**LAB NAME : AI ASSISTED CODING**

**ROLL NO :2503A51L30**

**BRANCH : CSE**

**NAME : B.MAHESH**

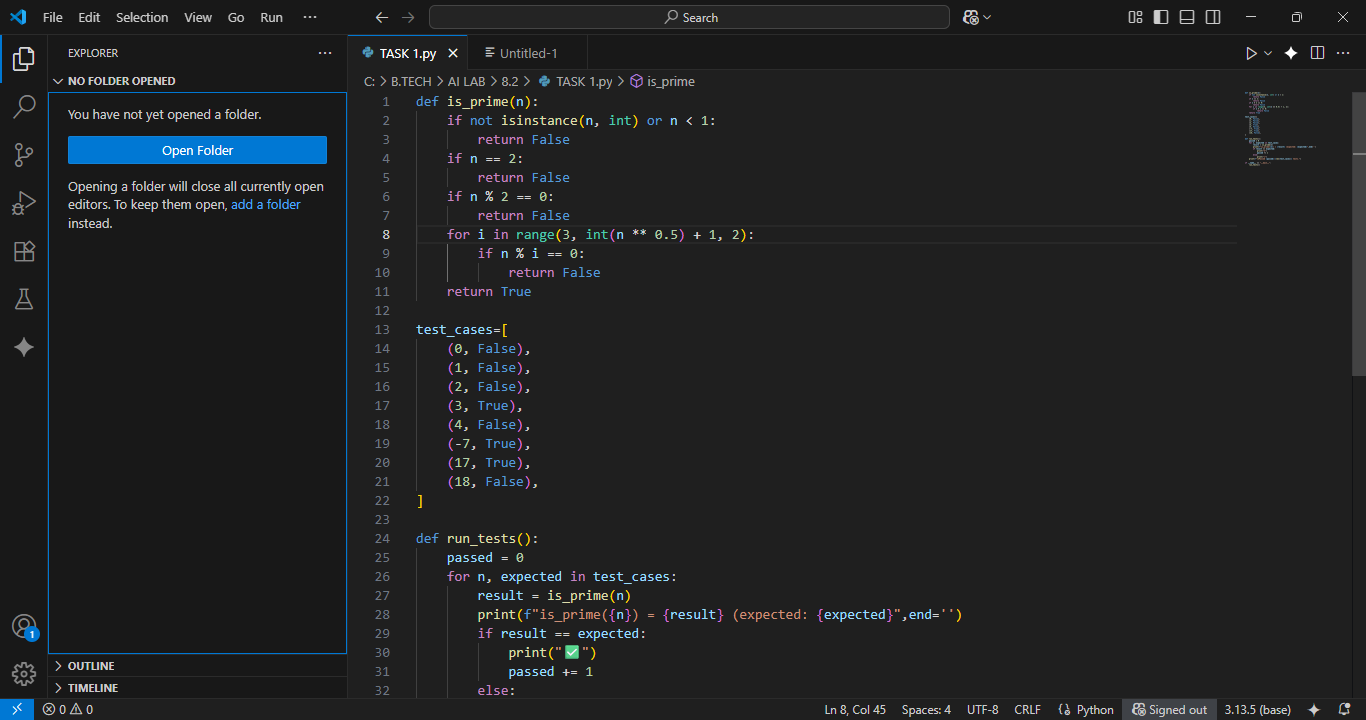
**TASK 1**

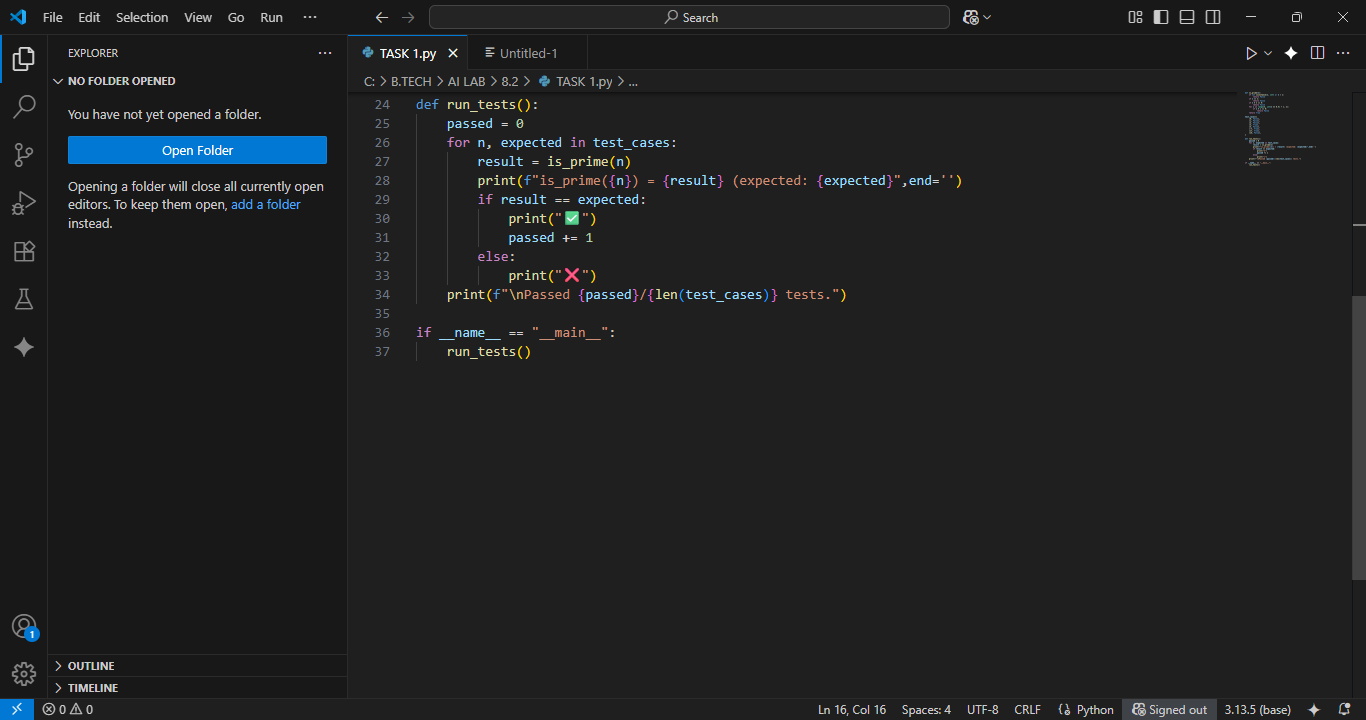
**Task Description:** Use AI to generate test cases for a function is\_prime(n) and then implement the function.  
Requirements:  
• Only integers > 1 can be prime.  
• Check edge cases: 0, 1, 2, negative numbers, and large primes**.**

**PROMPT:**

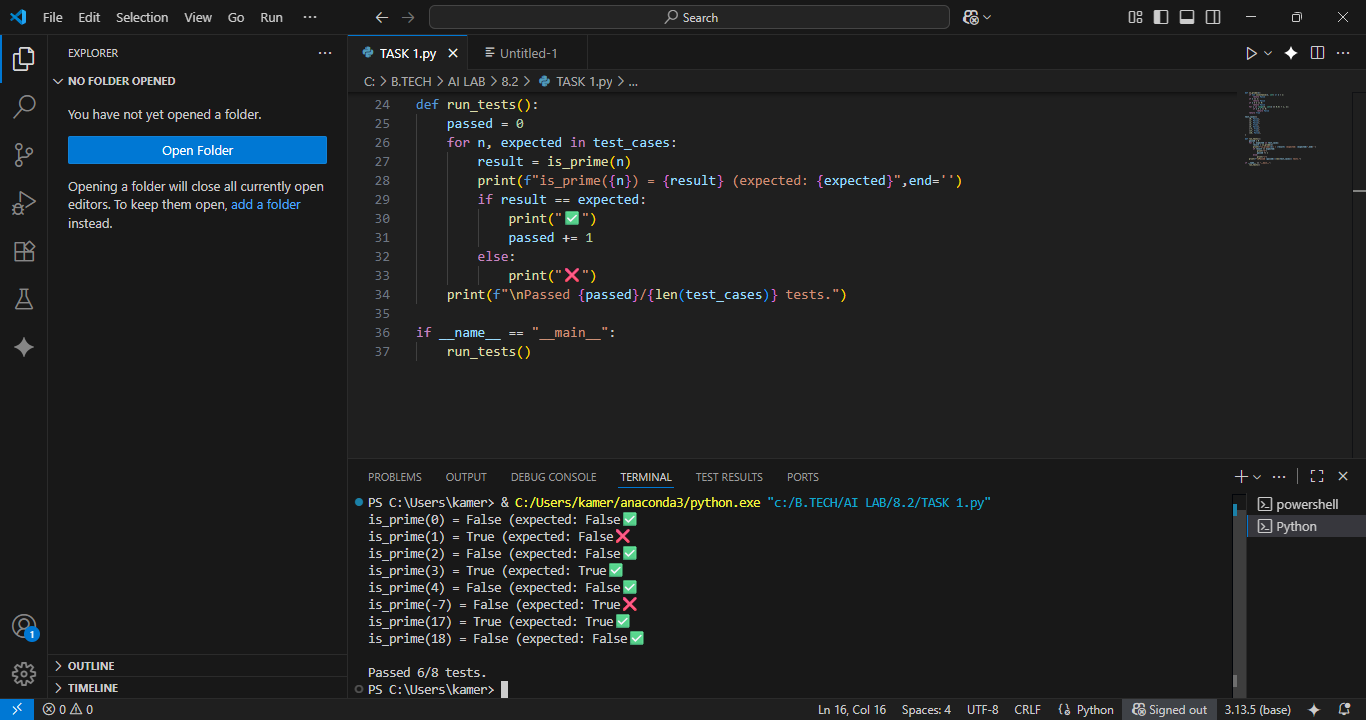
Please write a Python function is\_prime(n) and a corresponding suite of test cases to validate it. The function should be implemented first, followed by the tests.

