**ASSIGNMENT-12.1**

**SK JAVID AMAN**

**2503A52L17**

**Algorithms with AI Assistance – Sorting, Searching, and**

**Optimizing Algorithms**

**Lab Objectives:**

**\* Apply AI-assisted programming to implement and optimize**

**sorting and searching algorithms.**

**\* Compare different algorithms in terms of efficiency and use**

**\* Understand how AI tools can suggest optimized code and**

**complexity improvements.**

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

**\* Task: Use AI to generate a Python program that implements the**

**Merge Sort algorithm.**

**\* Instructions:**

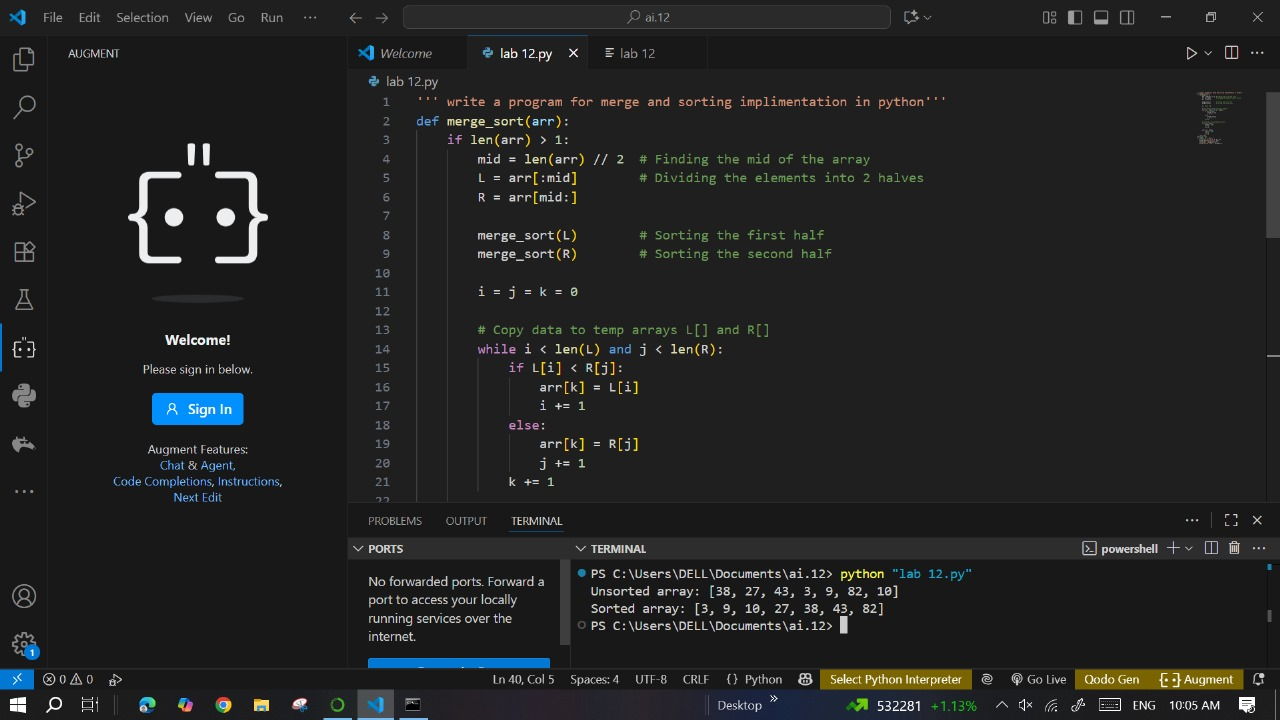
**o Prompt AI to create a function merge\_sort(arr) that sorts a**

