**AI ASSISTED CODING LAB**

**ASSIGNMENT 1.2**

**ENROLLMENT NO :**2503A51L25

**BATCH NO:** 19

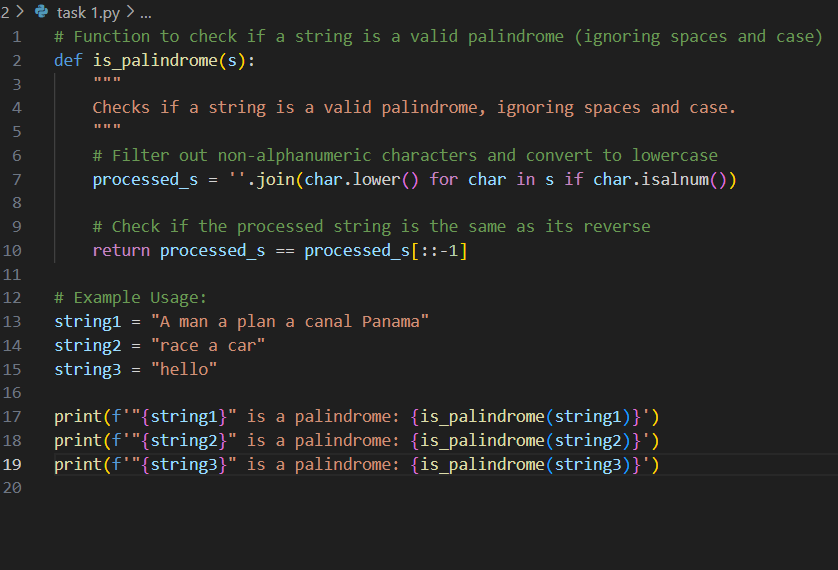
**NAME:** AJAY SRIRAMOJU

**TASK1**

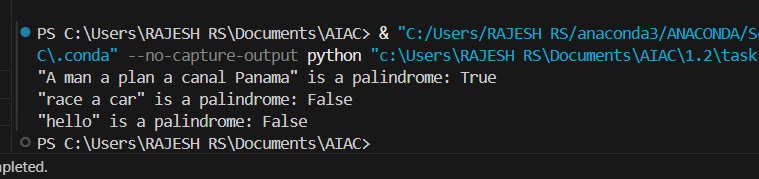
**TASK1 DESCRIPTION:-** Write a comment: # Function to check if a string is a valid palindrome (ignoring spaces and case) and allow Copilot to complete it

**PROMPT:-** Generate a comment: # Function to check if a string is a valid palindrome (ignoring spaces and case) and complete it.

