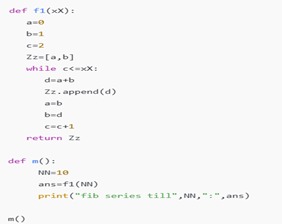
**ASSINGMENT-10.2**

**Name:**Srija Gattu

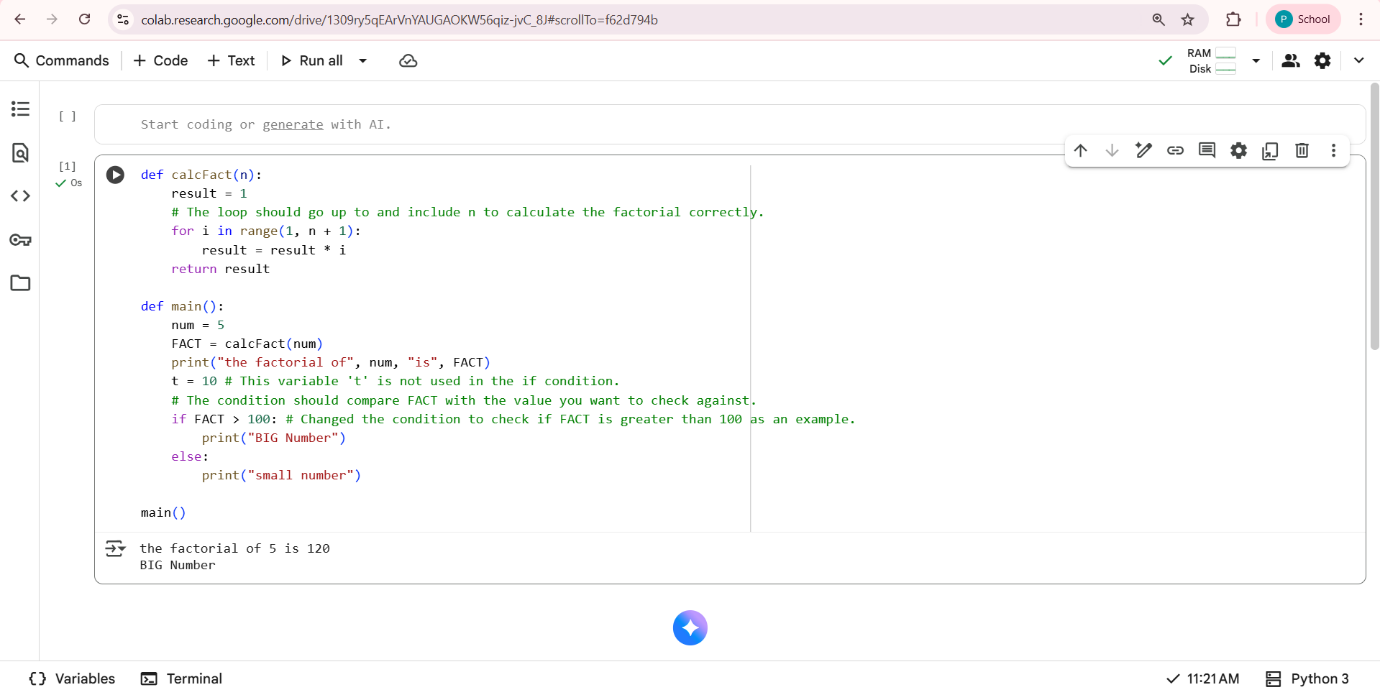
**HTNO:**2403A51320

**Batch:**13

**Task-1:**

• Write python program as shown below.  
• Use an AI assistant to review and suggest corrections.  