**CODE:**





**OUTPUT:**



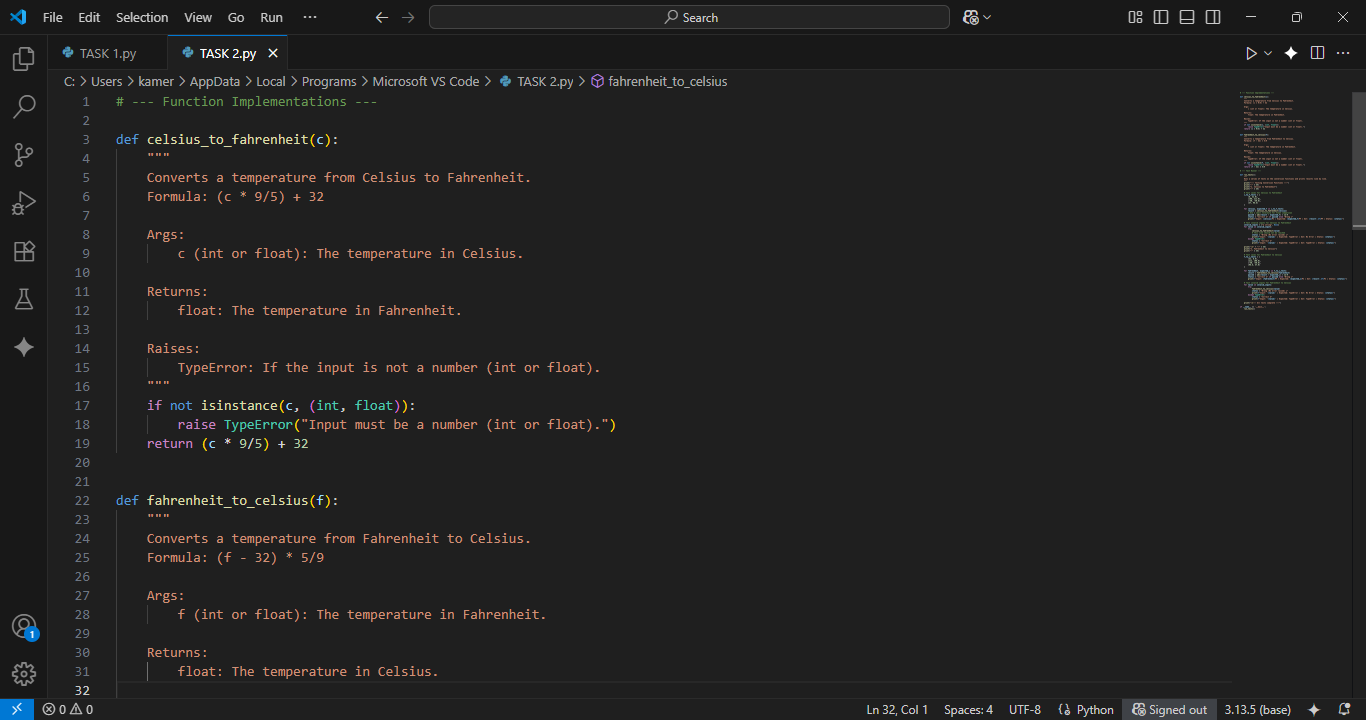
**TASK 2**

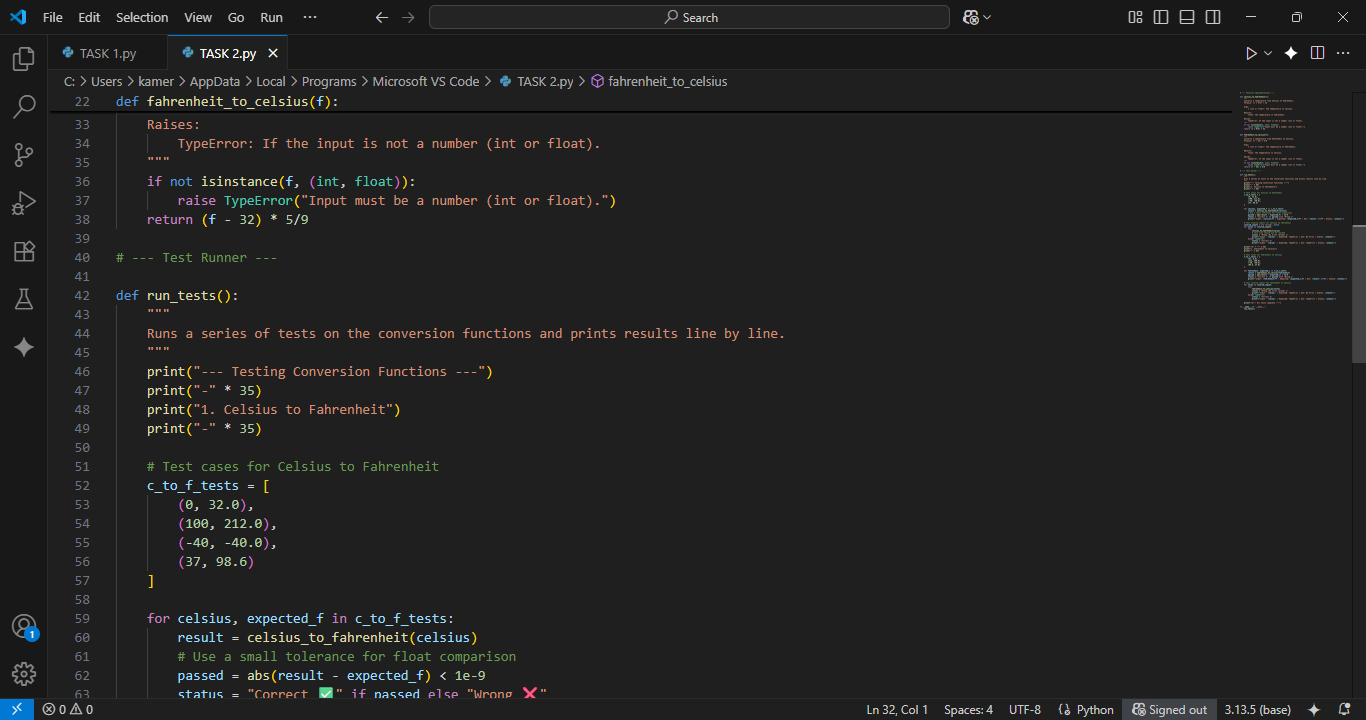
**Task Description**: Ask AI to generate test cases forcelsius\_to\_fahrenheit(c) and fahrenheit\_to\_celsius(f). **Requirements:**• Validate known pairs: 0°C = 32°F, 100°C = 212°F.  
• Include decimals and invalid inputs like strings or None.

