Project Prompt: Car Repair Tracking System

1. Create three structs representing Cars, Repair Records, and Customers, with the specified attributes.
   - Car struct:
     - CarID (int), Model (string), Manufacturer (string), Year (int), LicensePlate (string), CurrentMileage (int)
   - RepairRecord struct:
     - RecordID (int), CarID (int), Date (string), Technician (string), Description (string), Cost (float)
   - Customer struct:
     - CustomerID (int), FirstName (string), LastName (string), ContactNumber (string), Email (string), Address (string)

2. Implement an enum RepairStatus for the status of a repair record.

3. File Handling:
   - Store data in text files.
   - Define file formats for storing Cars, Repair Records, and Customers data.

4. Operations on Data:
   Cars:
   - Add a New Car
   - Find Car by ID or License Plate
   - Update Car Information
   - Delete Car

   Repair Records:
   - Add a New Repair Record
   - Find Repair Records for a Car
   - Delete Repair Record

   Customers:
   - Add a New Customer
   - Find Customer by ID or Contact Number
   - Update Customer Information
   - Delete Customer

5. String Operations and Pointers:
   - Use standard string functions (strcpy, strcat, strlen, strcmp) for string manipulation.
   - Utilize pointers for efficient string handling.

6. Function Pointers and Macros:
   - Use function pointers in a struct for customizable functionalities.
   - Utilize a macro function for code snippets expanded at compile time.

7. Library and Implementation Files:
   - Create a library file (e.g., car_repair_library.c) for common functionalities.
   - Include functions for file I/O, string operations, and other utilities.
   - Implement header files for encapsulation, reusability, and modularity.

8. Pointer to Pointer (Double Pointer):
   - Use double pointers as needed in the project context.

9. Calculation of Summarized Values:
   - Total Cost of Repairs: Sum costs of repair records.
   - Average Mileage Across Cars: Calculate the average mileage.

- Number of Cars in the System: Count total cars.

Remember to handle data integrity by validating and handling edge cases.