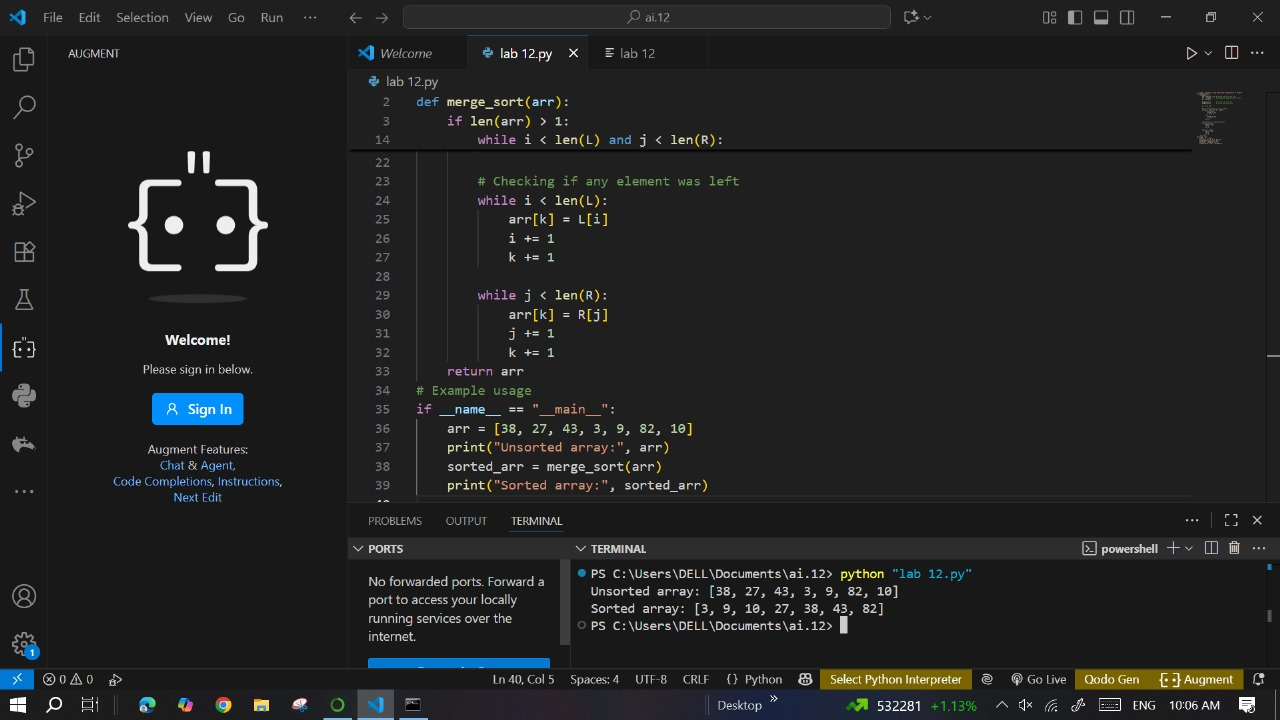
**list in ascending order.**

**o Ask AI to include time complexity and space complexity**

**in the function docstring**

**o Verify the generated code with test cases.**

****

****

**Prompt:**

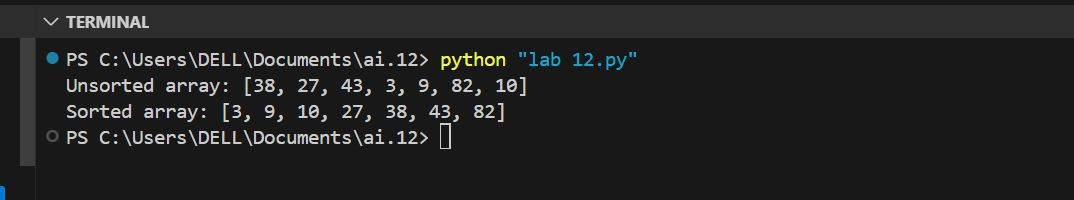
**A black background with orange text

AI-generated content may be incorrect.**

**Expected output:**

**o A functional Python script implementing Merge Sort with**

**proper documentation**

****

**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:**

**o Prompt AI to create a function binary\_search(arr, target)**

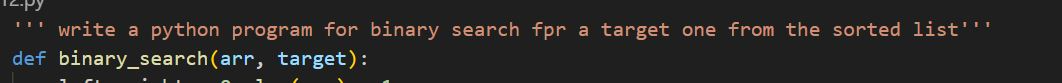
**returning the index of the target or -1 if not found.**

