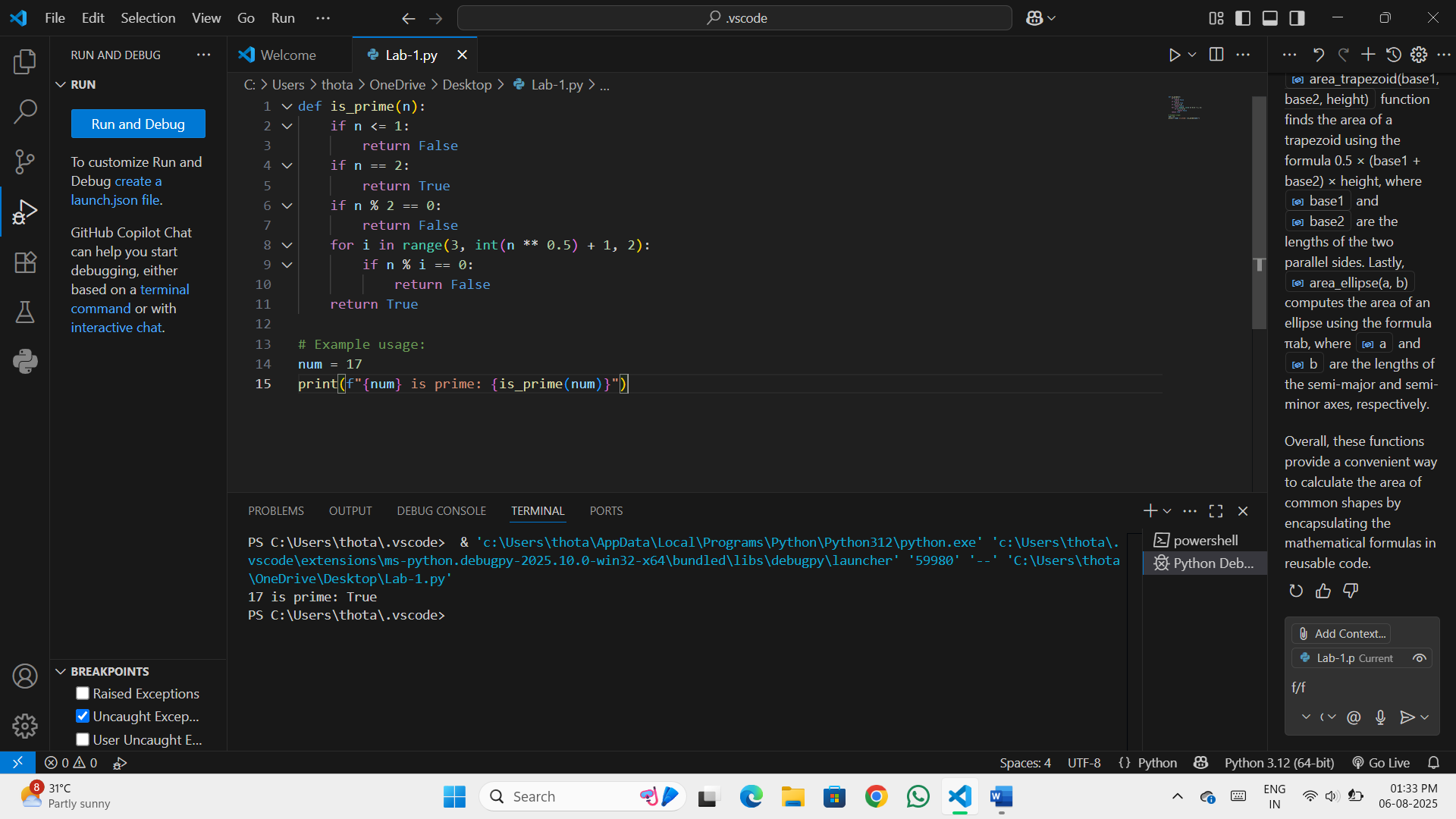
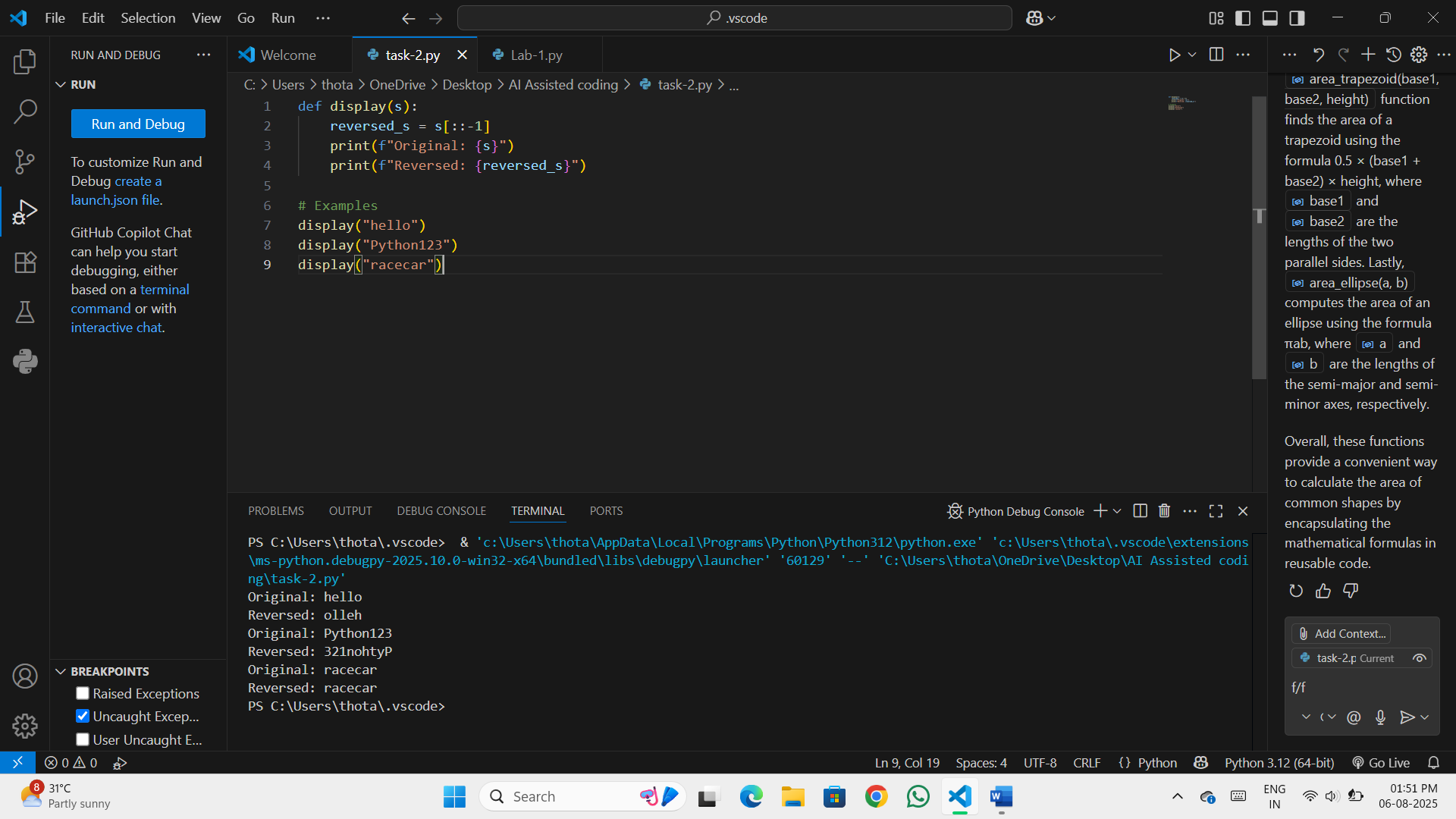