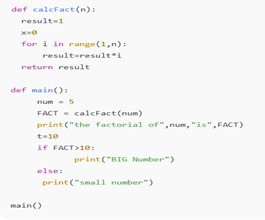
**Reviewed and suggested code**



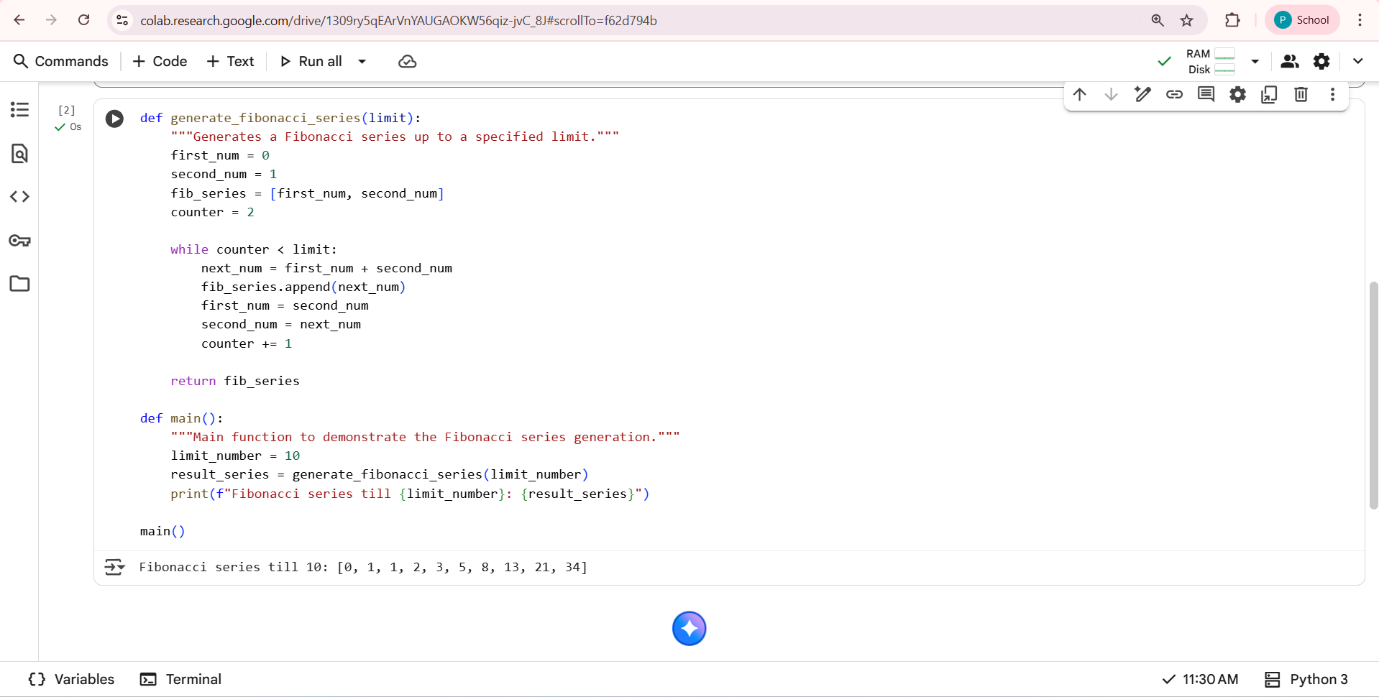
**Observation:**

* The function correctly calculates the factorial of the input number n.
* The loop iterates from 1 up to and including n.
* The main function demonstrates how to use calcFact and prints the result.
* The code includes an if condition that checks if the calculated factorial is greater than 100.

**Task-2:**Automatic Inline Comments  
• Write the Python code for Fibonacci as shown below and execute.  
• Ask AI to improve variable names, add comments, and apply PEP8 formatting(cleaned up)



**Ai-improved code**

****

**Explanation:**

**Improving variable names**: Using descriptive names like generate\_fibonacci\_series instead of f1, limit instead of xX, and fib\_series instead of Zz makes it clear what each part of the code represents.

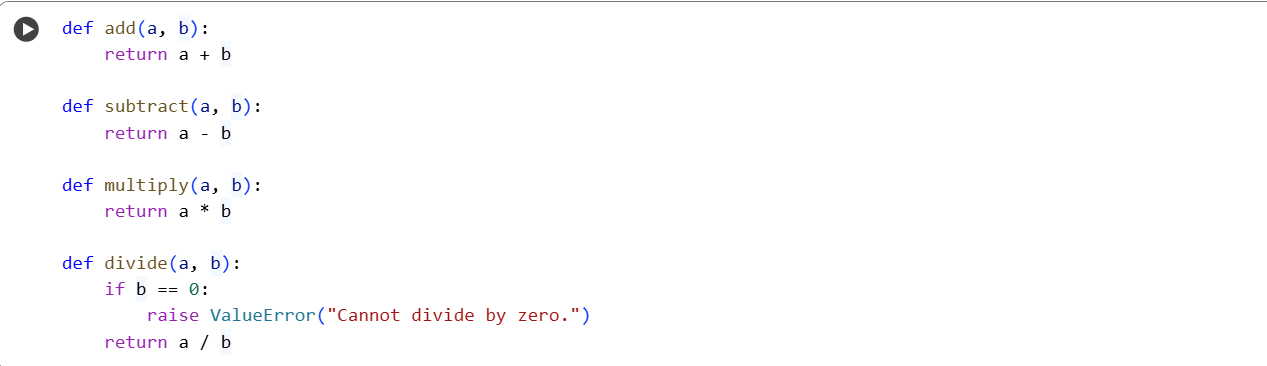
**Adding comments**: Comments explain the purpose of different sections of the code and clarify any non-obvious logic. This helps someone reading the code quickly grasp what's happening.

