Python Programming Lab Task Sets

Name: Mohammed Adnan

2503B05107

# Set 2

1. Q1. Write a Python function `is\_prime(n)` to check if a number is prime. Use an AI coding tool to generate and justify a set of test cases before implementation. Pass all generated test cases.

(Each question carries 5 marks)

Prompt: Make a Python function called is\_prime(n) to check if a number is prime.

But before writing the code, first create test cases.

Explain in a simple way why you chose each test case (like negative numbers, 0, 1, small primes, small composite numbers, and big numbers).

After explaining the test cases, write the function and make sure it passes all those tests.

Code:

def is\_prime(n: int) -> bool:

if n <= 1:

return False

if n <= 3:

return True

if n % 2 == 0 or n % 3 == 0:

return False

i = 5

while i \* i <= n:

if n % i == 0 or n % (i + 2) == 0:

return False

i += 6

return True

test\_values = {

"negative": -5,

"zero": 0,

"one": 1,

"small\_prime\_2": 2,

"small\_prime\_3": 3,

"small\_prime\_5": 5,

"small\_composite\_4": 4,

"small\_composite\_6": 6,

"small\_composite\_9": 9,

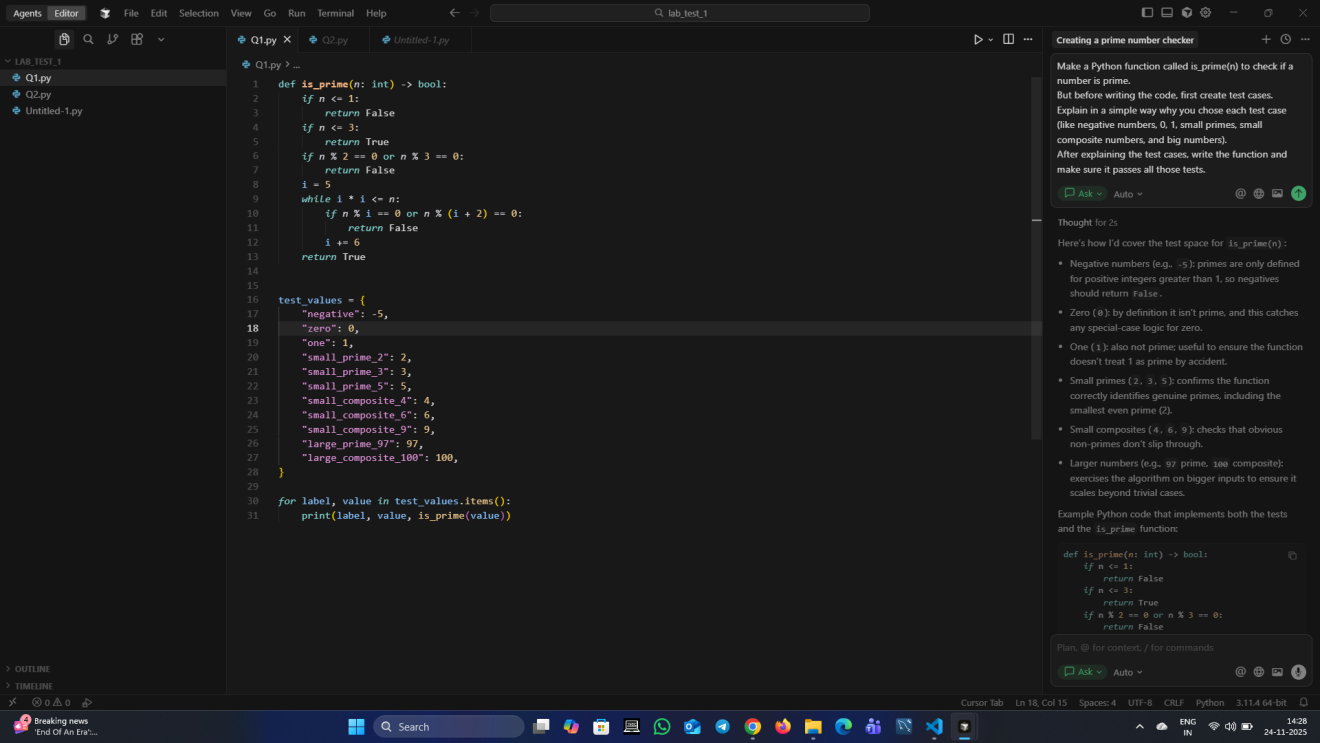
"large\_prime\_97": 97,

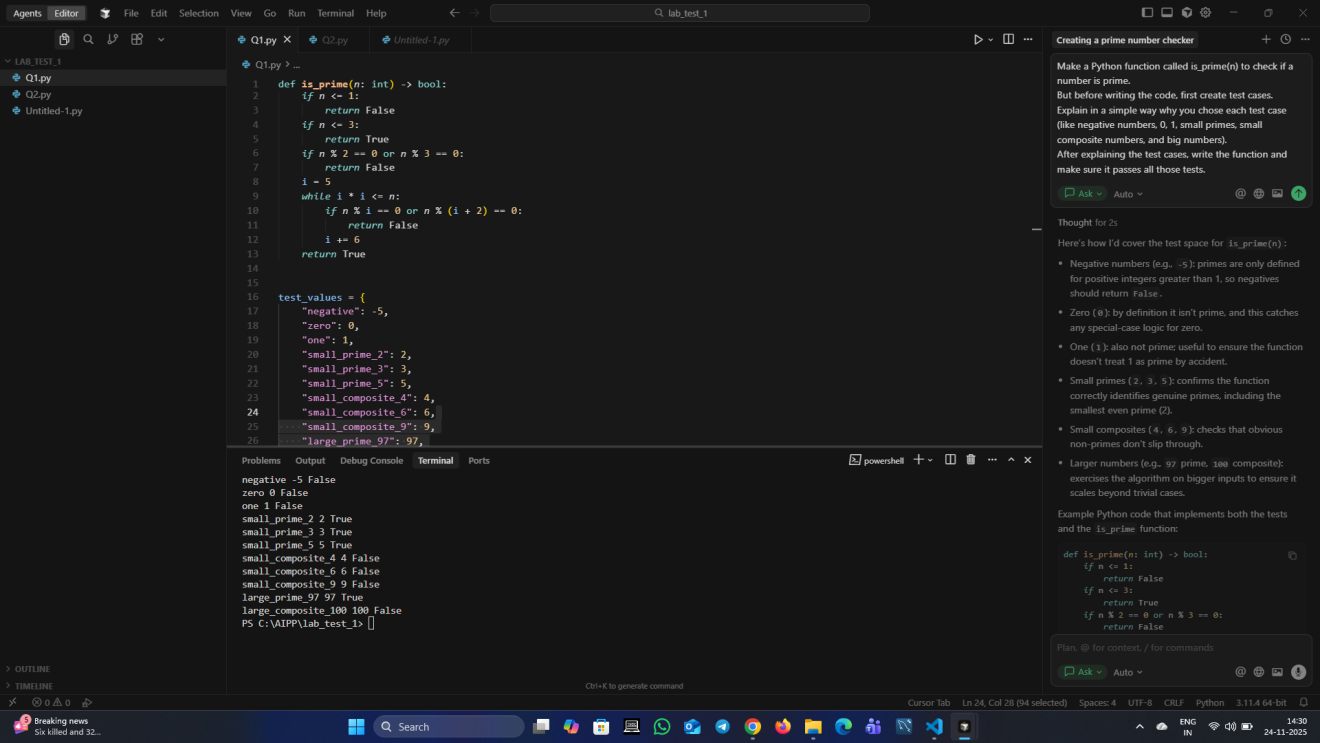
"large\_composite\_100": 100,

}

for label, value in test\_values.items():

print(label, value, is\_prime(value))





1. Q2. Develop a Python class `Rectangle` with methods for area and perimeter. Use an AI assistant for auto-generating docstrings and inline comments. Compare them with your manual documentation.

(Each question carries 5 marks)

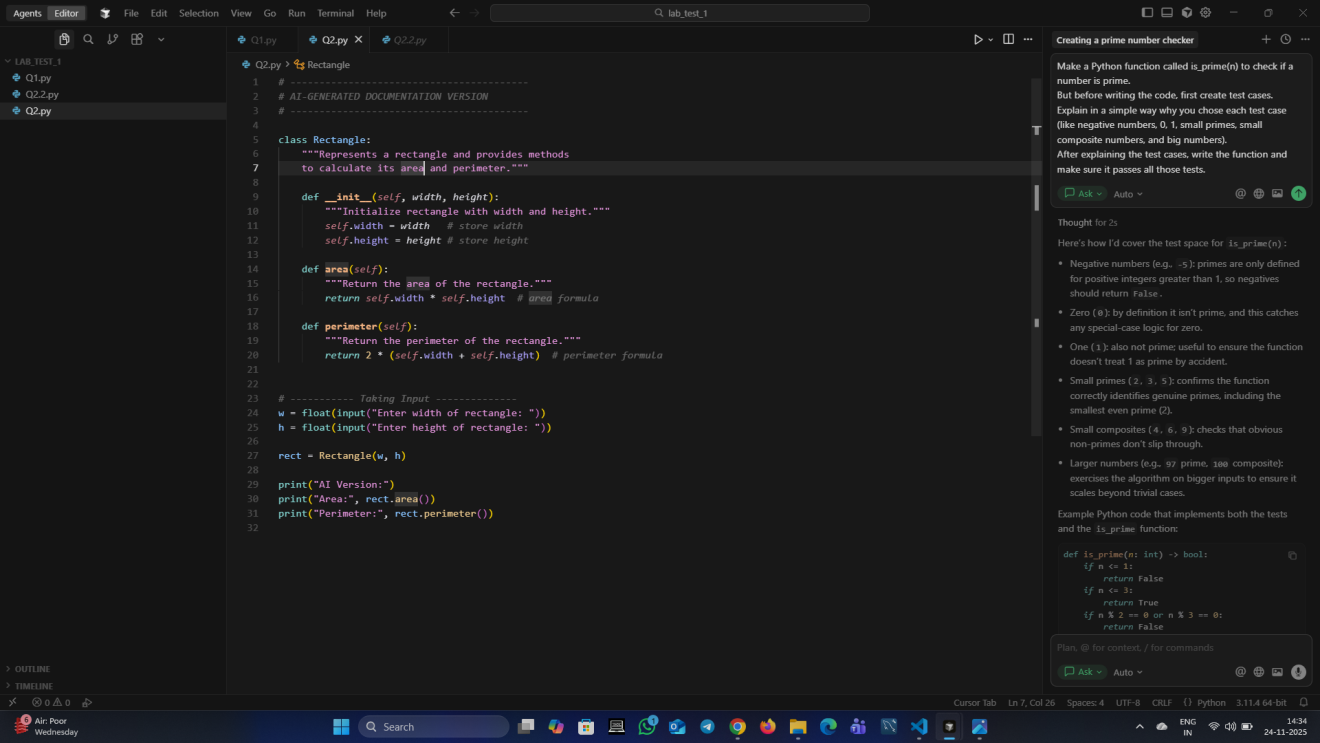
Prompt: Create a Python class called Rectangle with methods to calculate area and perimeter.