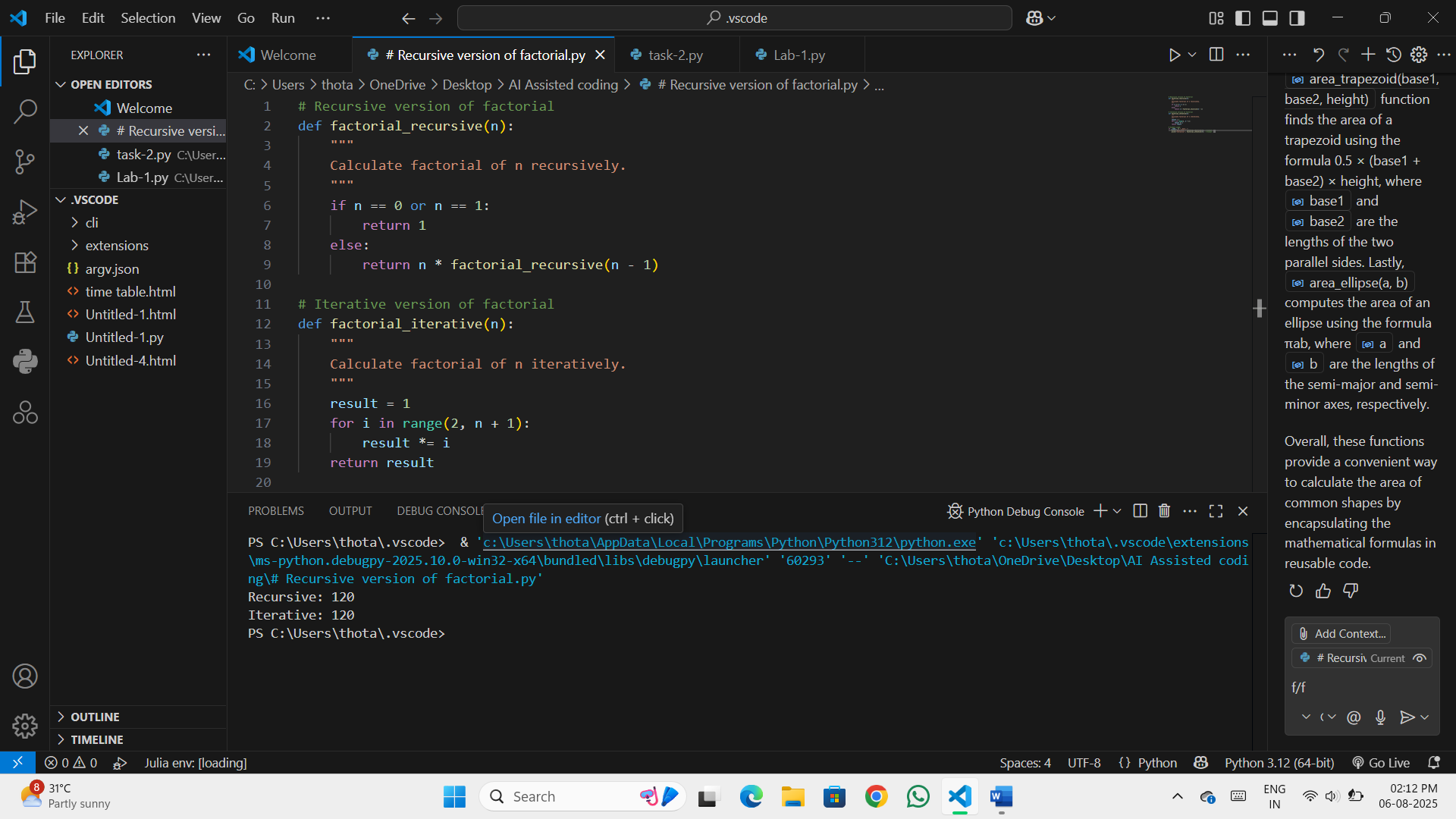
Course name: AI ASSISTED CODING