**CODE:**



**OUTPUT:-**

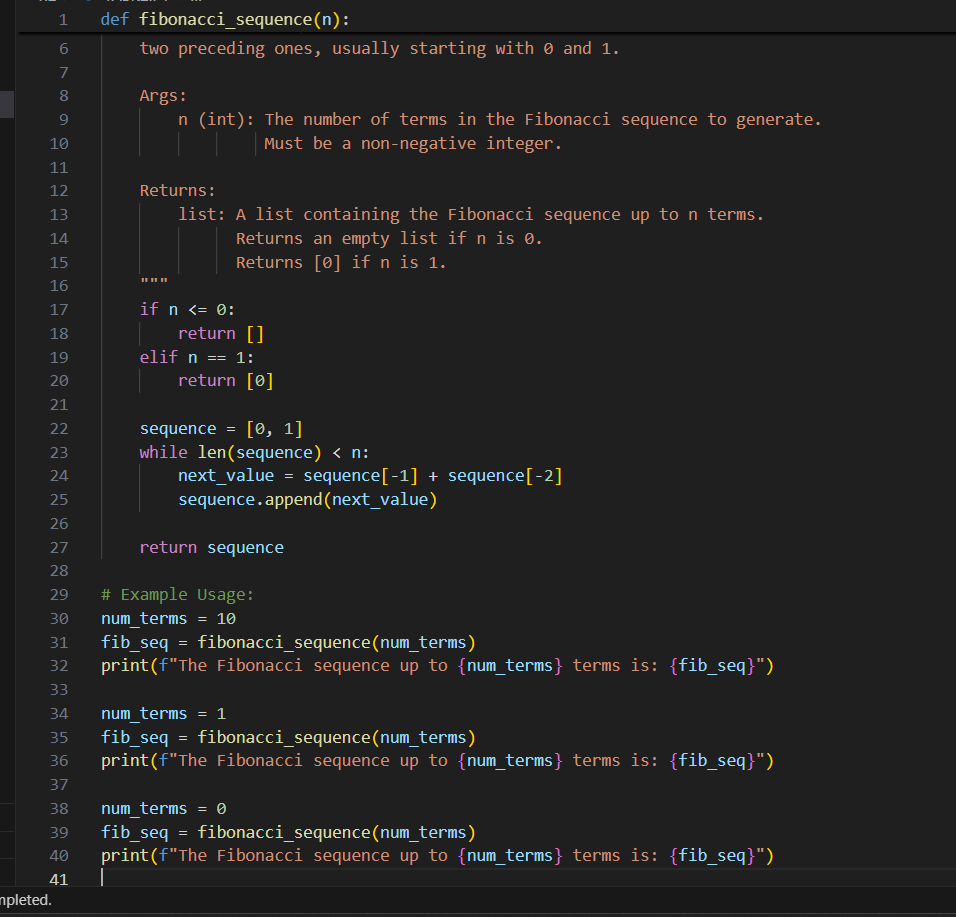


**TASK2**

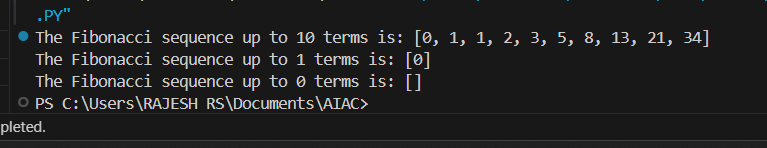
**TASK1 DESCRIPTION:-** Generate a Python function that returns the Fibonacci sequence up to n terms. Prompt with only a function header and docstring

**PROMPT:-**Generate a Python function that returns the Fibonacci sequence up to n terms. The function should include only the function header and a detailed docstring and should implement with example

**CODE:-**



**OUTPUT:-**

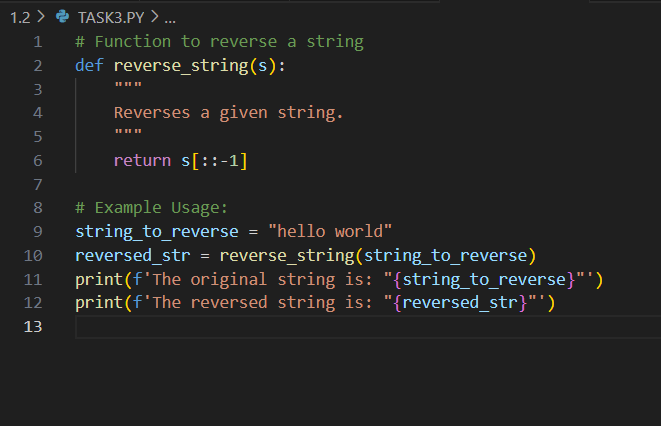


**TASK3**

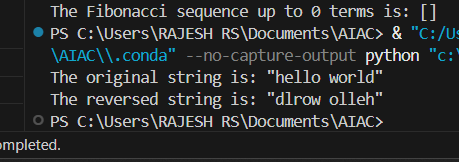
**TASK3 DESCRIPTION:-** Write a comment like # Function to reverse a string and use Copilot to generate the function.