**Applying PEP8 formatting**: PEP8 is a style guide for Python code. Following it consistently (like using consistent indentation and spacing) makes the code visually cleaner and easier to scan, reducing cognitive load.

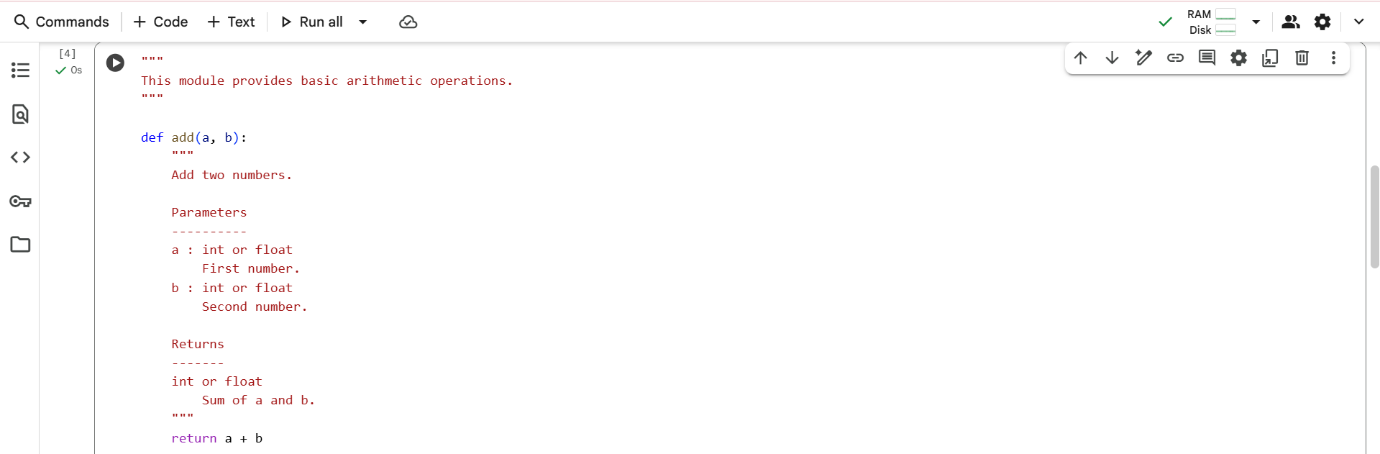
**Task-3:**

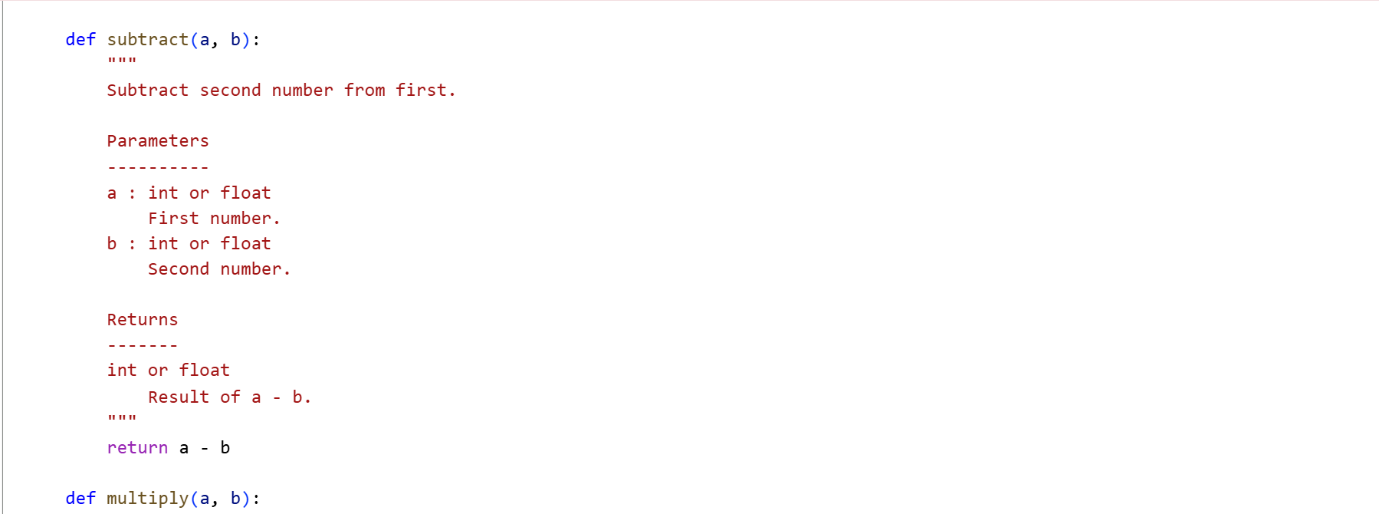
* Write a Python script with 3–4 functions (e.g., calculator: add, subtract, multiply,divide).
* Incorporate manual docstring in code with NumPy Style
* Use AI assistance to generate a module-level docstring + individual functiondocstrings.
* Compare the AI-generated docstring with your manually written one