Before writing the class, ask the AI assistant to automatically generate docstrings and inline comments for the code.

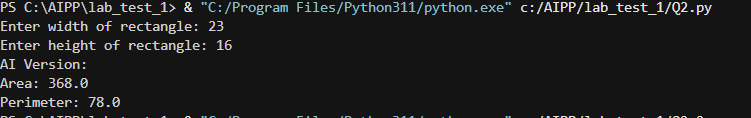
Then write your own manual docstrings and comments.

Finally, compare the AI-generated documentation with your manual documentation in a simple explanation

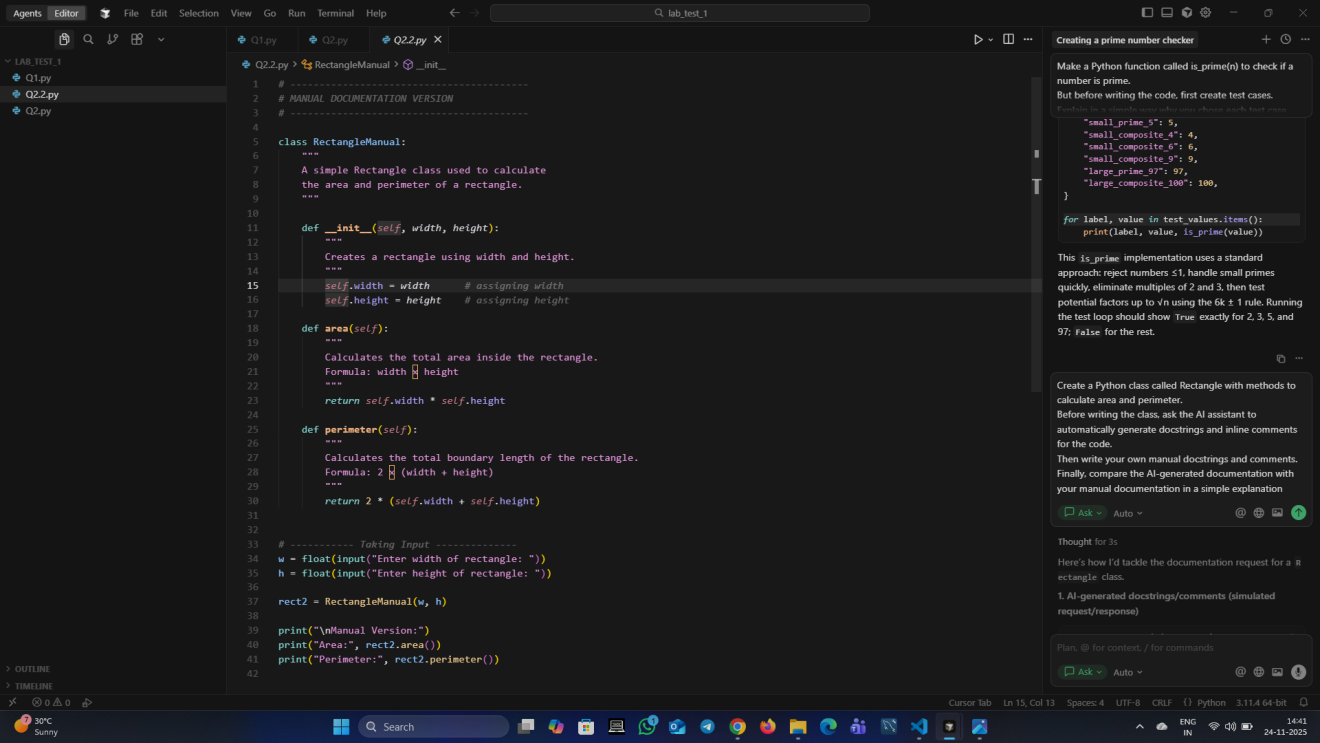
AI Generated



Output:



Manual:



Output:

