ASSIGNMENT-8

NAME: G. OMKAR

HT.NO: 2403A52039

BATCH: 03

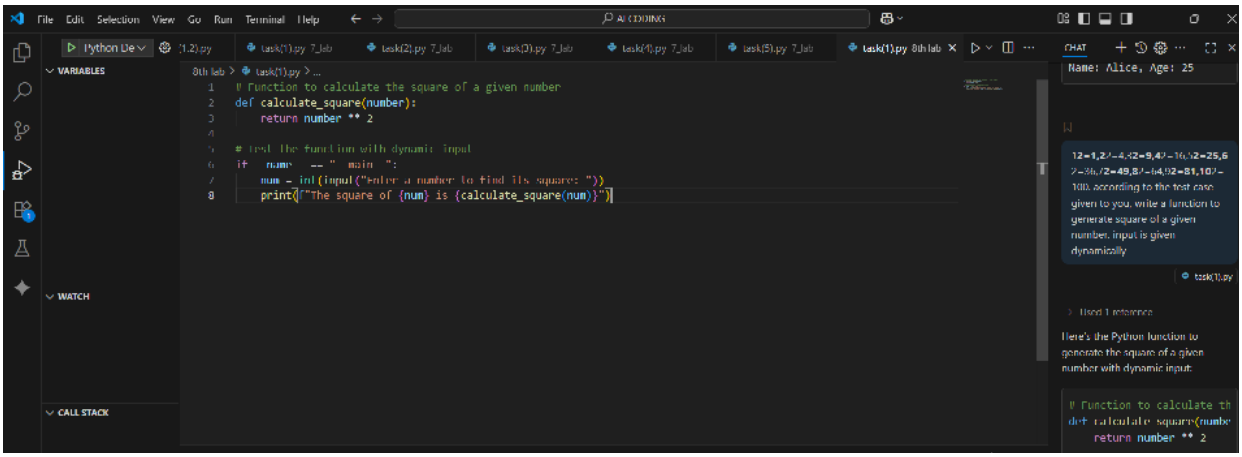
Task-1

Write a test case to check if a function returns the square of a number.

Then write the function with help from GitHub Copilot or Cursor AI.

Prompt:
1**2=1,2**2=4,3**2=9,4**2=16,5**2=25,6**2=36,7**2=49,8**2=64, 9**2=81,10**2=100. according to the test case given to you, write a function to generate square of a given number. input is given dynamically

Code:



OP:

Observation:

The calculate_square function takes a number as input and returns its square using the ** operator. I have prompted to input a number dynamically. The program calculates the square of the input number and prints the result.

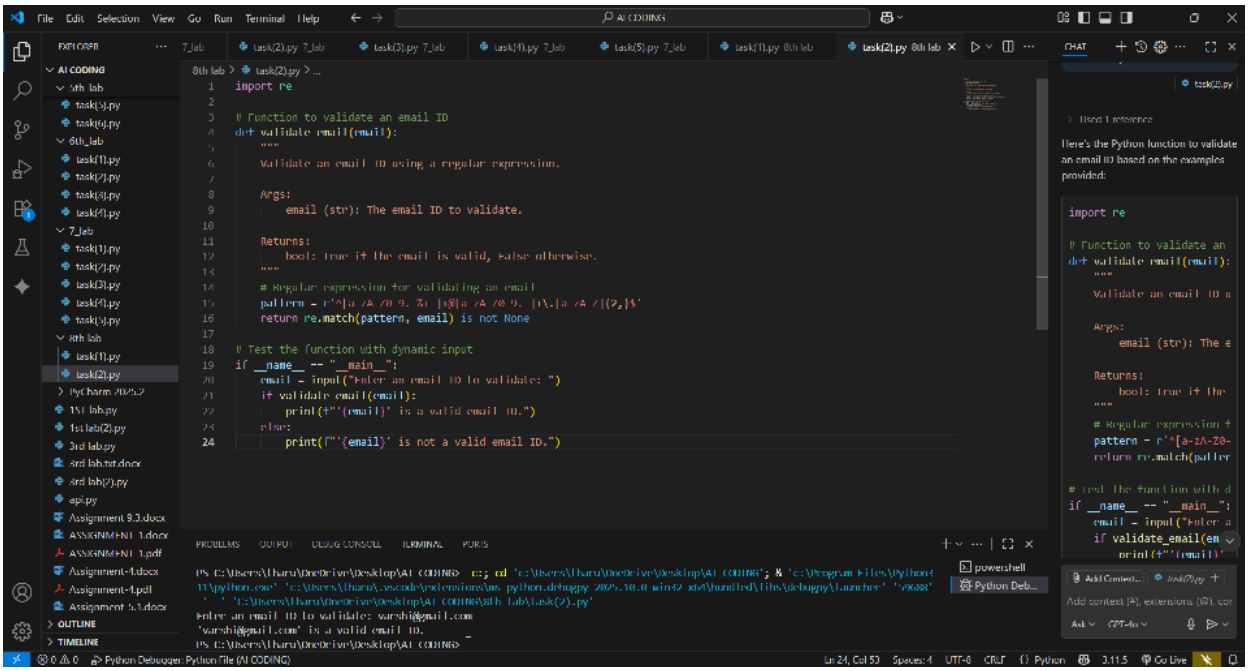
Task-2:

Create test cases to validate an email address (e.g., contains @ and .com).Use AI assistance to implement the validate_email() function

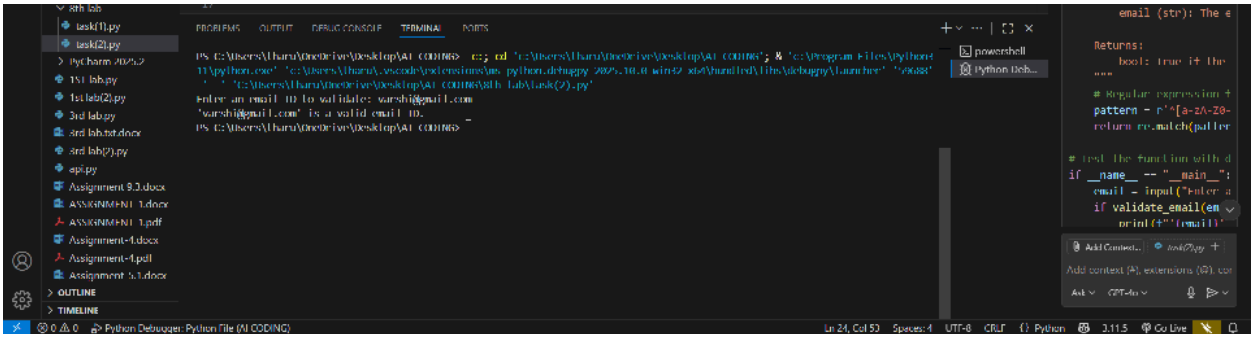
Prompt:

write a function which validates a email id. examples of a mail id is containt@gmail.com, madam@gmail.com, anyone@gmail.com. according to the test case/examples given generate code.

Code:



OP:



Observation:

Examples like containt@gmail.com, madam@gmail.com, and anyone@gmail.com are valid because, They contain alphanumeric characters before the @. They have a domain name after the @ (e.g., gmail.com). The domain name ends with a valid top-level domain (e.g., .com). Emails without an @ symbol or domain name are invalid. Emails with special characters in invalid positions (e.g., @example.com or name@.com) are invalid.

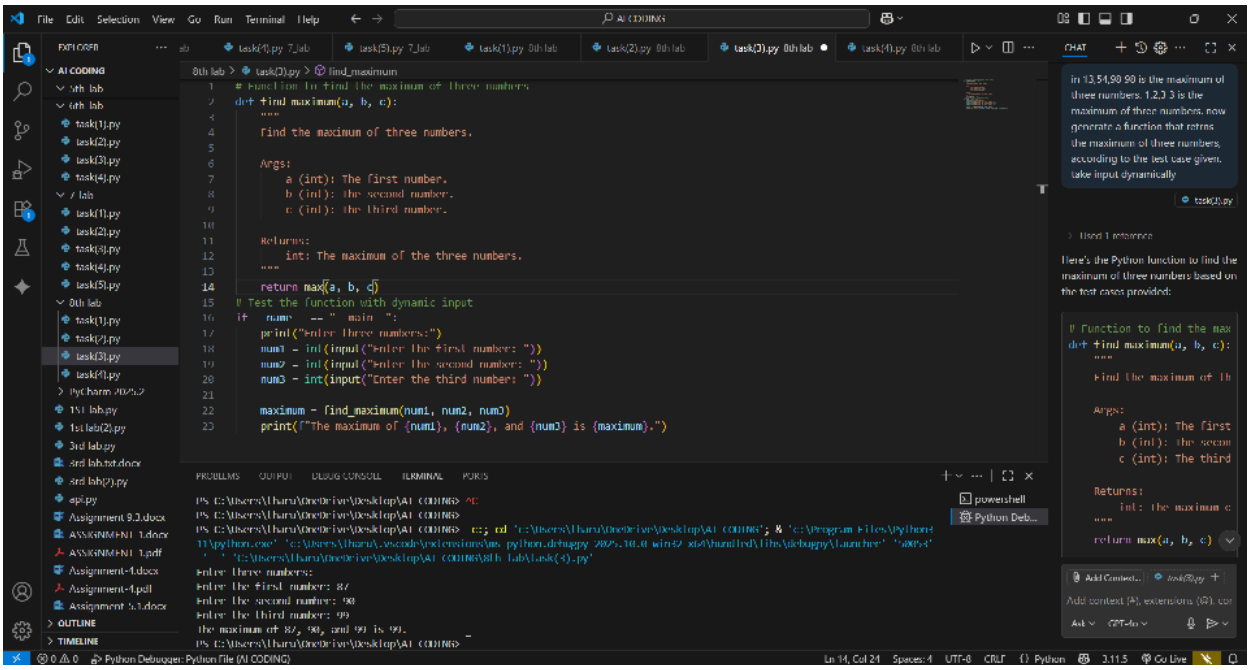
Task-3:

Write test cases for a function that returns the maximum of three numbers. Prompt Copilot/Cursor to write the logic based on tests

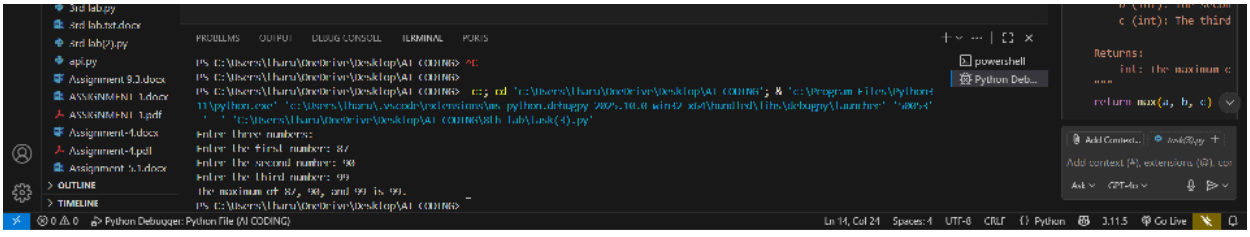
Prompt:

in 13,54,98 98 is the maximum of three numbers. 1,2,3 3 is the maximum of three numbers. now generate a function that returns the maximum of three numbers, according to the test case given. take input dynamically

Code:



OP:



Observation:

The find_maximum function takes three numbers as arguments

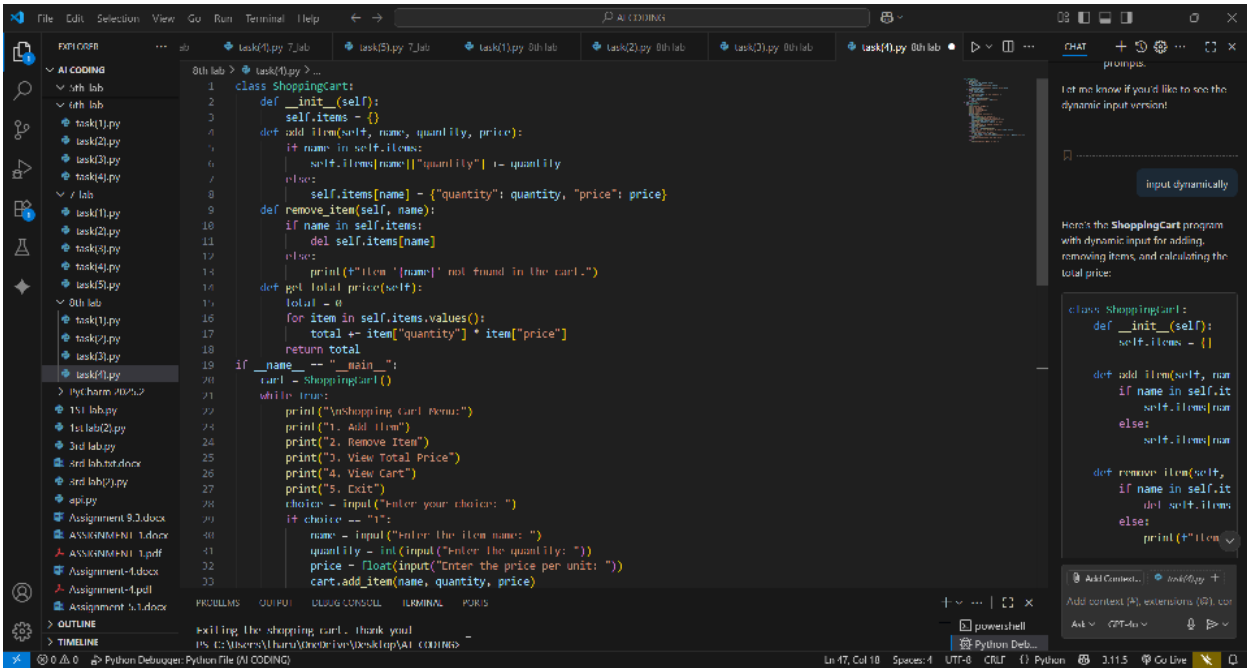
and returns the maximum using Python's built-in max() function. I have prompted to input three numbers dynamically. The program calculates the maximum of the three numbers and prints the result.