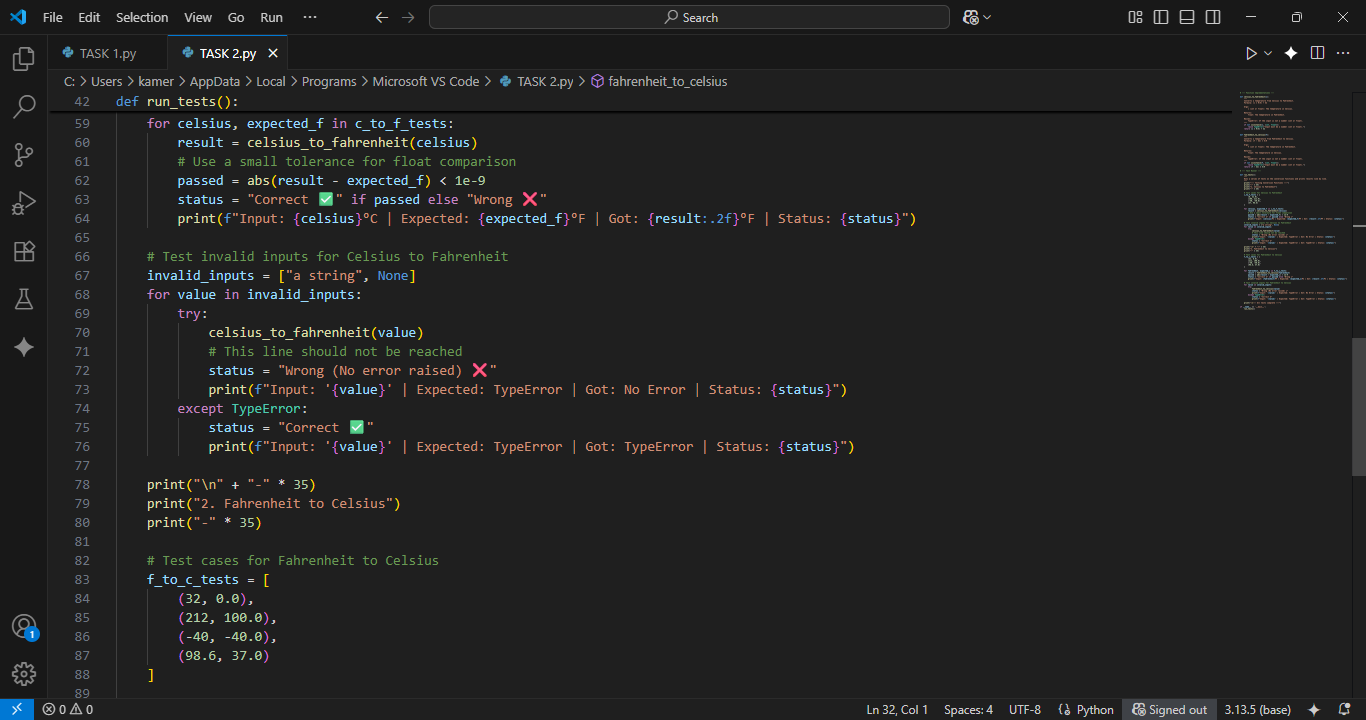
**PROMPT:**

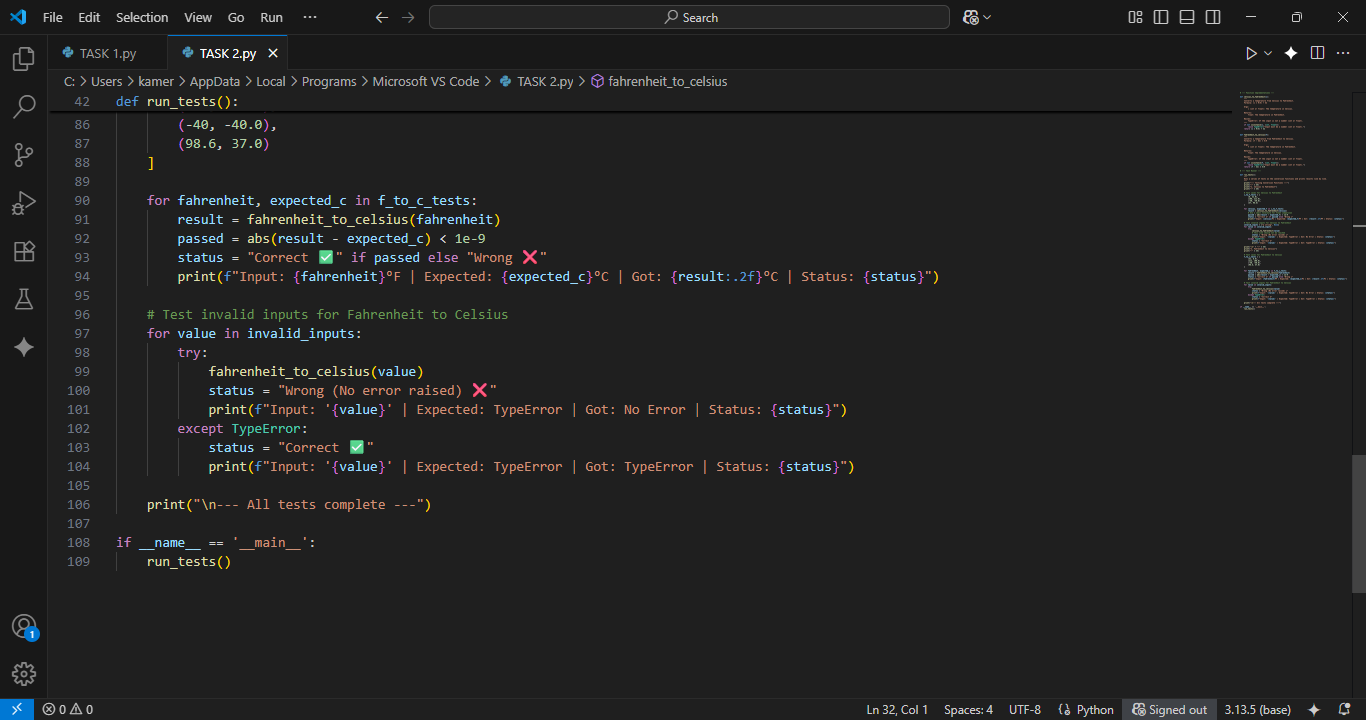
Please write two Python functions for temperature conversion, celsius\_to\_fahrenheit(c) and fahrenheit\_to\_celsius(f), and a corresponding suite of test cases to validate them.

