AI ASSISTED CODING

Name:CH.Srikruthi

Enrollno:2403a510H0

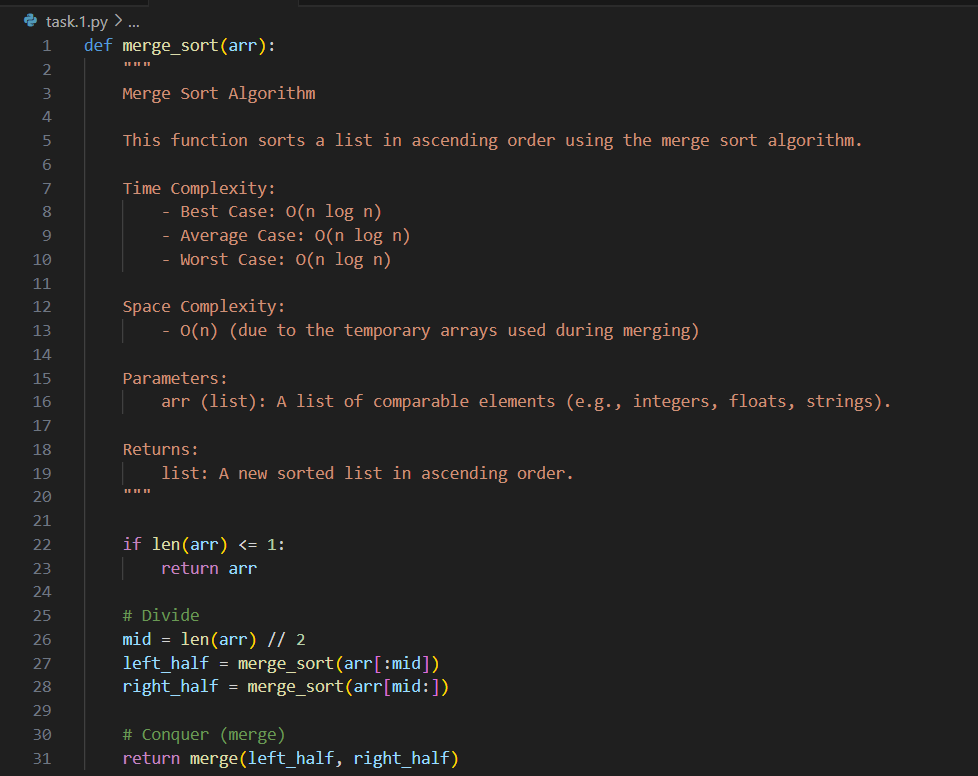
Assignment:12.1

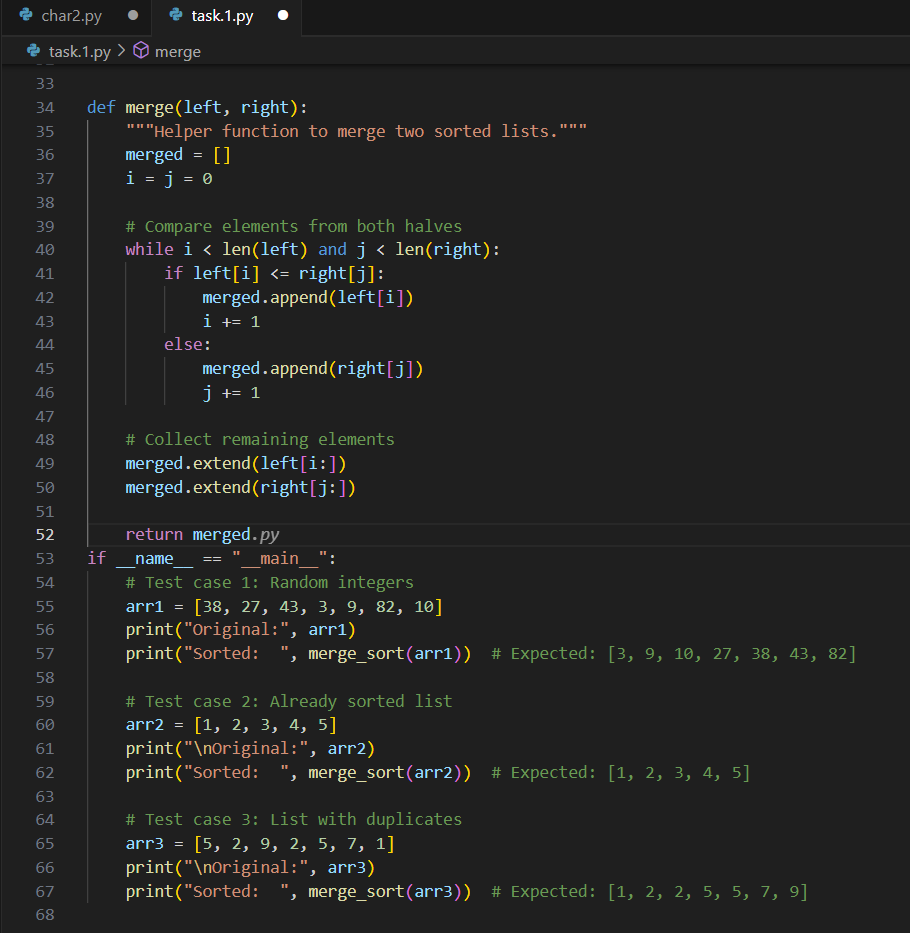
**Task1:**

**Prompt:**

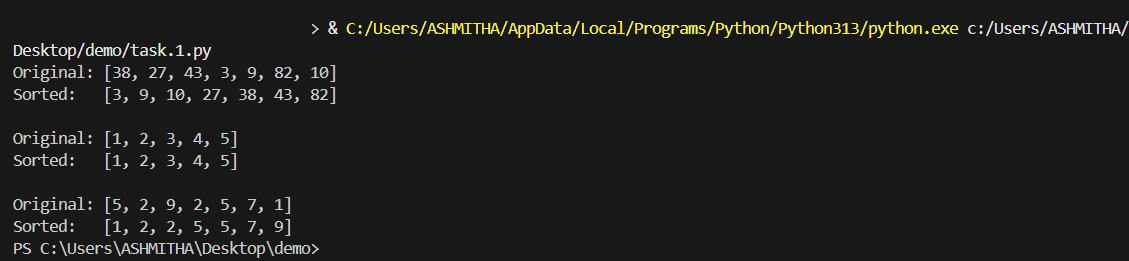
Write a Python program to implement merge sort. Create a function merge\_sort(arr) that sorts a list in ascending order. Add a docstring with time complexity and space complexity. Finally, test the function with 3 test cases.