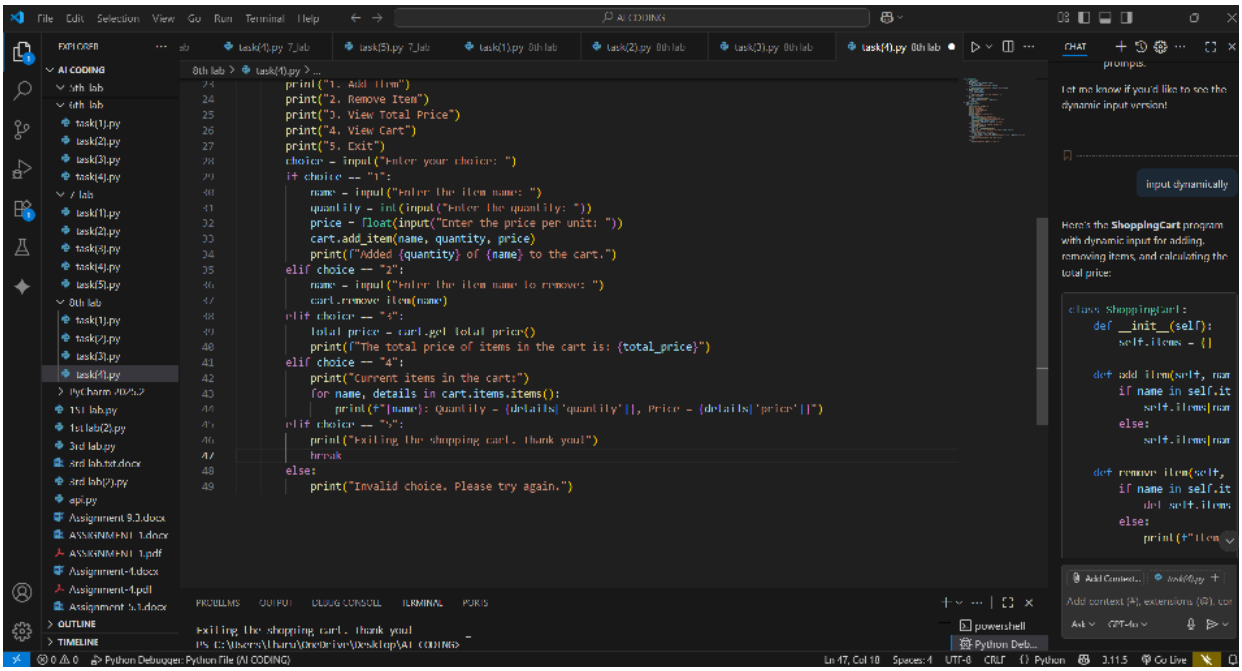
Task-4:

Use TDD to write a shopping cart class with methods to add, remove, and get total price. First write tests for each method, then generate code using AI.

