**C++ Code**

#include <iostream>

#include <string>

#include <vector>

using namespace std;

// Define a structure for storing information about each car

struct Car {

string make;

string model;

string licensePlate;

int speed;

};

// Function to add a new car to the system

void addCar(vector<Car>& cars) {

Car newCar;

cout << "Enter make: ";

cin >> newCar.make;

cout << "Enter model: ";

cin >> newCar.model;

cout << "Enter license plate: ";

cin >> newCar.licensePlate;

cout << "Enter speed: ";

cin >> newCar.speed;

cars.push\_back(newCar);

cout << "Car added successfully!" << endl;

cout << endl;

}

// Function to remove a car from the system

void removeCar(vector<Car>& cars) {

string licensePlate;

cout << "Enter license plate of car to remove: ";

cin >> licensePlate;

for (int i = 0; i < cars.size(); i++) {

if (cars[i].licensePlate == licensePlate) {

cars.erase(cars.begin() + i);

cout << "Car removed successfully!" << endl;

cout << endl;

return;}

}

cout << "Car not found in system." << endl; }

// Function to display all cars in the system

void displayCars(vector<Car>& cars) {

for (int i = 0; i < cars.size(); i++) {

cout << "Make: " << cars[i].make << endl;

cout << "Model: " << cars[i].model << endl;

cout << "License plate: " << cars[i].licensePlate << endl;

cout << "Speed: " << cars[i].speed << endl;

if (cars[i].speed > 100)

cout<< cars[i].licensePlate << " needs to pay 50$" << endl;

cout << endl;

cout << endl;

}

}

int main() {

vector<Car> cars;

int choice;

do {

cout << "1. Add car" << endl;

cout << "2. Remove car" << endl;

cout << "3. Display all cars" << endl;

cout << "4. Exit" << endl;

cout << "Enter choice: ";

cin >> choice;

switch (choice) {

case 1:

addCar(cars);

break;

case 2:

removeCar(cars);

break;

case 3:

displayCars(cars);

break;

case 4:

break;

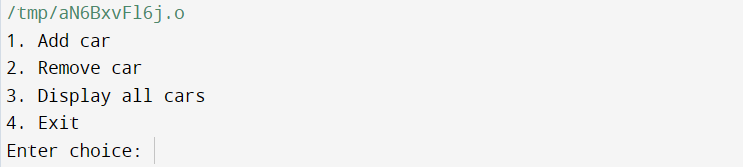
default:

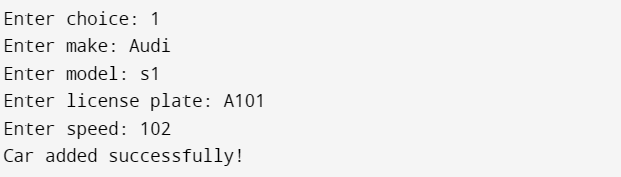
cout << "Invalid choice." << endl;}

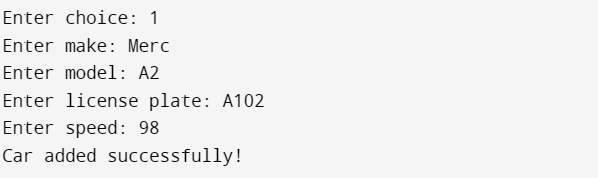
} while (choice != 4);

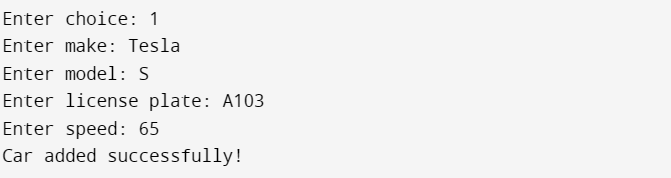
return 0; }

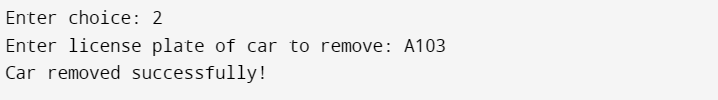
# OUTPUT

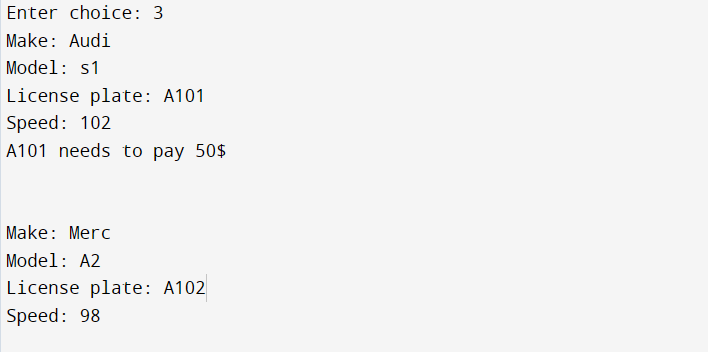














**EXIT**