**Code:**

****

****

**Output:**

****

**Observation:**

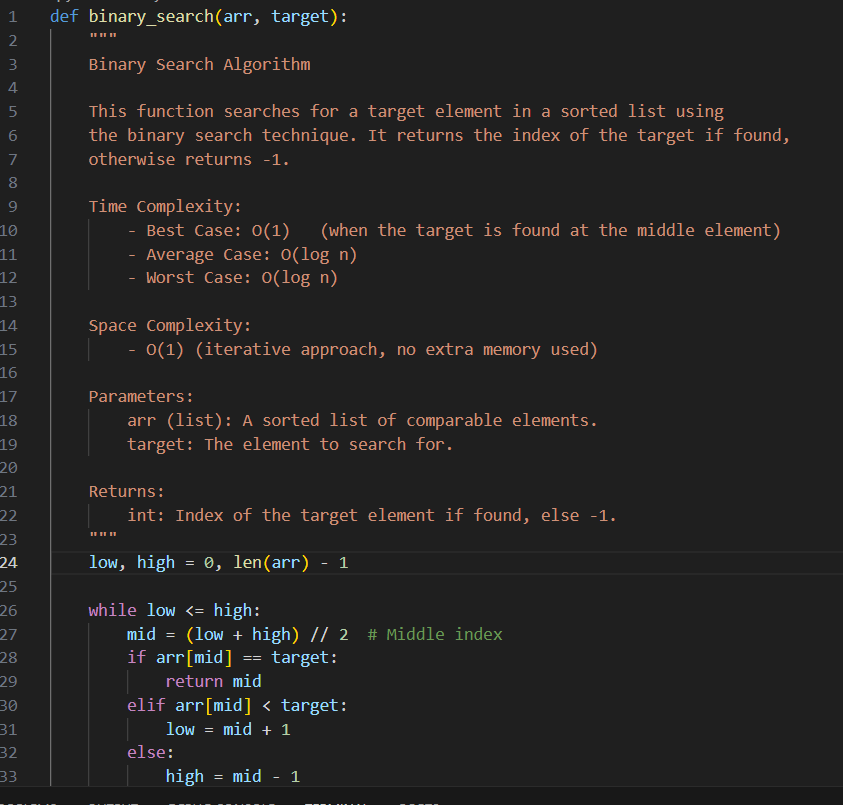
The program correctly implements the Merge Sort algorithm using recursion and a helper merge function. It includes clear documentation of time and space complexity and is tested with three cases: random integers, a sorted list, and duplicates. The results are accurate, proving the implementation is efficient and reliable.