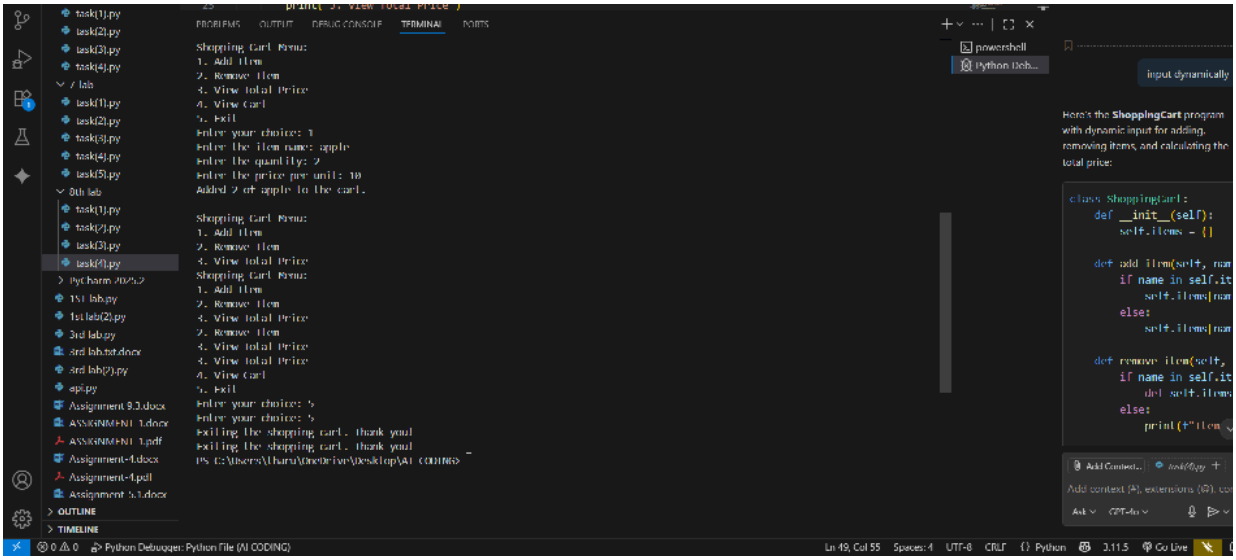
Prompt: Now generate a Python class ShoppingCart that can add items, remove items, and calculate the total price according to the given test cases. Take input dynamically from the user. Input: Add Apple with quantity 2 and price 3.0 Output: {"Apple": {"quantity": 2, "price": 3.0}}. Input: Remove Apple from the cart Output: {}. Input: Add Apple (quantity 2, price 3.0) and Banana (quantity 1, price 1.5) Output: Total price is 7.5.

Code:





OP:



Observation: The program uses input() to allow the user to interact with the shopping cart dynamically. Users can add items, remove items, view the total price, and see the cart's contents.

- oOption 1: Add an item to the cart.
- oOption 2: Remove an item from the cart.

- oOption 3: Calculate and display the total price of items in the cart.
- oOption 4: Display all items in the cart.
- oOption 5: Exit the program.

If the user tries to remove an item that doesn't exist, the program displays an appropriate message

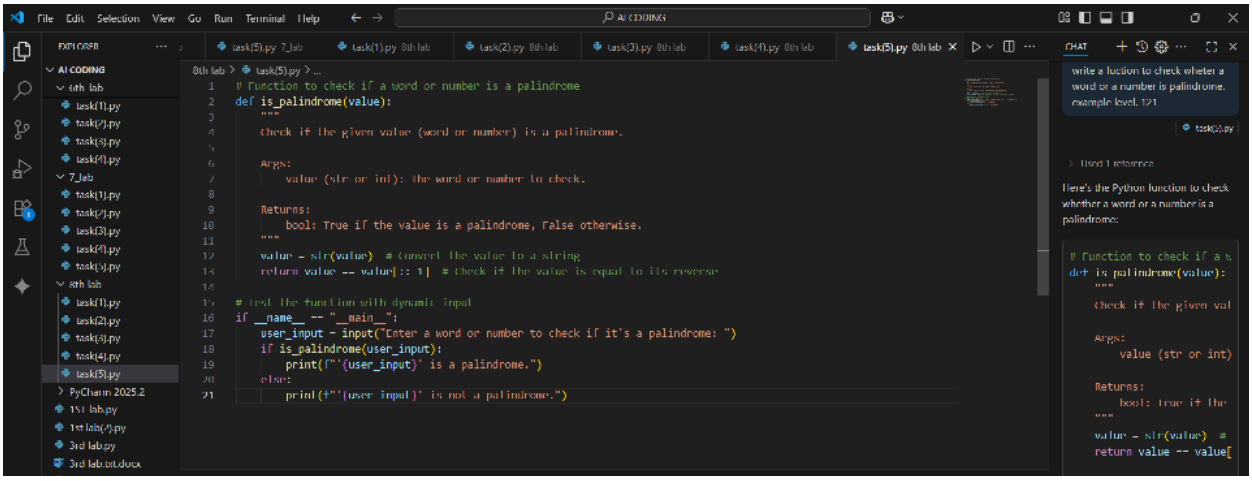
Task-5:

Write tests for a palindrome checker (e.g., is_palindrome("level") True). Let Copilot suggest the function based on test case expectations.

Prompt:

write a function to check whether a word or a number is palindrome or not. Example: level, 121

Code:



OP:

Observation:

The input value is converted to a string using str(value) to handle both words and numbers. The function checks if the string is equal to its reverse using slicing (value[::-1]). I have prompted to enter a word or number dynamically. The program prints whether the input is a palindrome