**SR UNIVERSSITY**

**AI ASSISTED CODING**

**ASSIGNMENT 12.1**

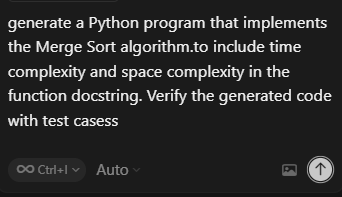
**NAME :** MOHAMMED MUTAKABIR HUSSAIN

**H.NO :** 2503A51L35

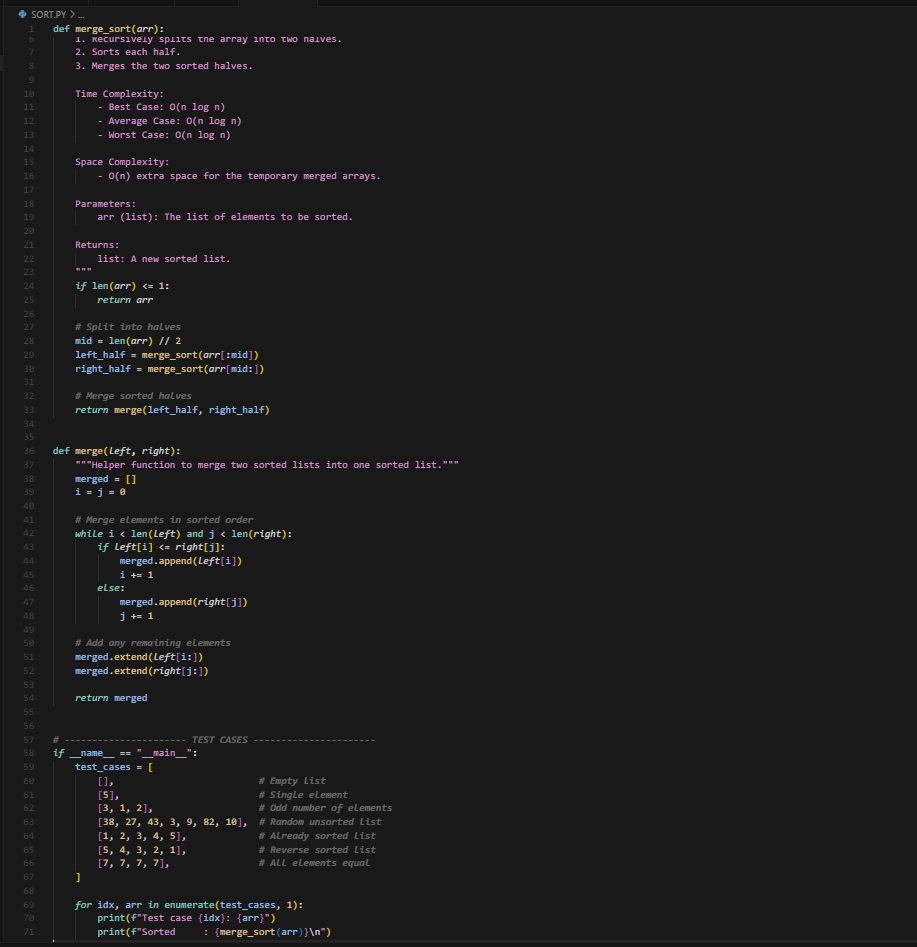
**TASK 1**

**Task Description #1 (Sorting – Merge Sort Implementation)**

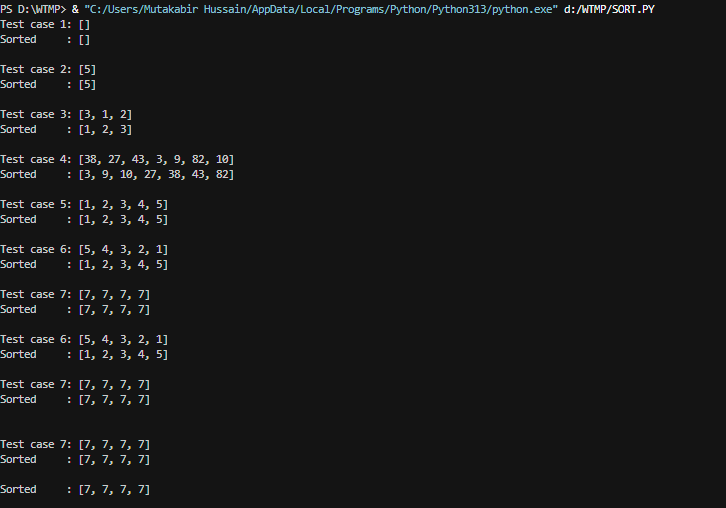
* Task: Use AI to generate a Python program that implements the Merge Sort algorithm.
* Instructions:
  + Prompt AI to create a function merge\_sort(arr) that sorts a list in ascending order.
  + Ask AI to include time complexity and space complexity in the function docstring.
  + Verify the generated code with test cases.