**PROMPT:-**Write a comment like # Function to reverse a string and generate the  
function

**CODE:-**



**OUTPUT:-**

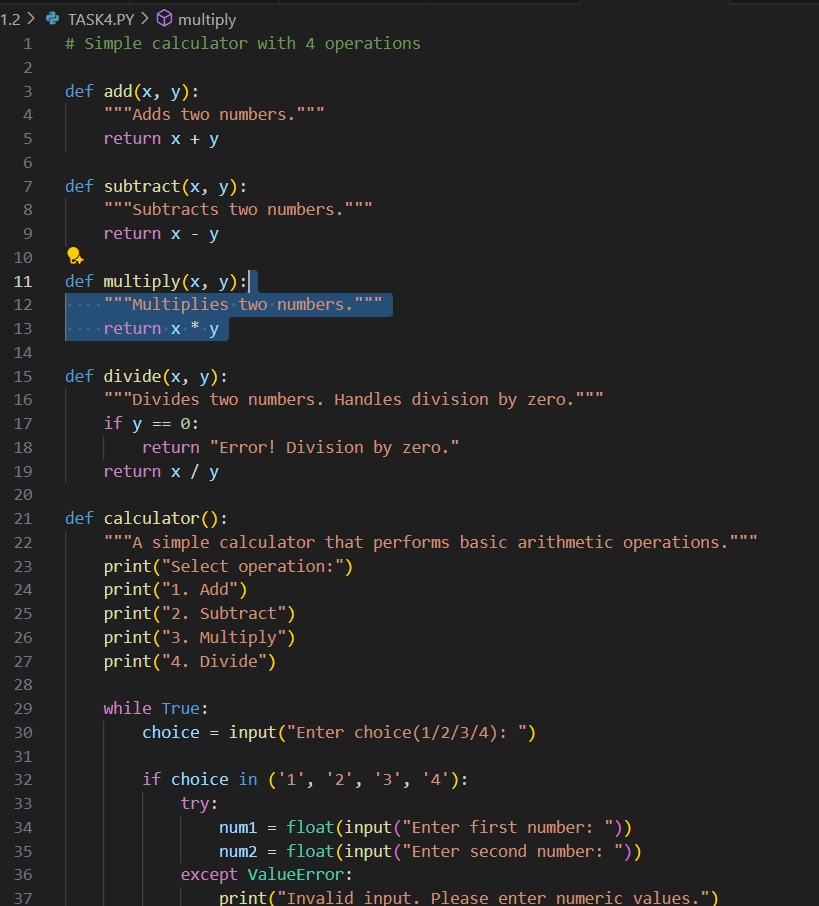


**TASK4**

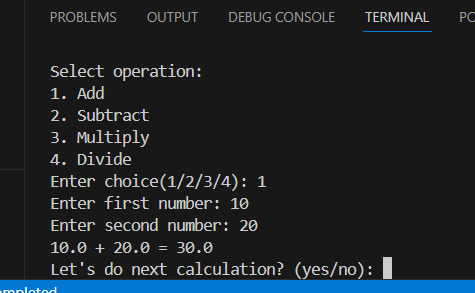
**TASK4 DESCRIPTION:-** Generate a program that simulates a basic calculator (add, subtract, multiply, divide). Write the comment: # Simple calculator with 4 operations and let AI complete it.

**PROMPT:-**Write the comment: "# Simple calculator with 4 operations" The program should be a fully working calculator with input/output and operator selection logic, supporting add, subtract, multiply, and divide operations.