**CODE:**

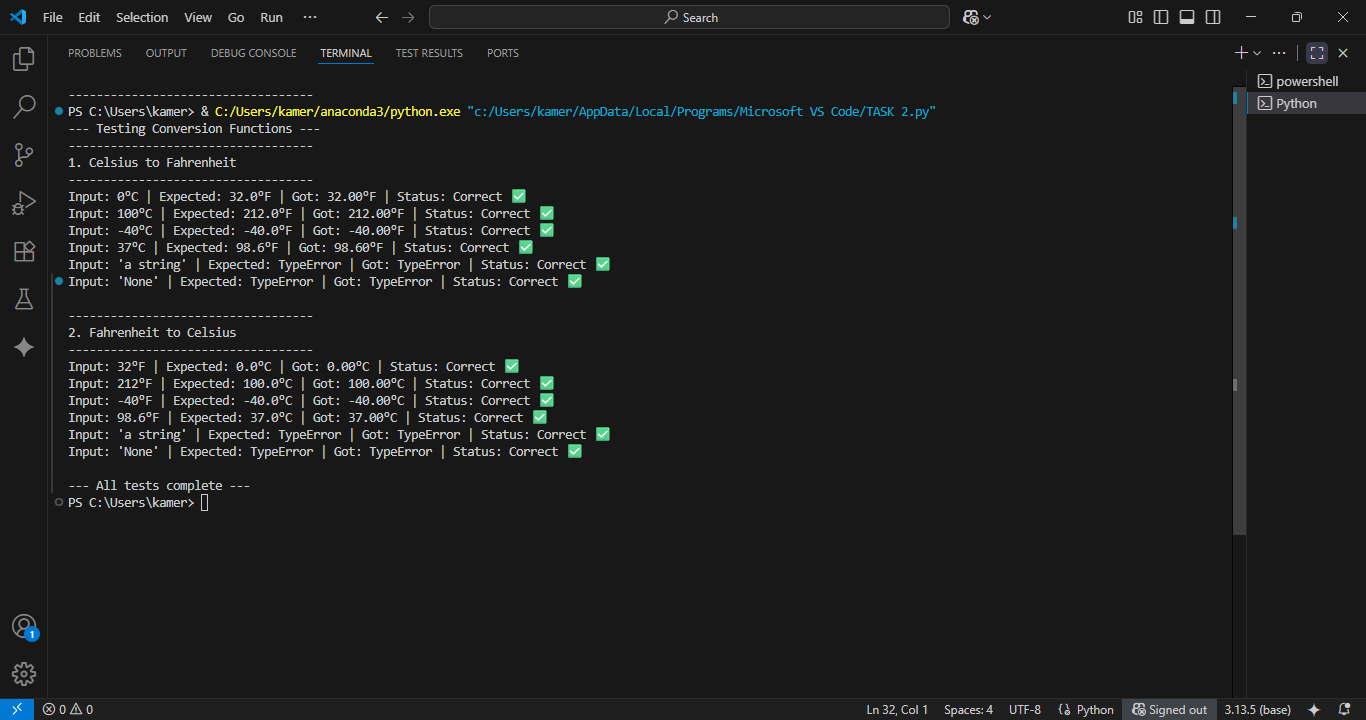
****

****

****

****

**OUTPUT:**



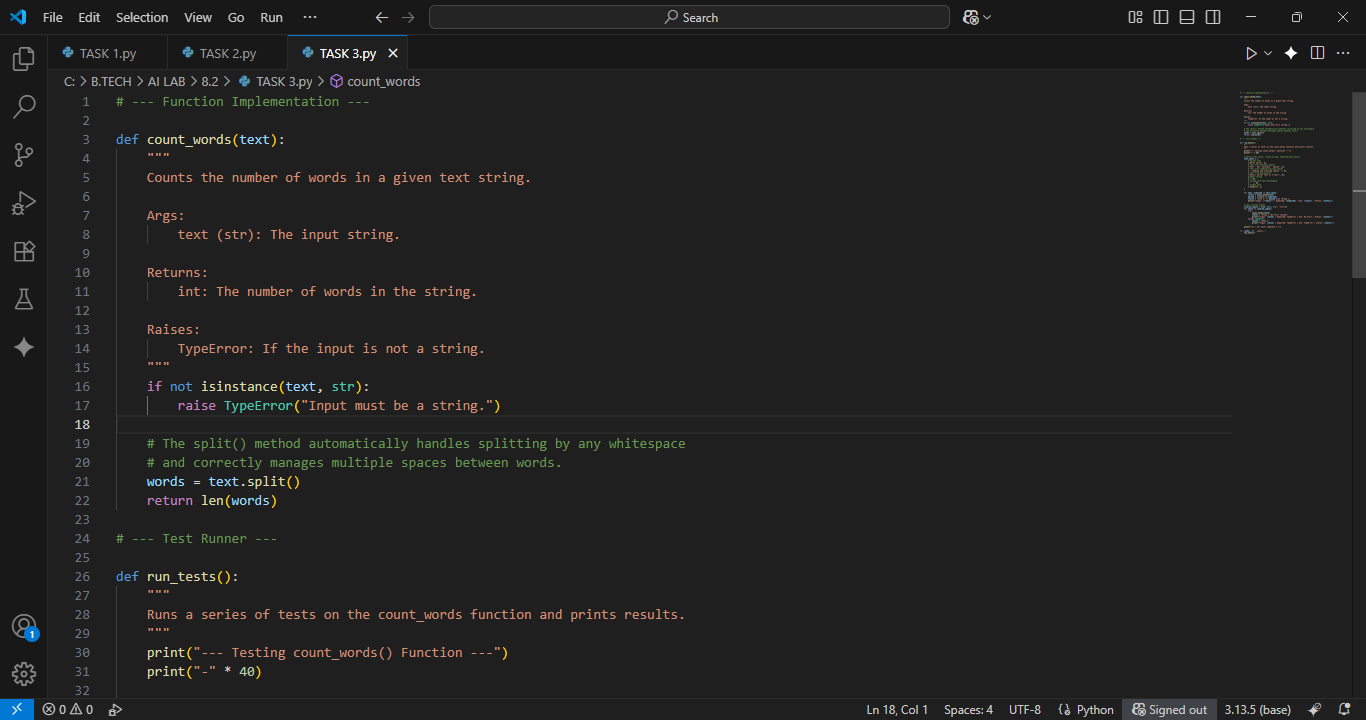
**TASK 3**

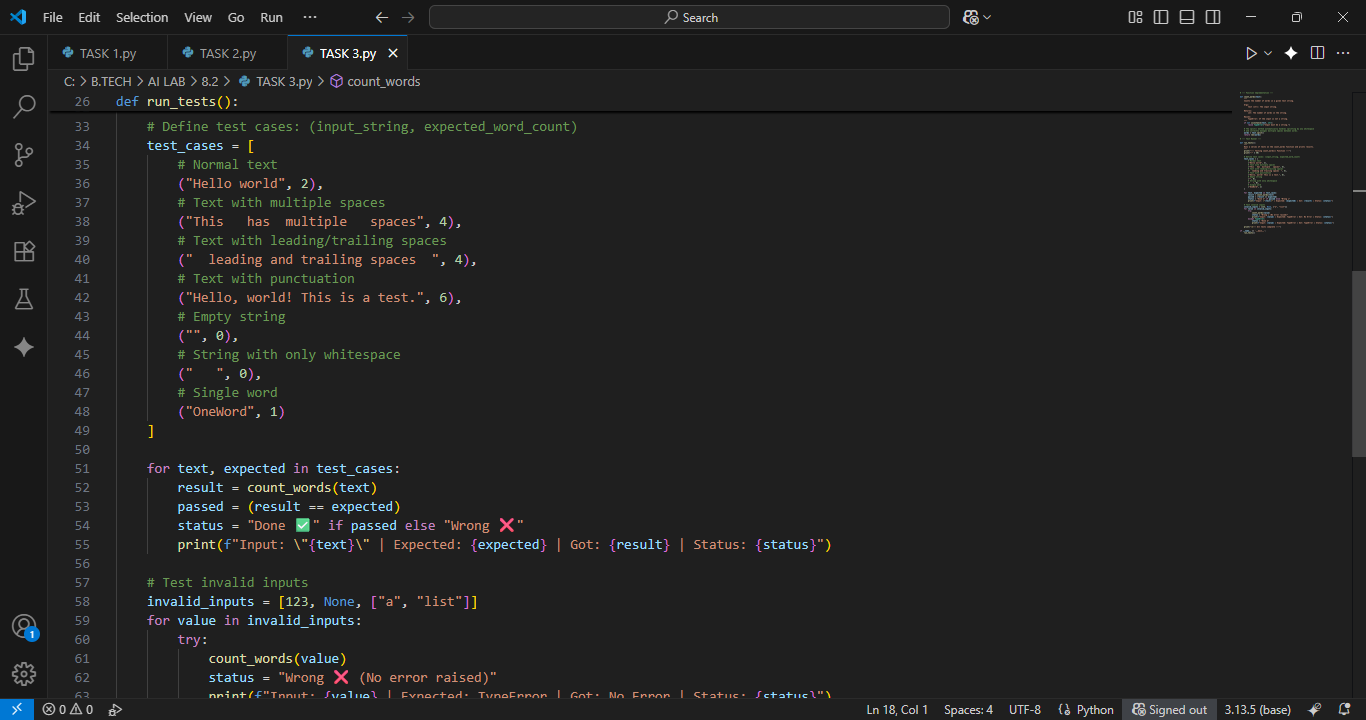
**Task Description:**Use AI to write test cases for a function count\_words(text) that returns the number of  
words in a sentence**.  
Requirements**Handle normal text, multiple spaces, punctuation, and empty strings.

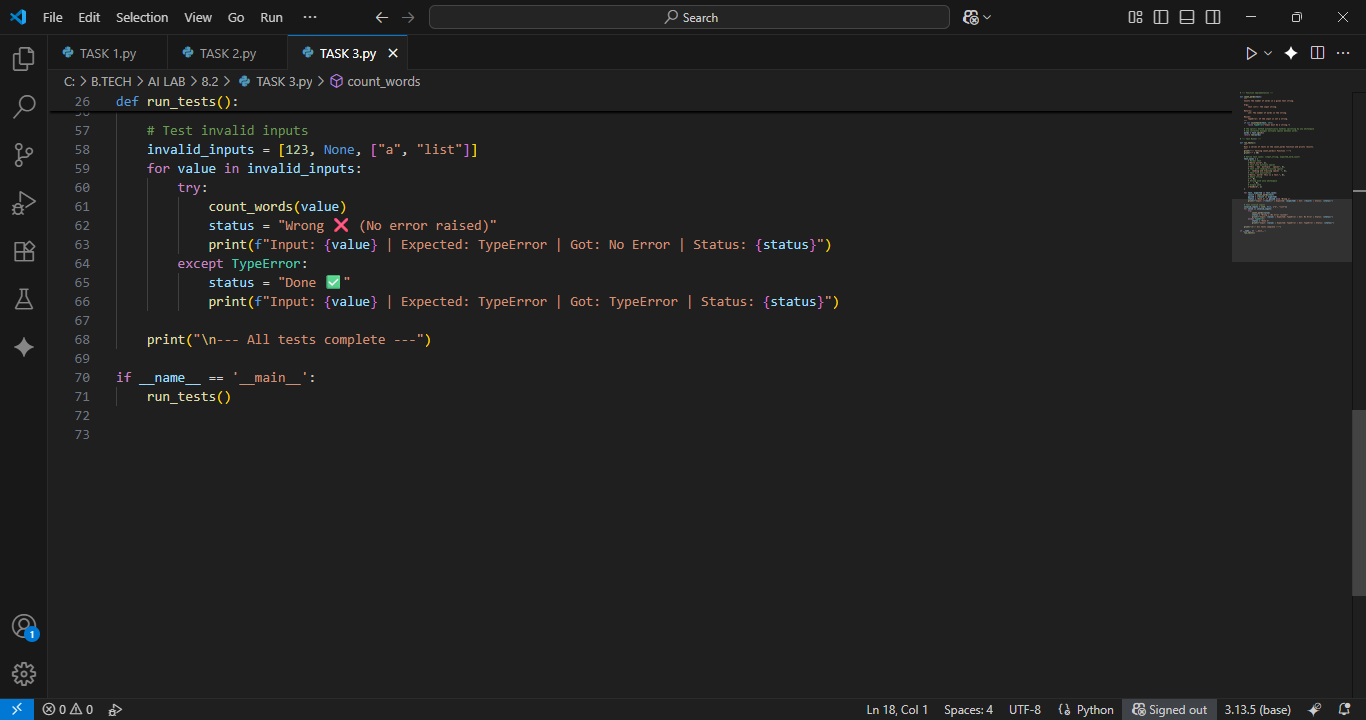
**PROMPT:**

Please write a Python function count\_words(text) and a corresponding suite of test cases to validate it.