**PROMPT:**  


**CODE:**



**OUTPUT:**

****

**OBSERVATION:**

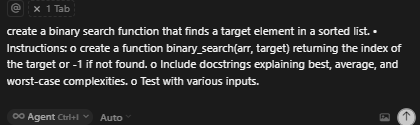
I HAVE OBSERVED THAT WITH A PROPER PROMPT AI HAS GENERATED A PYTHON PROGRAM WHICH IMPLEMENTS THE MERGE SORT ALGORITHM WITH ASCENDING ORDER AND ALSO INCLUDING TIME AND SPACE COMPLEXITY IN FUNCTION DOCSTRING ALSO VERIFIED IT WITH TEST CASES.

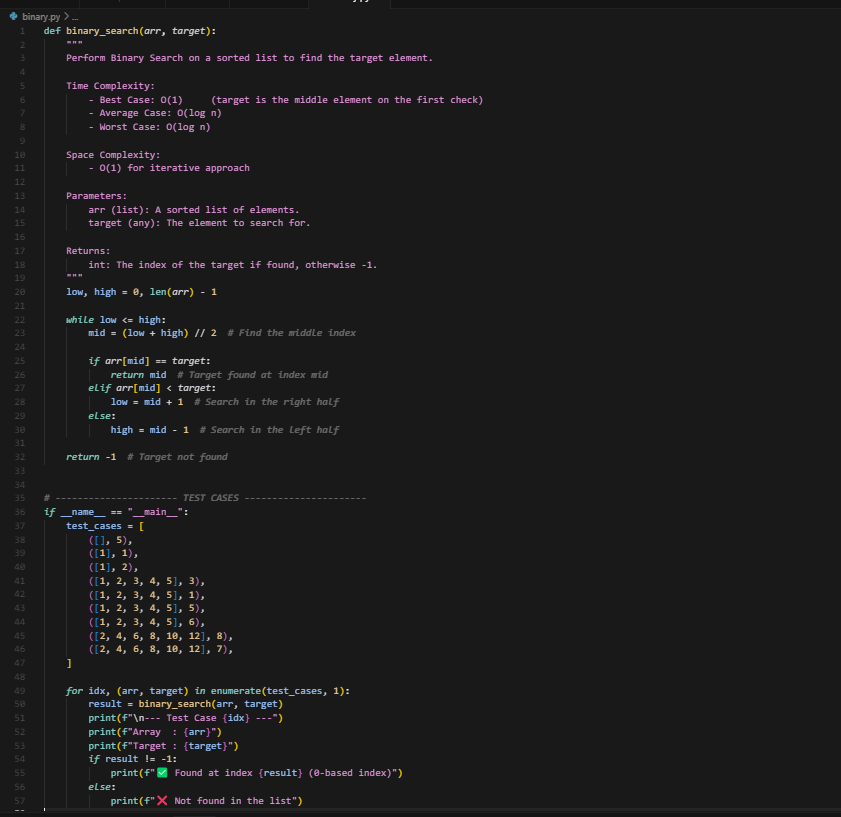
**TASK 2**

**Task Description #2 (Searching – Binary Search with AI Optimization)**

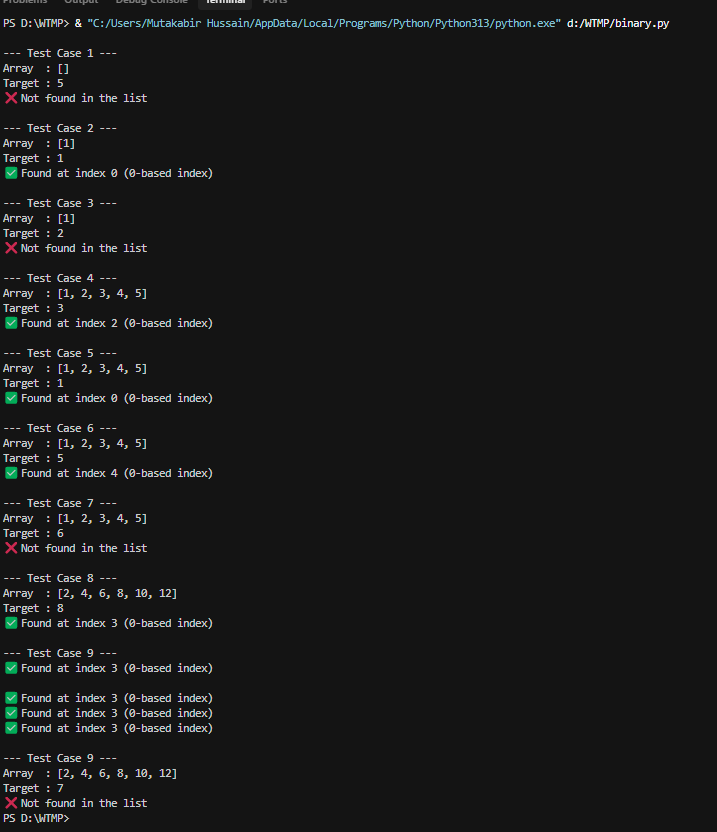
* Task: Use AI to create a binary search function that finds a target element in a sorted list.
* Instructions:
  + Prompt AI to create a function binary\_search(arr, target) returning the index of the target or -1 if not found.
  + Include docstrings explaining best, average, and worst-case complexities.
  + Test with various inputs.