* **Task Description 2**: Use a Copilot to generate a is prime () python functions:
* **Expected output**: Functions to check primality with correct Logic.
* **Prompt2**: write a python code to check whether number is prime are not.
* 

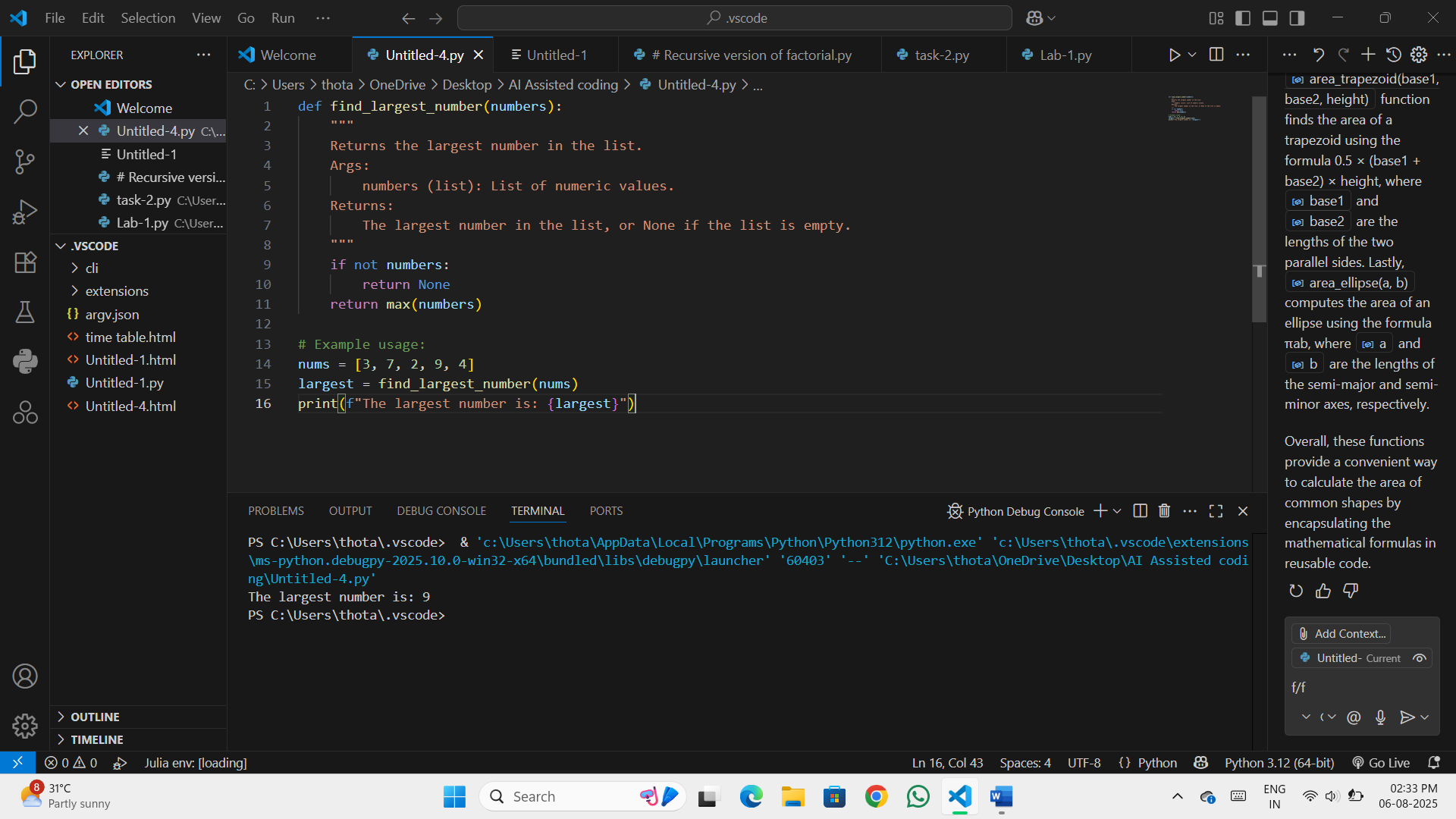
Observation: A prime number is a number greater than 1 that has no positive divisors other than 1 and itself.