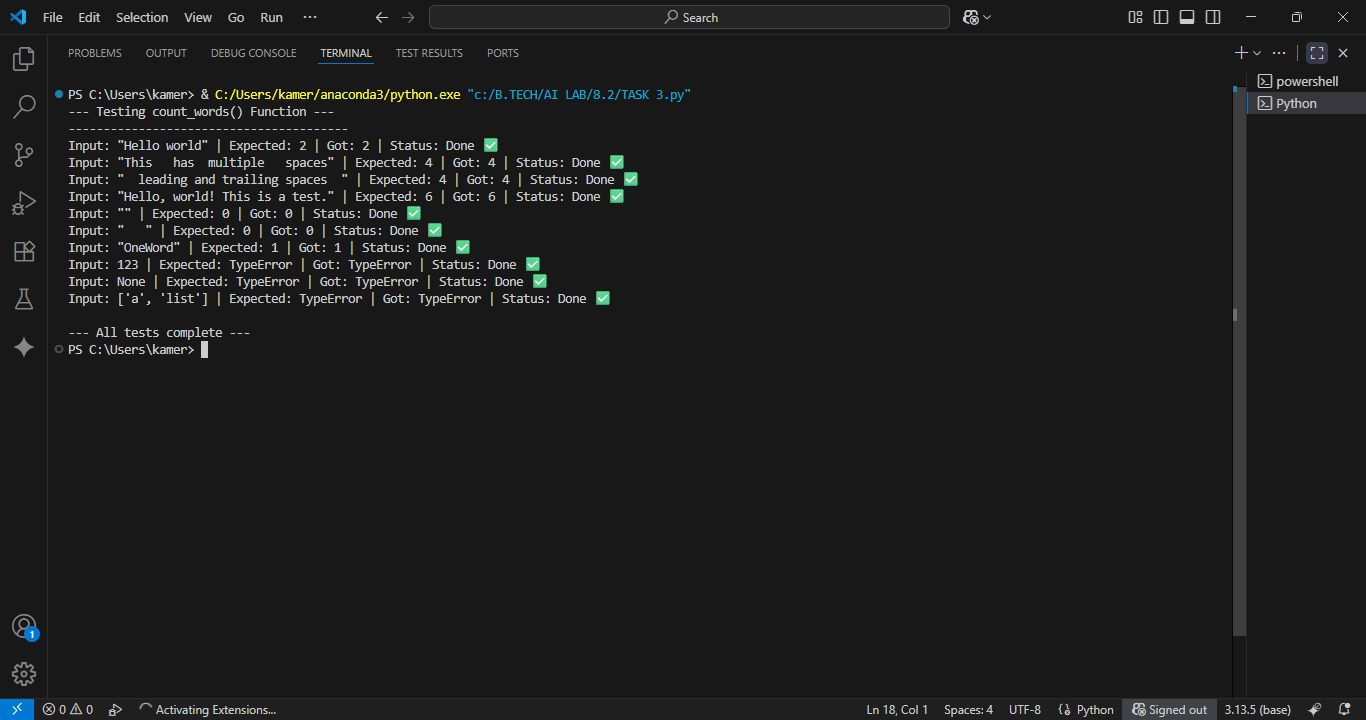
**CODE:**

****





**OUTPUT:**



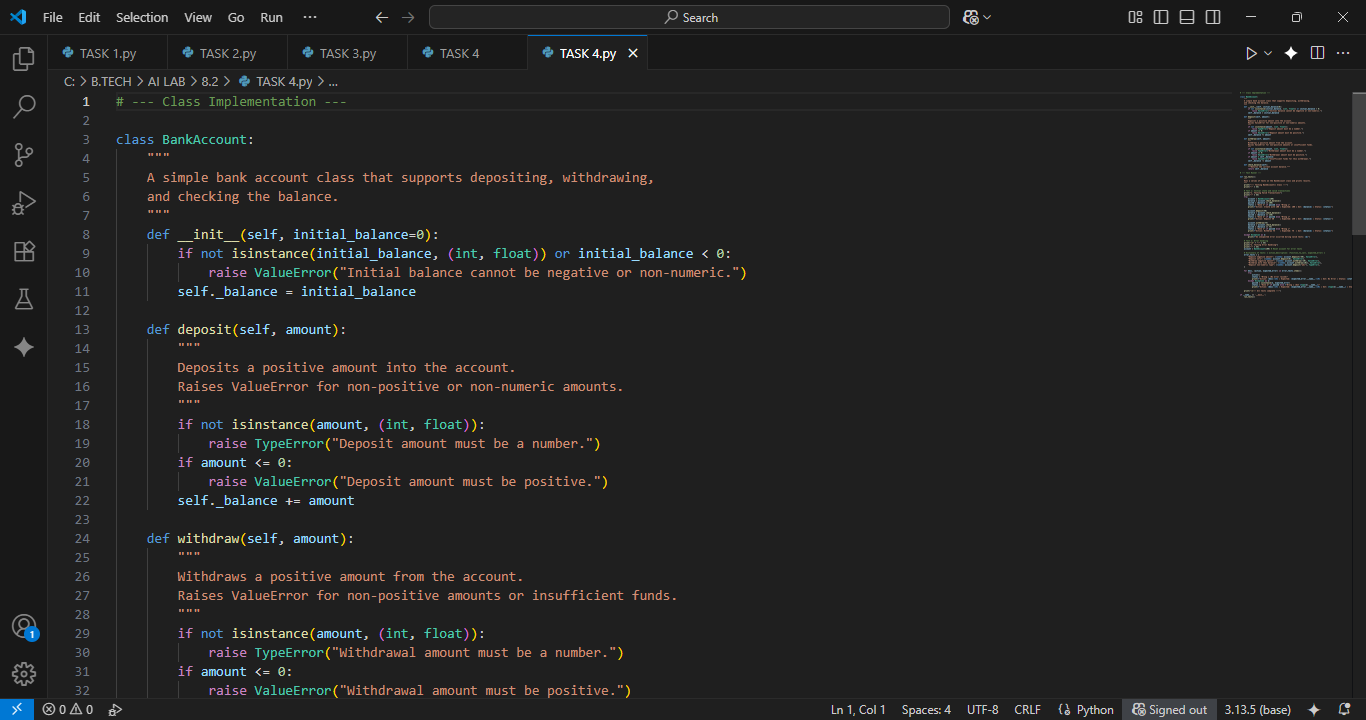
**TASK 4**

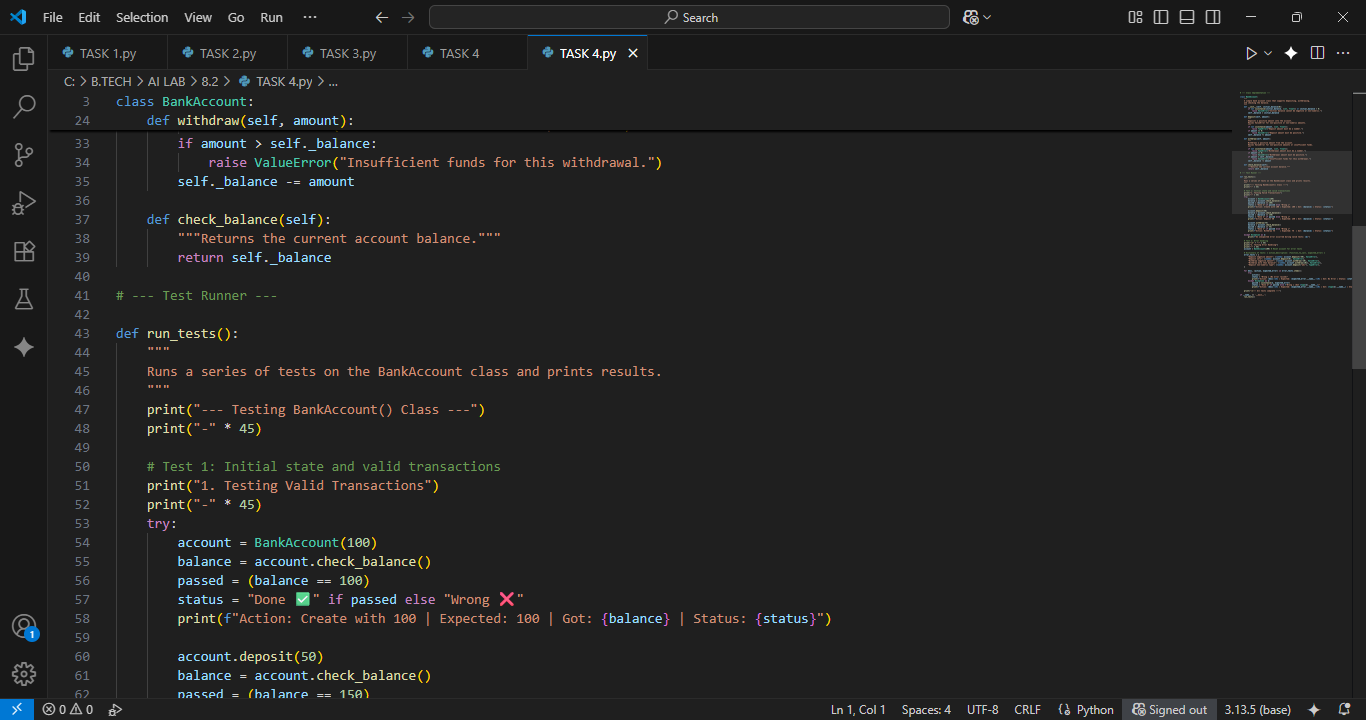
**Task Description:**• Generate test cases for a BankAccount class with:  
Methods:  
deposit(amount)  
withdraw(amount)  
check\_balance() **Requirements:**• Negative deposits/withdrawals should raise an error.  
• Cannot withdraw more than balance.

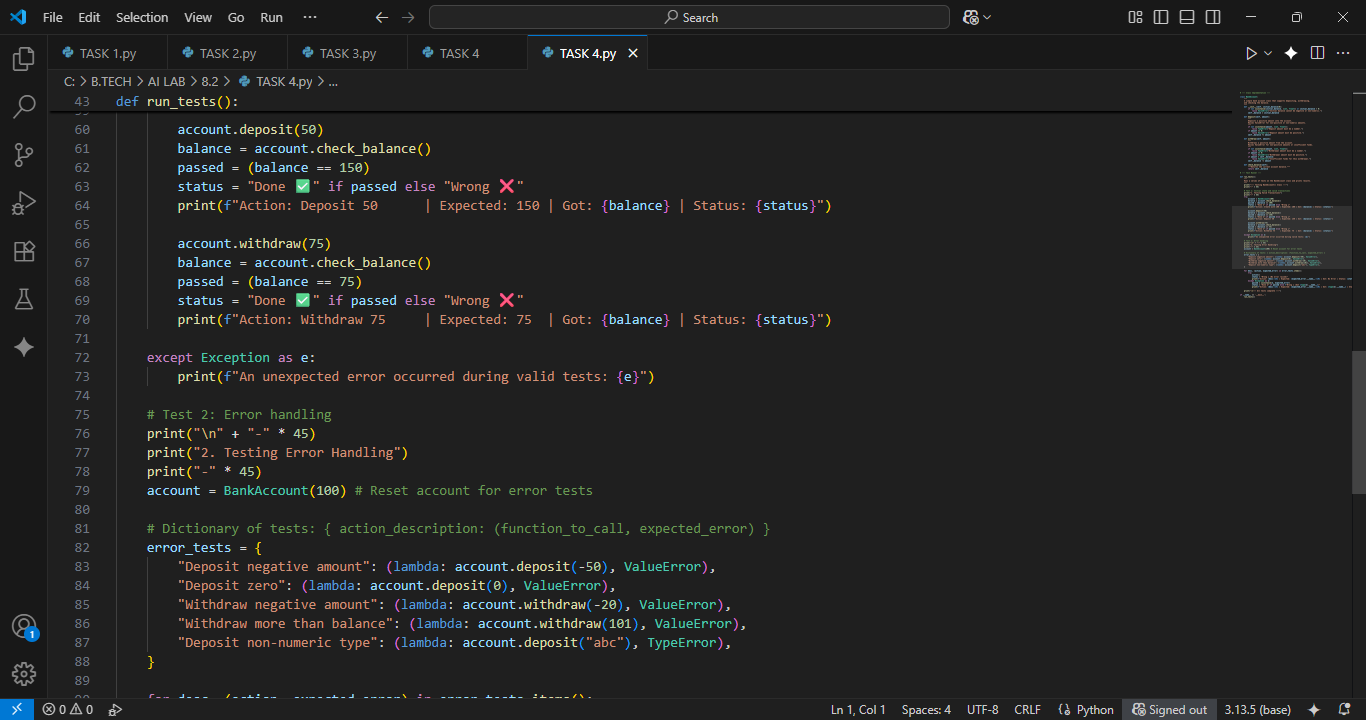
**PROMPT:**

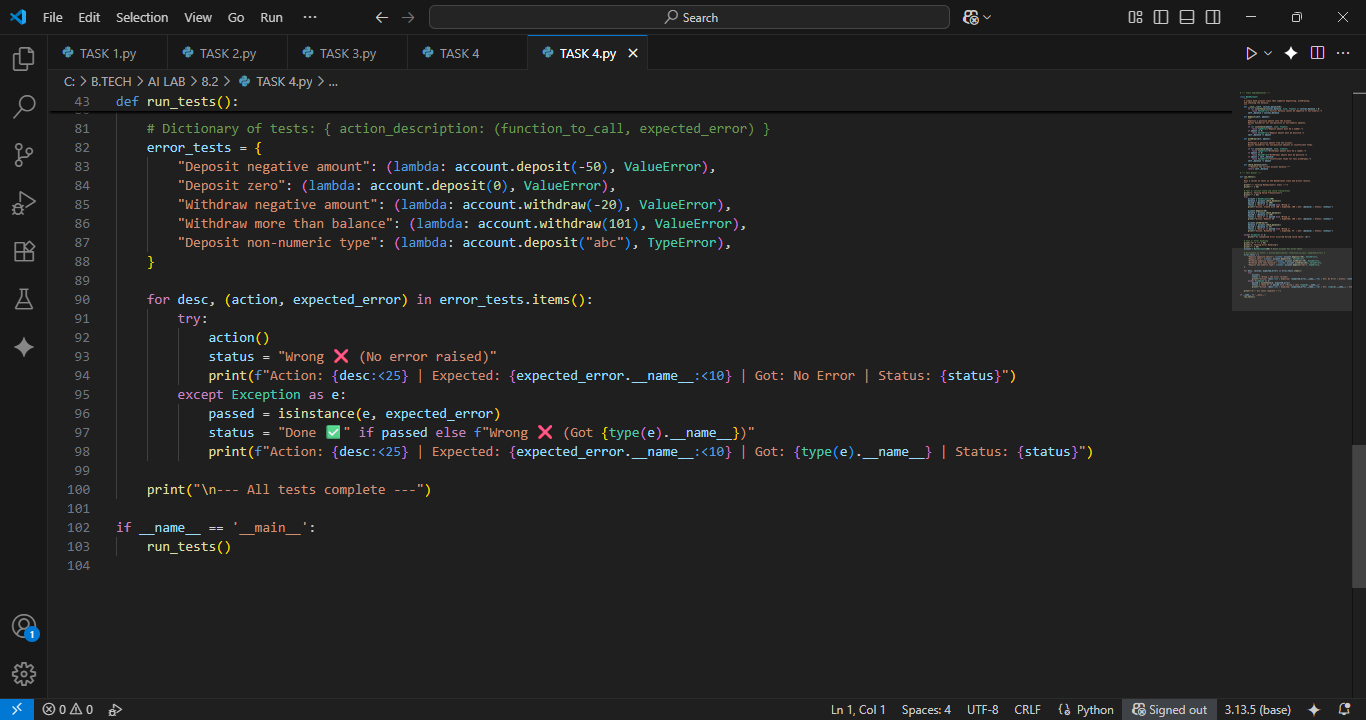
Please write a Python BankAccount class and a corresponding suite of test cases to validate its functionality**.**