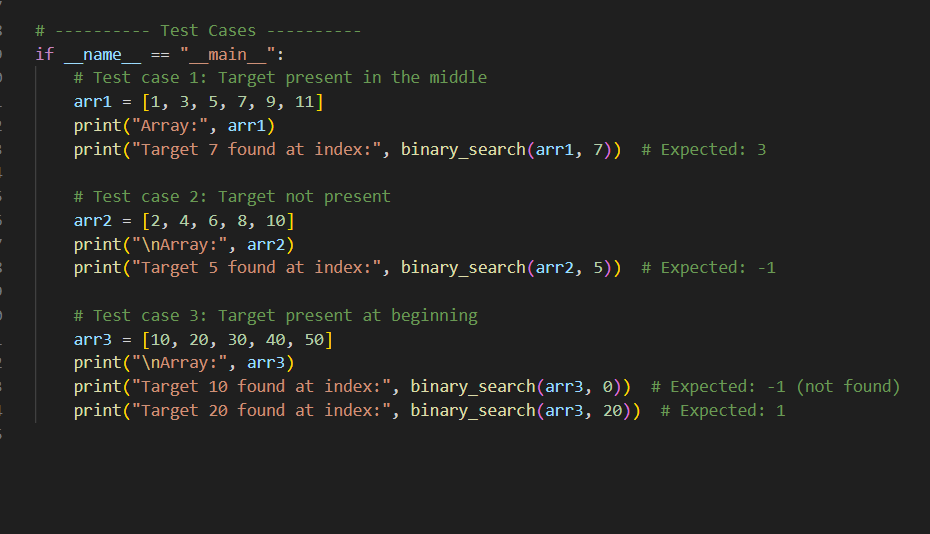
**Task2:**

**Prompt:**

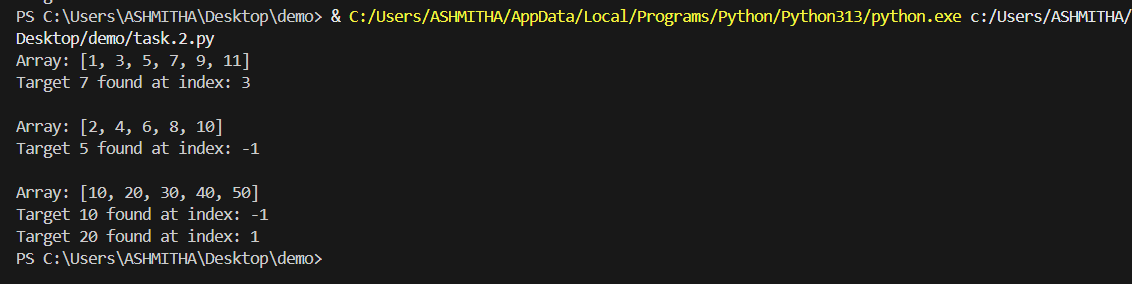
Write a Python program to implement binary search. Create a function binary\_search(arr, target) that returns the index of the target element or -1 if not found. Add a docstring with best, average, and worst-case time complexities. Finally, test the function with 3 different inputs.

**Code:**

****

****

**Output:**

****

**Observation:**

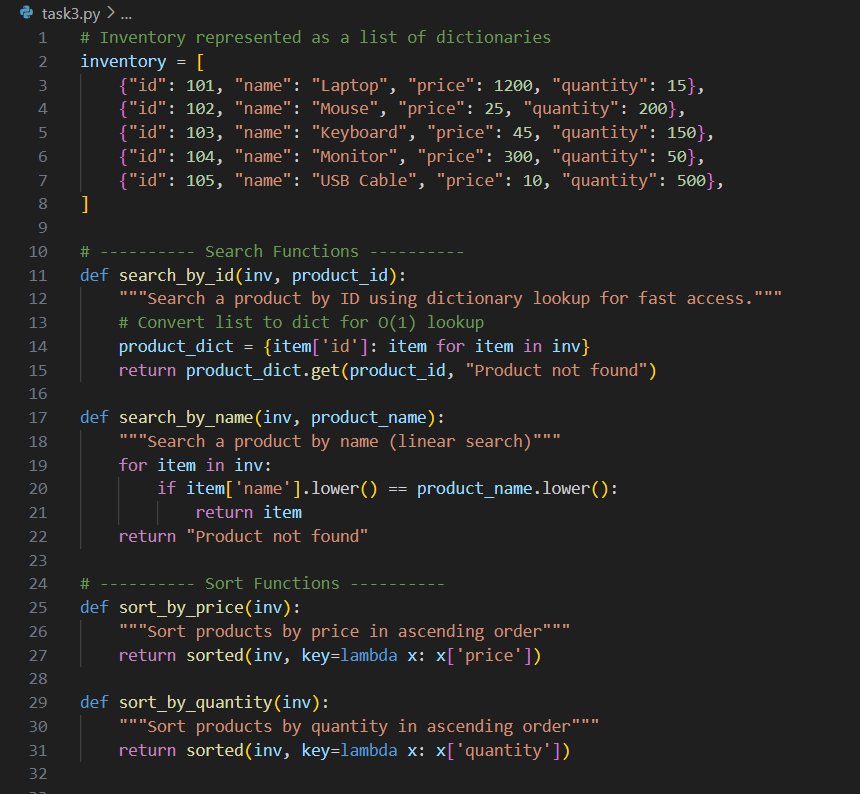
The program correctly implements the Binary Search algorithm using an iterative approach. It efficiently searches for a target in a sorted list and returns the index if found, or -1 otherwise. The function is well-documented with time and space complexities, and the three test cases demonstrate its correctness for cases where the target is present, absent, or at different positions in the list.

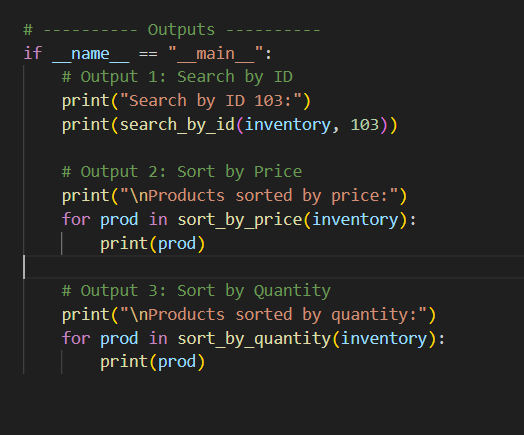
**Task3:**

**Prompt:**

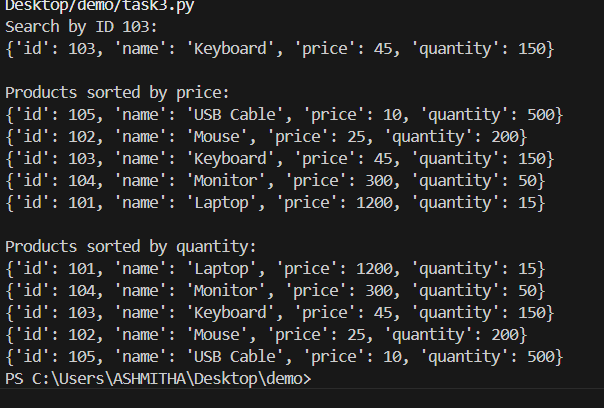
Write a Python program for an inventory management system. Include functions to search products by ID or name and to sort products by price or quantity. Justify the choice of algorithms for large datasets and provide 3 sample outputs.

**Code:**

****

****

**Output:**

****

**Observation:**

The program efficiently manages an inventory of products using a list of dictionaries. It allows quick search by ID with a dictionary lookup and by name with a linear search. Products can be sorted by price or quantity using Python’s built-in sorted() function. The outputs demonstrate accurate search results and correctly sorted lists, making it suitable for small to medium-sized inventories.