**PROMPT:**



**CODE:** 

**OUTPUT:**



**OBSERVATION:**

We have observed that ai has generated a binary search function using python

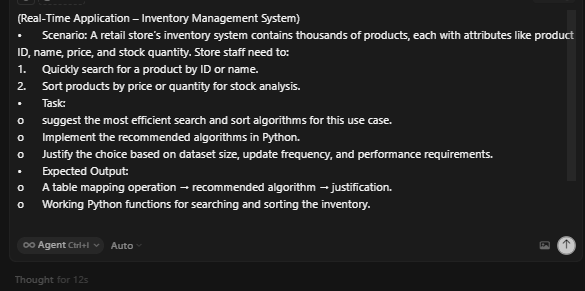
code implementing binary search with AI-generated comments and docstrings.

**TASK 3**

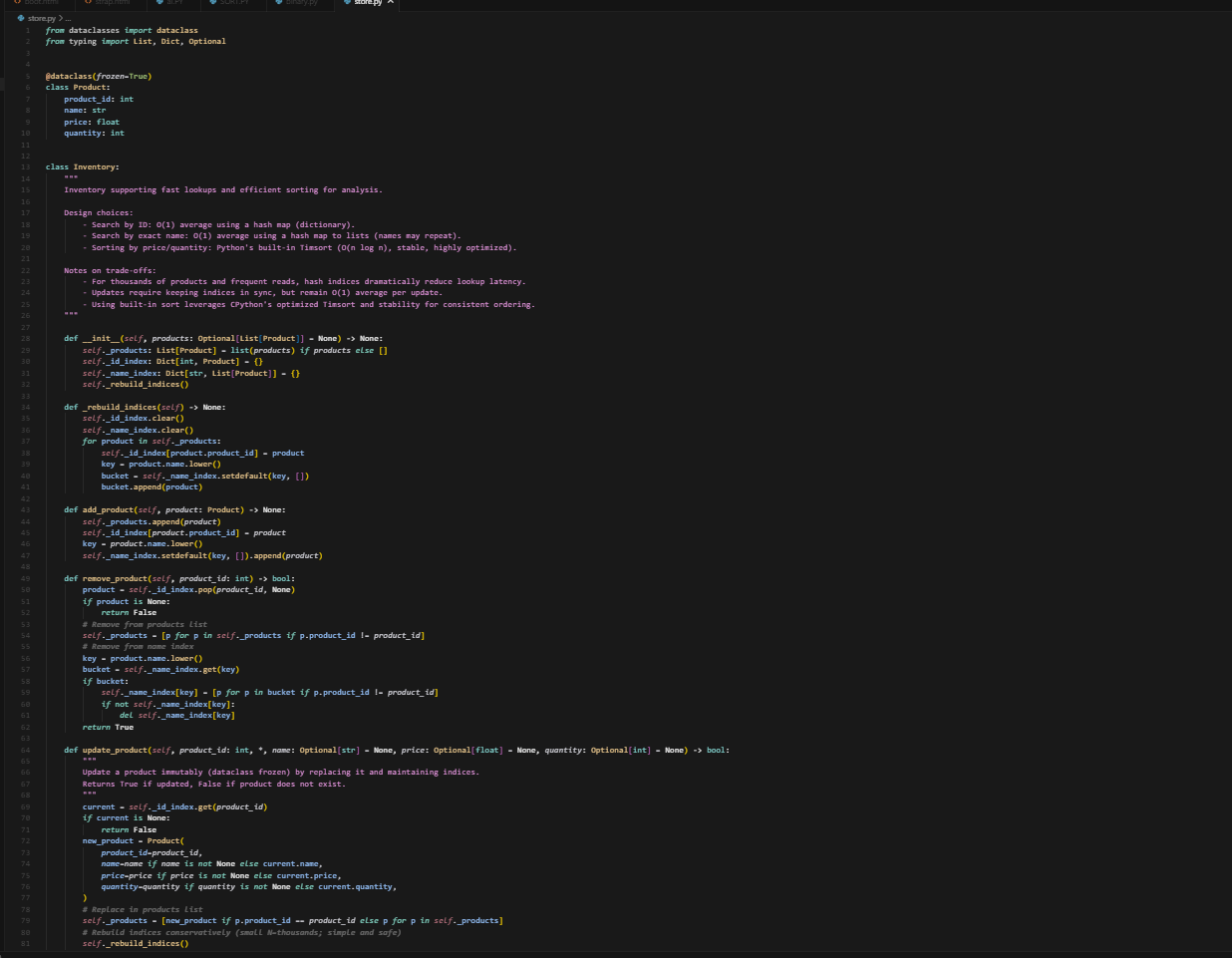
**Task Description #3 (Real-Time Application – Inventory Management System)**