**CODE:-**



**OUTPUT:-**

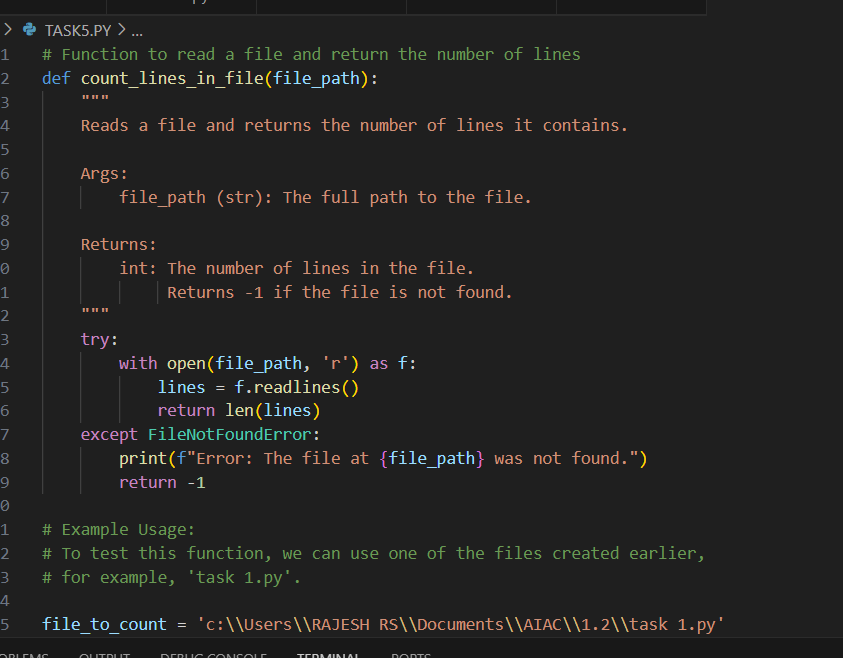


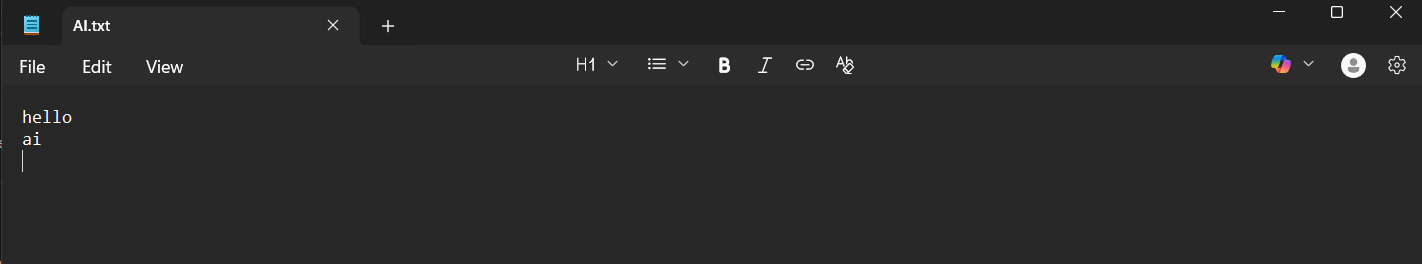
**TASK5**

**TASK5 DESCRIPTION:-** Use a comment to instruct AI to write a function that reads a file and returns the number of lines.

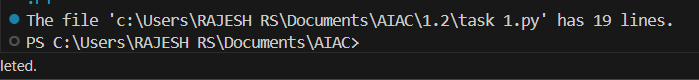
**PROMPT:-** write a function that reads a file and returns the number of lines.

**CODE:-**



****

**OUTPUT:-**



**OBSERVATION:-** I explored how **GitHub Copilot** can generate complete and functional Python programs from minimal prompts such as comments, function headers, or docstrings. Each task demonstrated the usefulness of AI-assisted coding in reducing effort, saving time, and improving productivity.

* In **Task 1**, I observed that a simple comment describing the functionality (checking if a string is a palindrome) was enough for Copilot to generate a correct implementation with case and space handling.
* In **Task 2**, providing only a function header and docstring for generating the Fibonacci sequence highlighted how Copilot understands documentation and converts it into working logic.
* In **Task 3**, a short comment about reversing a string allowed Copilot to create a concise and accurate function, showing its ability to handle basic operations efficiently.
* In **Task 4**, using a comment to describe a “simple calculator with four operations” led Copilot to produce a fully functional program with input handling and operator selection, proving its capability to generate interactive applications.
* In **Task 5**, Copilot was able to generate file-handling logic from a single comment, showing its usefulness in automating repetitive tasks like counting lines in a file.