**CODE:**

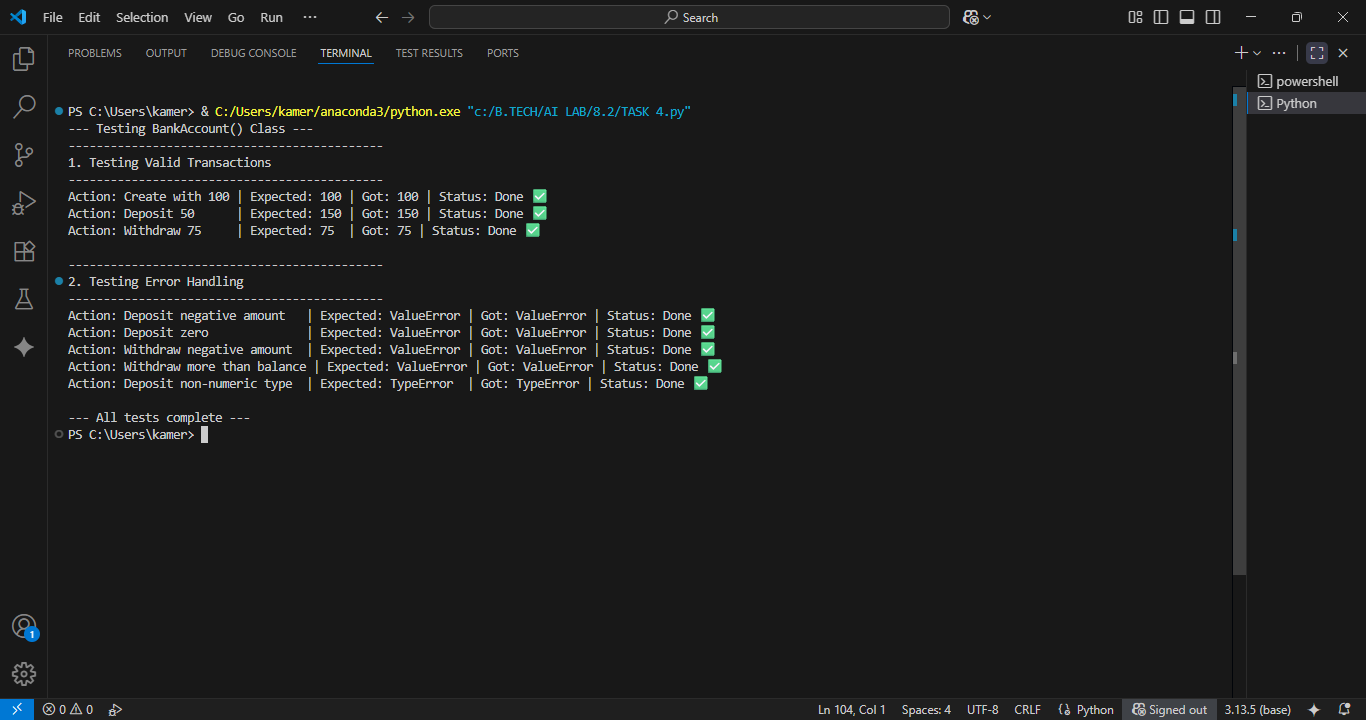








**OUTPUT:**



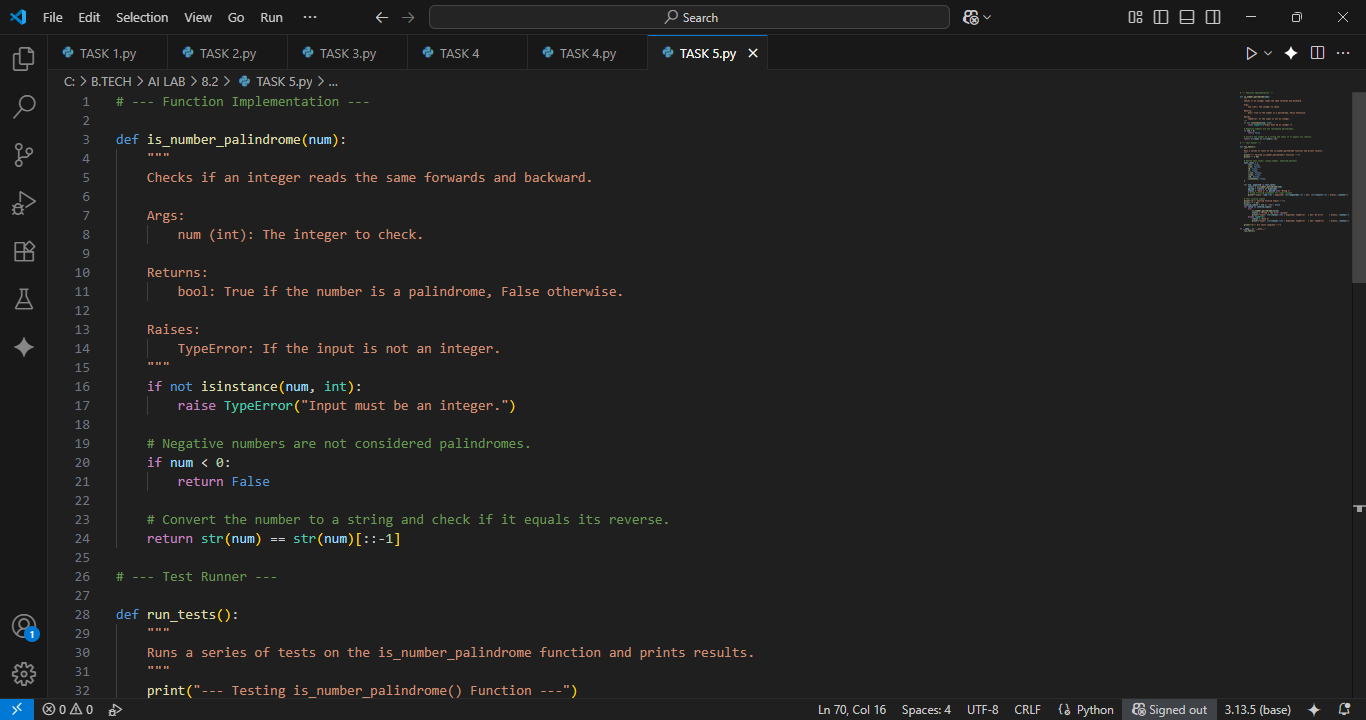
**TASK 5**

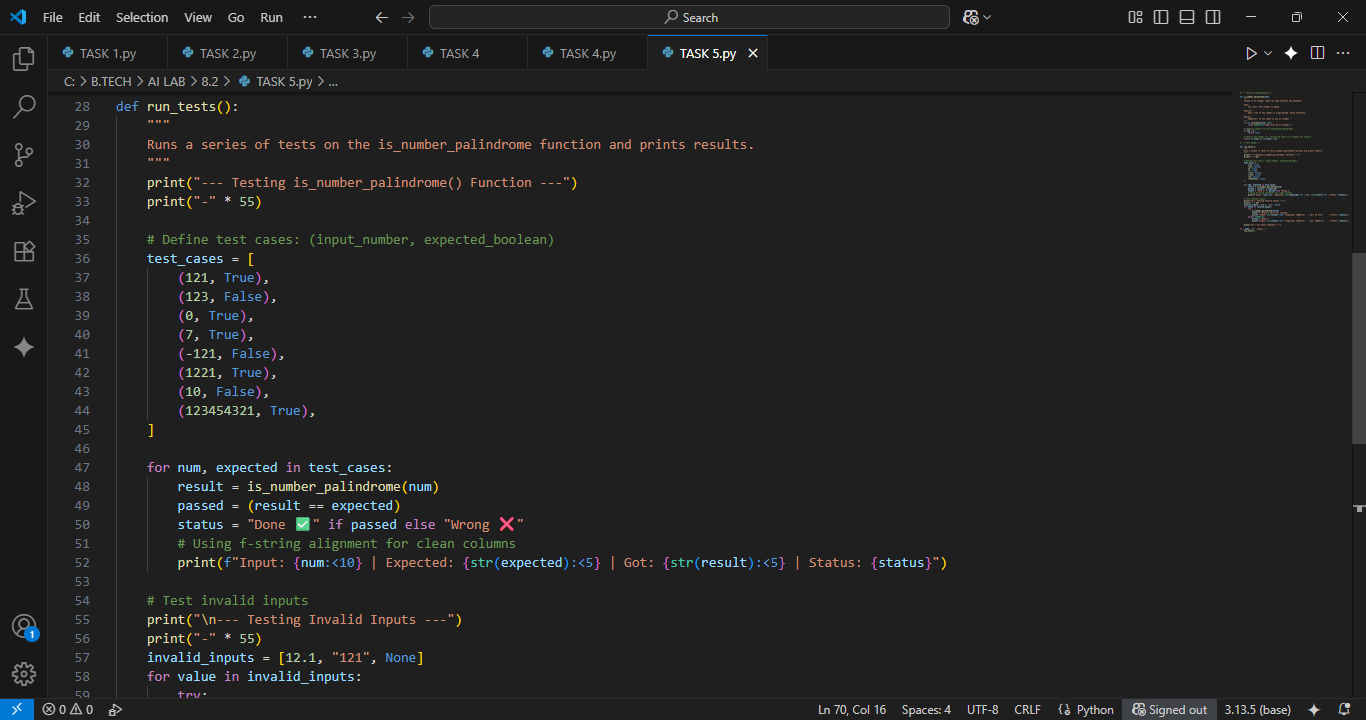
**Task Description:**Generate test cases for is\_number\_palindrome(num), which checks if an integer reads  
the same backward. **Examples:**121 → True  
123 → False  
0, negative numbers → handled gracefully.

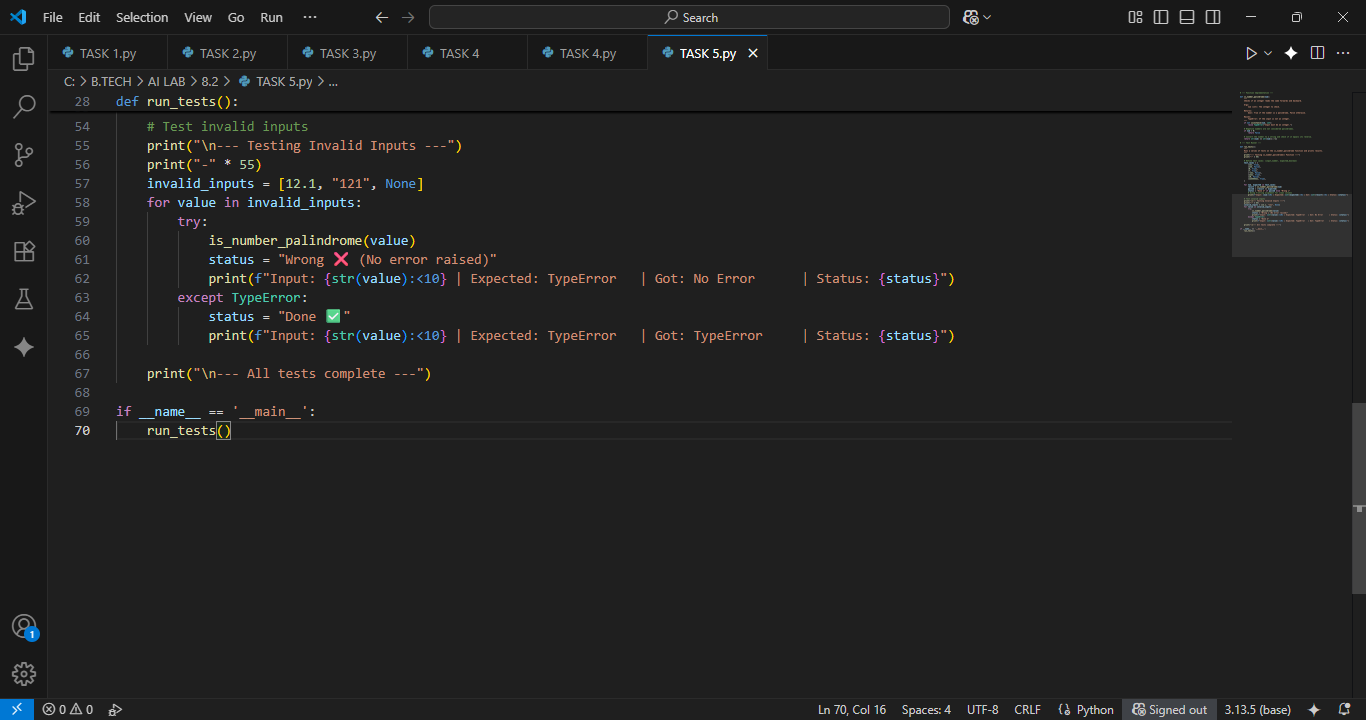
