Q1: In a university, there is an upcoming final exam for a large class, and the exam hall has a limited number of seats arranged in rows and columns. You need to assign seats to students while satisfying the following constraints:

a) Each student must be assigned a unique seat.

b) Some students have specific preferences for where they want to sit.

c) Certain seats are reserved for students with disabilities, and these seats cannot be assigned to other students.

d) Students with the same last name cannot be assigned adjacent seats in the same row.

Provide a recursive algorithm to find a valid seating arrangement for the students while adhering to all constraints.

Q2) Given a linked list and a value x, partition it such that all nodes less than x come first, then all nodes with a value equal to x, and finally nodes with a value greater than or equal to x. The original relative order of the nodes in each of the three partitions should be preserved. The partition must work in place.

Input : 1->4->3->2->5->2->3,

x = 3

Output: 1->2->2->3->3->4->5

Q3) Imagine you work for an online car dealership, and your company has recently received a large shipment of cars of various makes and models. Your task is to create a C++ program to manage the inventory and allow customers to easily find the cars they are interested in. You need to sort the cars based on their models and prices.

Your program should allow users to perform the following operations:

1) Add Cars: Add new cars to the inventory.
2) List Cars by Model: Display a list of all cars sorted by model (alphabetically).
3) List Cars by Price: Display a list of all cars sorted by price (in ascending order).
4) List Cars by Engine size (CC): Display a list of all cars sorted by CC (in descending order).
5) Remove a Car: Remove a car from the inventory.