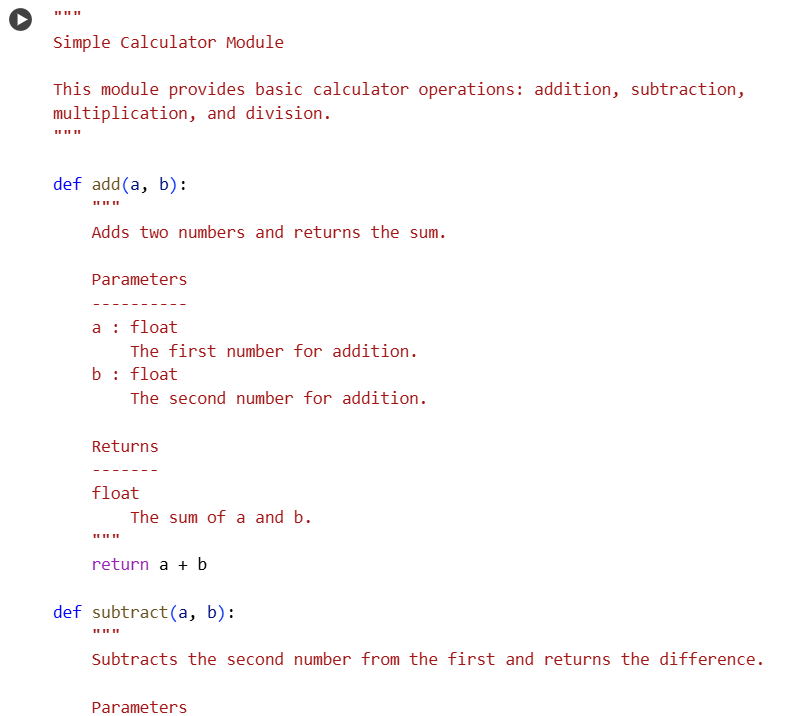
**Observation:**

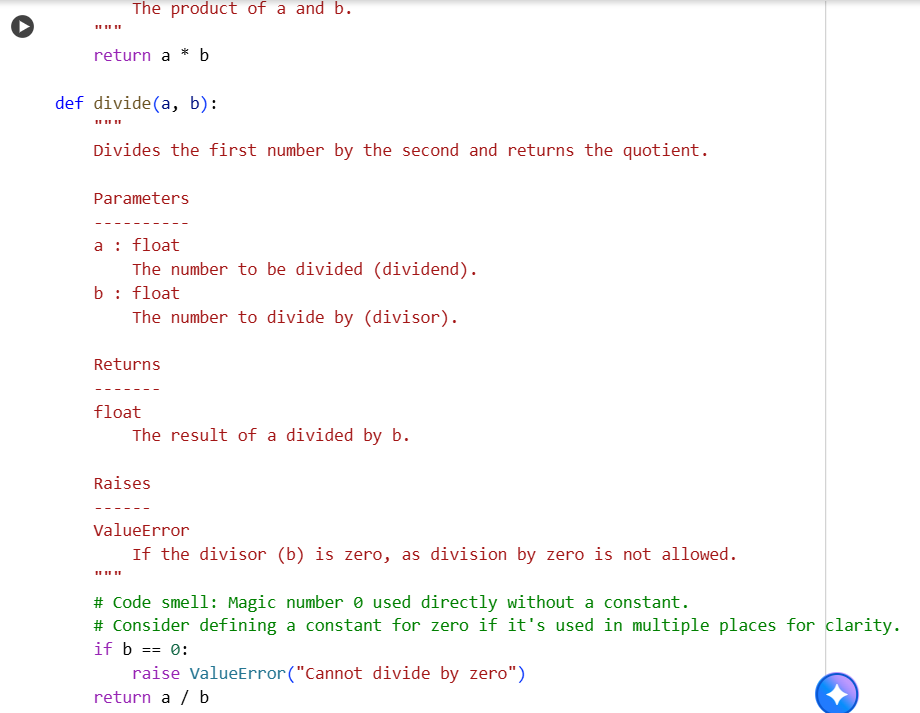
In this program, I observed that the original code had unclear variable names and no comments, which made it hard to understand. After improving the variable names, adding inline comments, and applying proper formatting, the program became much more readable and easier to follow, while still correctly generating the Fibonacci series up to the specified number of terms.

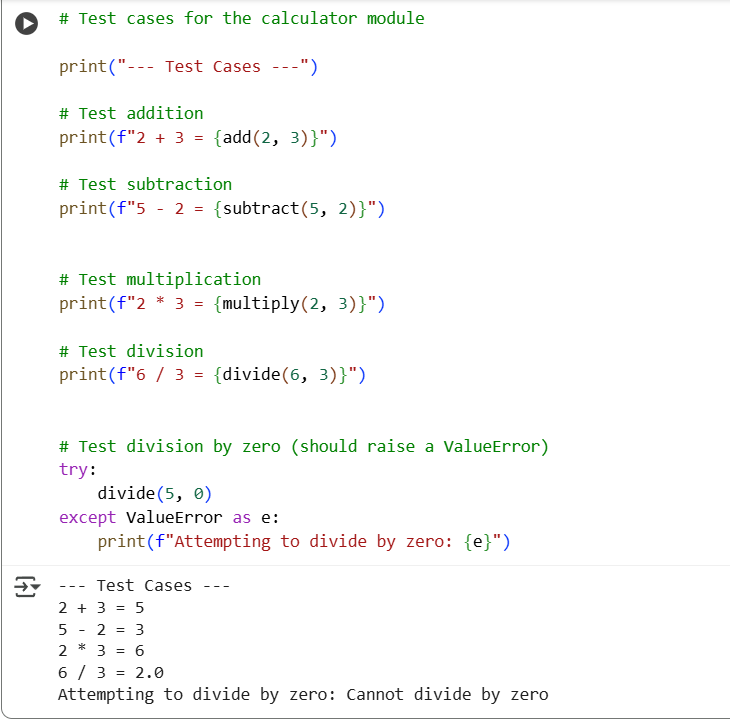
**#Task3**

**Prompt:** I have written a Python script with multiple functions (for example, a calculator with add, subtract, multiply, and divide). The code already has manual docstrings in NumPy style. Please review the code and generate an AI-assisted module-level docstring and individual function docstrings. Make the docstrings clear, concise, and readable. Also, point out if any code smells exist in the code such as long functions, duplicate code, poor naming, unused variables, magic numbers, or deep nesting

**Code :**

****

****

****

**Observation:**

In this program, I observed that the manual docstrings were more detailed and followed the NumPy style, clearly describing parameters, return types, and exceptions, which makes the code easier to maintain and document. The AI-generated docstrings were shorter and concise, making them quick to read, but they lacked detailed type information and structured sections. Both approaches improve readability, but the manual docstrings provide better guidance for professional or scientific projects.