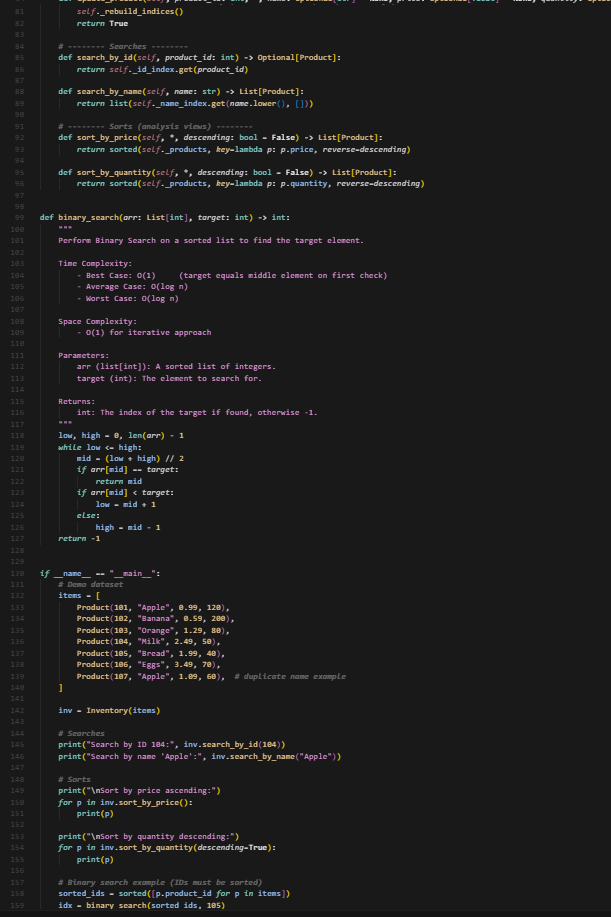
* Scenario: A retail store’s inventory system contains thousands of products, each with attributes like product ID, name, price, and stock quantity. Store staff need to:
  1. Quickly search for a product by ID or name.
  2. Sort products by price or quantity for stock analysis.
* Task:
  1. Use AI to suggest the most efficient search and sort algorithms for this use case.
  2. Implement the recommended algorithms in Python.
  3. Justify the choice based on dataset size, update frequency, and performance requirements.

**PROMPT:**



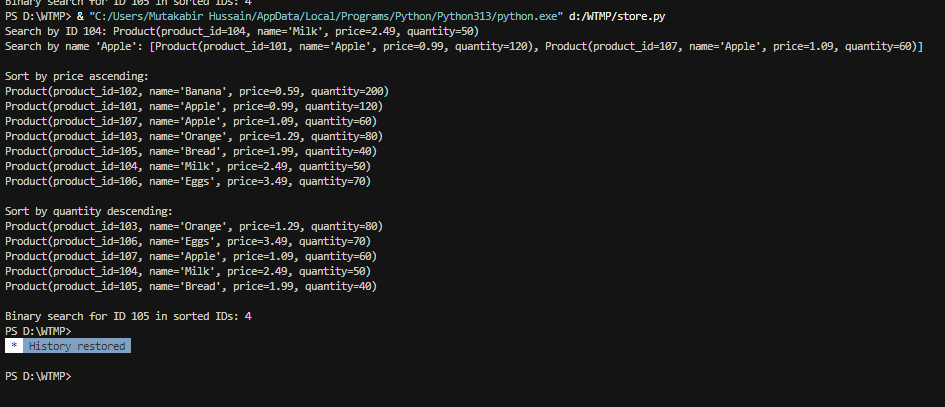
**CODE:**







**OUTPUT:**



**OBSERVATION**: Ai has generated a python code based on the Scenario

And suggested the most efficient search and sort alogorithms for this use case