a) Problem specification – What problems are solving?

Data Management:

- Creating a list: A predefined data set is added to the program.
- Adding elements: It enables users to add new elements to the existing list.
- Deleting elements: It facilitates removing specific elements from the list based on user input.
- Updating elements: It allows users to modify existing elements in the list with new values.

Data Organization:

- Sorting: It provides functionalities to sort the list using two algorithms (selection sort and merge sort) for comparison purposes. This helps in organizing the list in a specific order (e.g., ascending or descending).

Data Search:

- Linear search: It implements a linear search algorithm to find a specific element within the list.
- Binary search: It utilizes a binary search algorithm to efficiently search for an element in a sorted list. This is typically faster than linear search for large datasets.
- Hash search (potential): While not explicitly shown in the searchList pseudocode, the program might potentially use a hash table for searching elements. Hash tables can offer faster average search times compared to linear search, especially for large datasets.

Data Analysis:

- Comparison of sorting algorithms: It helps users compare the performance of selection sort and merge sort by displaying the number of comparisons made during each sorting process.
- Overall, the program tackles the problems of managing, organizing, searching, and analyzing data within a list structure. It provides various functionalities to create, modify, search, and sort the list elements, along with the ability to compare different sorting algorithms.