**PROMPT:**

Please write a Python function is\_number\_palindrome(num) and a corresponding suite of test cases to validate it**.**

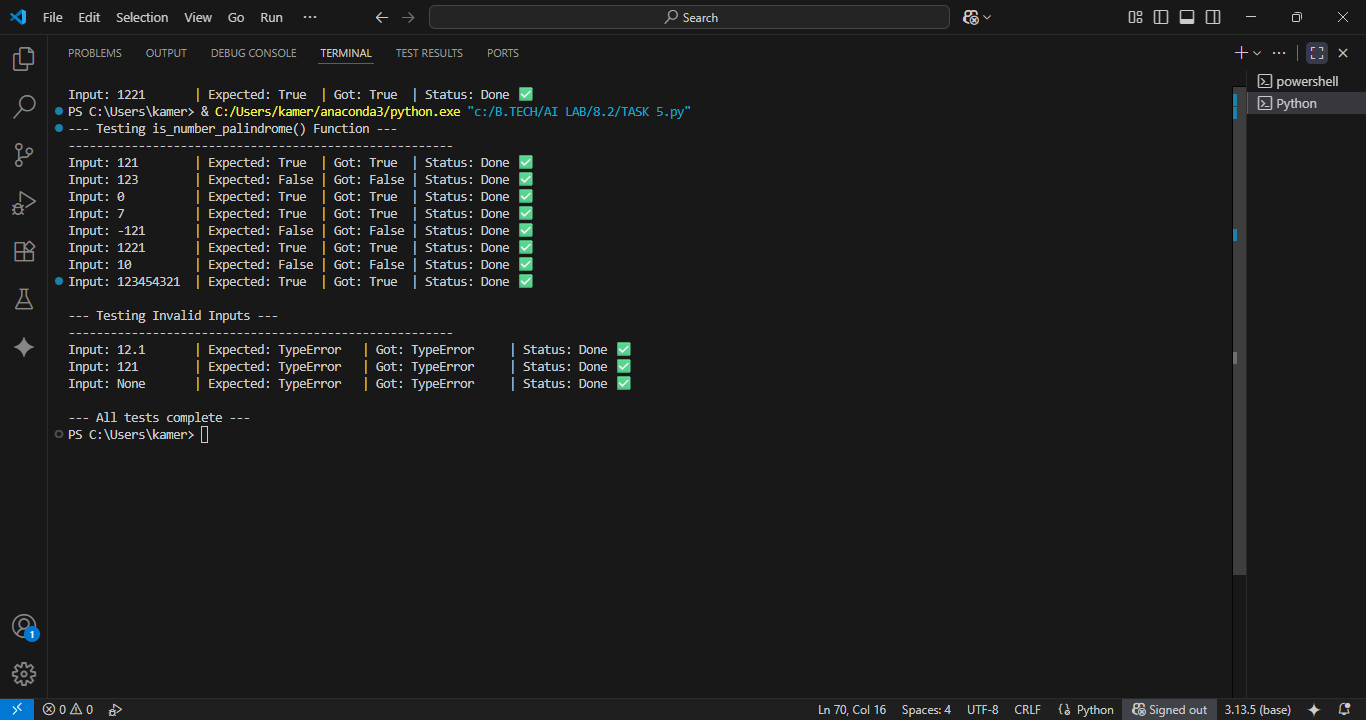
**CODE:**







**OUTPUT:**



**OBSERVATION:**

I Observed that Gemini AI can easily generate the programs correctly from the given prompt and Gemini AI Provides a Detailed Explanation and Debugging. Gemini AI is a fascinating tool to observe—especially in how it transforms the developer experience.