**o Include docstrings explaining best, average, and worst-**

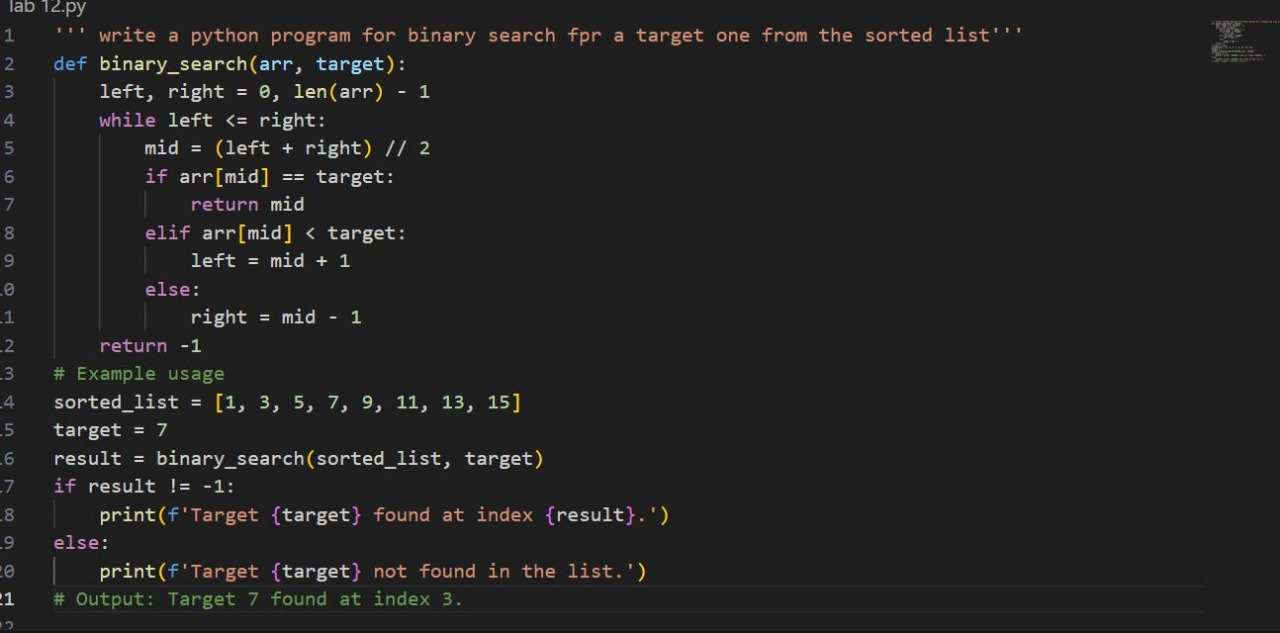
**case complexities.**

**o Test with various inputs.**

**Prompt:**

****

**Code:**

****

**\* Expected Output:**

**o Python code implementing binary search with AI-**

**generated comments and docstrings.**

**A black screen with white text

AI-generated content may be incorrect.**

**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:**

**o Use AI to suggest the most efficient search and sort**

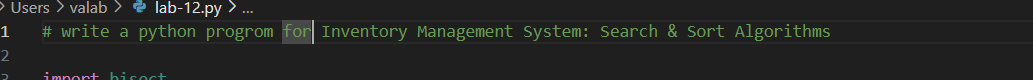
**algorithms for this use case.**

**o Implement the recommended algorithms in Python.**

**o Justify the choice based on dataset size, update frequency,**

**and performance requirements.**

**Prompt:**

****

**A computer screen shot of a program

AI-generated content may be incorrect.**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**\* Expected Output:**

**o A table mapping operation → recommended algorithm →**

**justification.**

**o Working Python functions for searching and sorting the**

**inventory.**

**Deliverables (For All Tasks)**

**1. AI-generated prompts for code and test case generation.**

**2. At least 3 assert test cases for each task.**

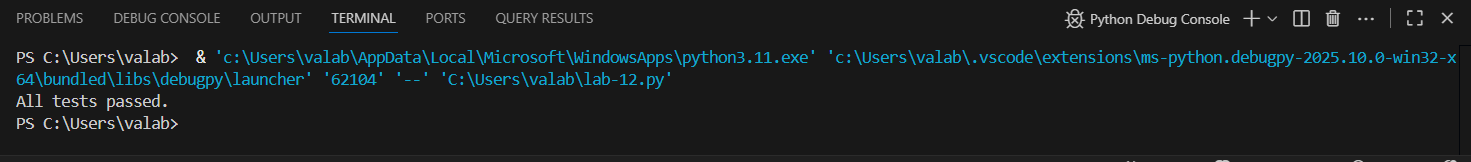
**3. AI-generated initial code and execution screenshots.**

**4. Analysis of whether code passes all tests.**

**5. Improved final version with inline comments and explanation.**

**6. Compiled report (Word/PDF) with prompts, test cases, assertions,**

**code, and output.**

****