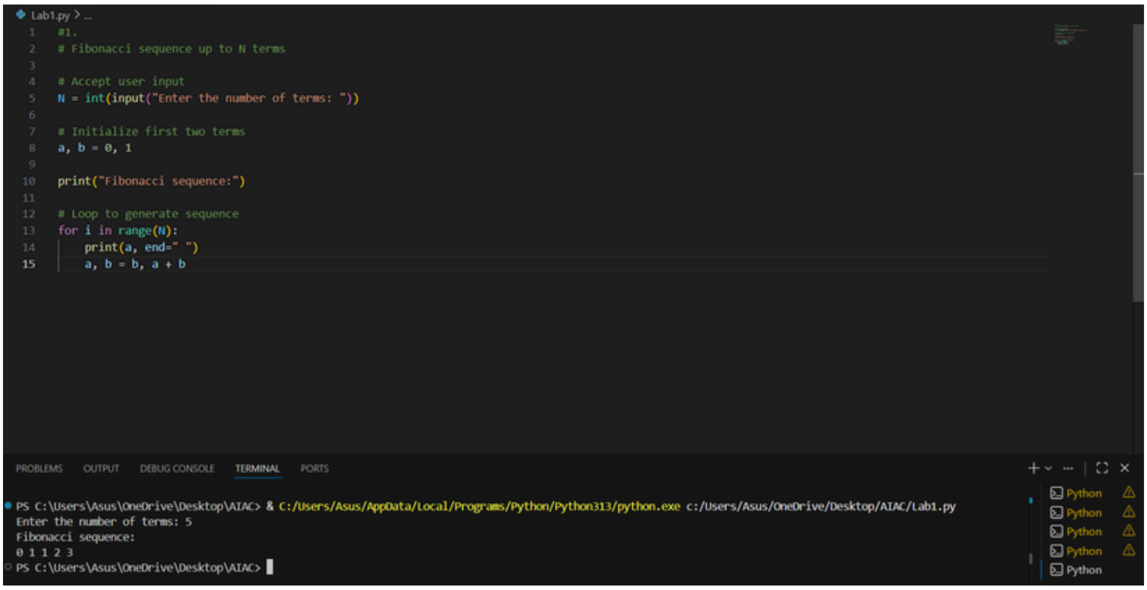
Assignment-1

Rollno:2303A54038
Batch:48

Scenario
You are asked to write a quick numerical sequence generator for a learning platform prototype.
❖ Task Description
Use GitHub Copilot to generate a Python program that:
➢ Prints the Fibonacci sequence up to n terms
➢ Accepts user input for n
➢ Implements the logic directly in the main code
➢ Does not use any user-defined functions



Prompt:you are a software engineer and you are given a task of printing Fibonacci sequence upto N terms.you should be able to accept the user input for N.Make sure to implement the logic directly in the main code and not to use any user defined functions

code-explanation:The code takes user input N and iteratively prints the Fibonacci sequence by updating two variables (a and b) each loop.It starts with 0 and 1, then repeatedly prints a and updates a, b = b, a + b until N terms are shown.

Scenario

The prototype will be shared with other developers and needs optimization.

❖ **Task Description**

➢ **Examine the Copilot-generated code from Task 1 and improve it by:**

➢ **Removing redundant variables**

➢ **Simplifying loop logic**

➢ **Avoiding unnecessary computations**

Prompt:Examine the code above and give me the optimized version of code by removing redundant variables and simplifying loop logic

Code explanationThe code initializes two variables x and y as the first Fibonacci numbers and iteratively prints x for N terms.After each print, it updates both variables in one step with x, y = y, x + y to generate the next term.

Scenario

The Fibonacci logic is now required in multiple modules of an application.

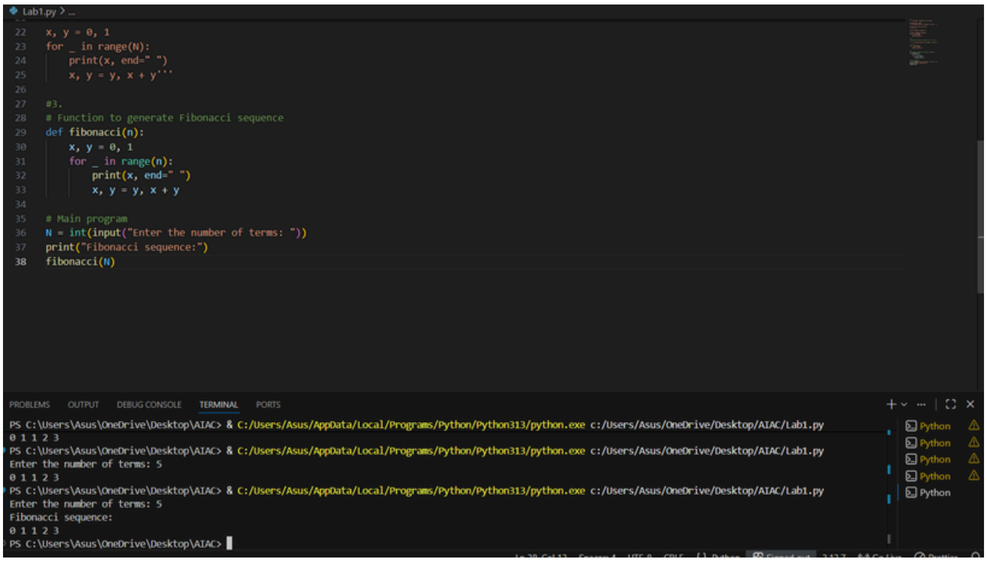
❖ Task Description

Use GitHub Copilot to generate a function-based Python program that:

➢ Uses a user-defined function to generate Fibonacci numbers

➢ Returns or prints the sequence up to n

➢ Includes meaningful comments (AI-assisted)



prompt:you are a software engineer and you are given a task to generate a function based python program that uses a user defined function to generate fibonnaci numbers upto N

Code-Explanation:This program defines a function fibonacci(n) that prints the Fibonacci sequence up to n terms by starting with 0 and 1 and updating both values in each loop.
The main block takes user input for N and calls the function, keeping the logic reusable and separate from user interaction.

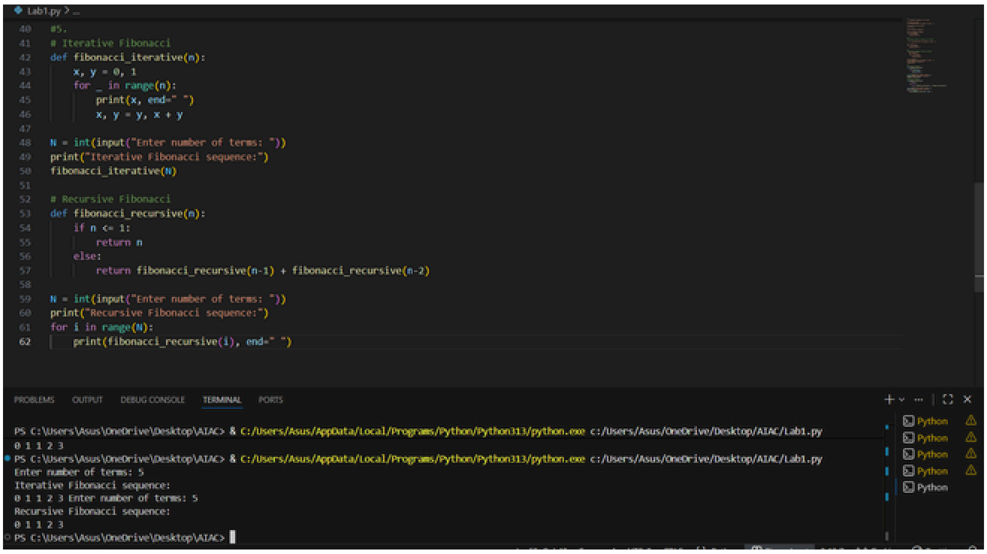
Scenario
You are participating in a code review session.
❖ Task Description
Compare the Copilot-generated Fibonacci programs:
➢ Without functions (Task 1)
➢ With functions (Task 3)
➢ Analyze them in terms of:

- **Code clarity**
- **Reusability**
- **Debugging ease**
- **Suitability for larger systems**

Prompt:you are code reviewer and you are participating in a code review session.you are tasked with comparing the codes of fibonnaci sequence without functions and with functions.you must provide a comparision table and make sure to involve categories like code clarity,code reusability ,debugging ease and suitability for larger systems

Category	Without Functions (Inline Code)	With Functions (User-Defined)
Code Clarity	Logic is directly in the main block, simple for beginners but can look cluttered.	Encapsulates logic in a function, making the main program cleaner and easier to read.
Code Reusability	Cannot be reused easily; the sequence logic must be rewritten if needed elsewhere.	Function can be called multiple times with different inputs, promoting reuse.
Debugging Ease	Debugging is harder since logic and execution are mixed together.	Easier to isolate and debug the function independently from the rest of the program.
Suitability for Larger Systems	Not suitable for larger projects; lacks modularity and separation of concerns.	Well-suited for larger systems; modular design allows integration, testing, and maintenance.

Scenario
Your mentor wants to assess AI's understanding of different algorithmic paradigms.
✣ Task Description
Prompt GitHub Copilot to generate:
An iterative Fibonacci implementation
A recursive Fibonacci implementation



Prompt:you are a student and your mentor wants to check your coding skills,give me two different implementations of Fibonacci sequence,one iterative and other recursive and also give a comparision table of those 2 ways making sure to include categories like time and space complexity,performance for large n,when recursion should be used

Category	Iterative Approach	Recursive Approach
Time Complexity	$O(n)$ — each term is computed once in a loop.	$O(2^n)$ — repeated recalculations make it exponential without memoization.
Space Complexity	$O(1)$ — only two variables are maintained.	$O(n)$ — due to recursive call stack depth.
Performance for Large n	Efficient and fast, suitable for very large values of n.	Very slow for large n unless optimized with memoization or dynamic programming.
Code Clarity	Straightforward, easy to follow, minimal overhead.	Elegant and mathematically intuitive, but harder to trace for beginners.
Debugging Ease	Easier to debug since logic is linear and explicit.	Harder to debug due to deep recursion and repeated calls.
When Recursion Should Be Used	Best avoided for large n; useful for demonstrating recursion concepts or small inputs.	Suitable for teaching recursion, mathematical elegance, or when memoization is added.