* **Functionality**: Returns True for prime numbers and False otherwise.
* **Suitable for checking primality of large numbers due to reduced iterations.**
* **Task Description 3**: write a comment like # functions to Reverse a String and use copilot to generate the functions:
* **Expected output:** Auto-completed reverse function.
* **Prompt3**: write a python code to Reverse a string.

Observation:

* **Logic: Uses Python slicing [::-1] to reverse the string efficiently.**
* **Simplicity: One-liner function; concise and readable.**
* **Functionality: Works for letters, numbers, symbols, and even empty strings.**
* **Task Description 4**: Generate both Recursive and Iterative version of a factorial using comments :
* **Expected output:** Two Working Factorial of Implementation.
* **Prompt4:** write a python code for both recursive and Iterative factorial.Observation:
* **Recursive Version**:
  + Elegant and mirrors the mathematical definition.
  + May cause stack overflow for large n due to deep recursion.
  + Time complexity: **O(n)**; Space complexity: **O(n)** (due to call stack).
* **Iterative Version**:
  + More memory-efficient and avoids recursion limits.
  + Preferred for large values of n.
  + Time complexity: **O(n)**; Space complexity: **O(1)**.
* **Both Implementations**:
  + Correctly handle base cases (0! = 1, 1! = 1).
  + Produce identical results for valid non-negative integers.
* **Task Description 5:** Use Copilot to find the largest Number in a List and Access code quality and efficiency:

**Expected output:** a valid function with your revie**Prompt5:** write a python code to find largest number. Prompt: Generate a largest number

Prompt: Generate the python code of largest number

**Observation:**

* **Correctness: Accurately finds the largest number by comparing each element.**
* **Edge Case Handling: Returns None for an empty list, avoiding errors.**
* **Efficiency:**
  + **Time complexity: O(n) — linear scan through the list.**
  + **Space complexity: O(1) — uses constant extra space.**
* **Code Quality:**
  + **Clear variable naming (largest, numbers).**