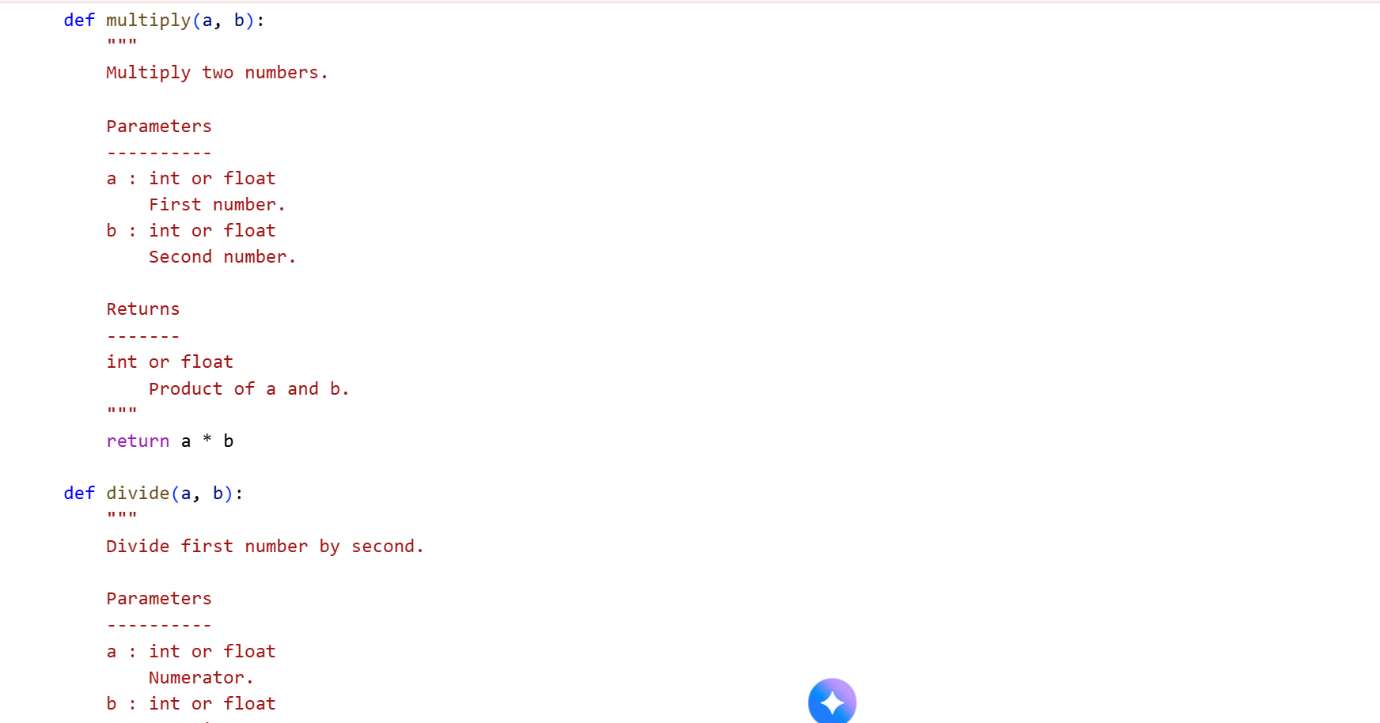
**Manual code:**

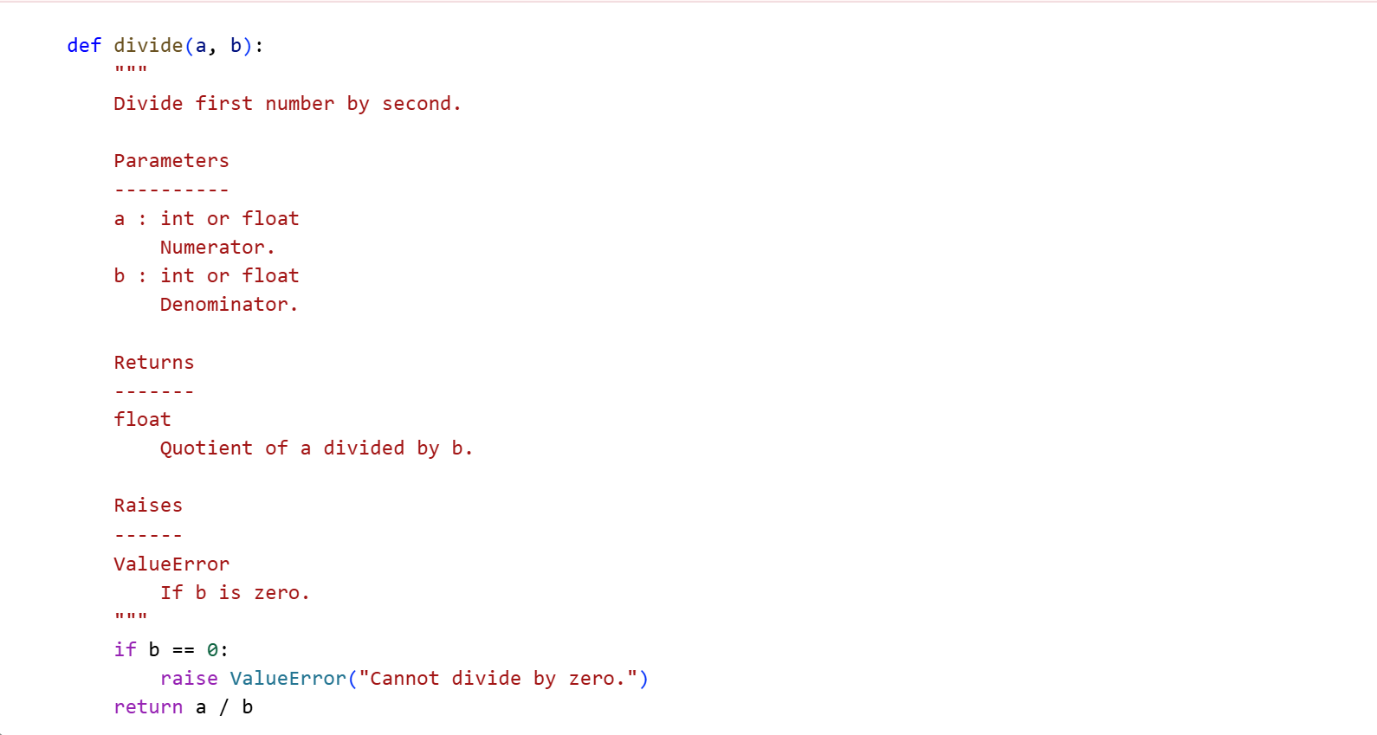
****

**Ai-generated code:**

****

****

****

****

**Comparision of both the codes:**

| Feature | User's Code Docstrings | Generated Code Docstrings |
| --- | --- | --- |
| Module Docstring | Missing | Present, describes module purpose |
| Function Docstrings | Present but incomplete (e.g., multiply, divide) | Present and complete (purpose, params, returns, raises) |
| Formatting | Inconsistent | Consistent (NumPy style) |
| Readability | Lower | Higher |
| Usability | Lower | Higher |