Assessment 2(14/01/25)

```
#include <stdio.h>
#include <string.h>
#define MAX_CAPACITY 100
#define MAX_VEHICLES 10

// Static variable to track total vehicles
static int totalVehicles = 0;

// Structure for driver information
typedef struct {
    char name[50];
    char license[20];
} Driver;

// Structure for vehicle information
typedef struct {
    int id;
    char type[20];
    int capacity;
    int available;
    Driver driver;
} Vehicle;

// Union for maintenance tasks
typedef union {
    char engineWork[50];
    char tireChange[50];
```

```
    char oilChange[50];
} MaintenanceTasks;

// Nested union for maintenance records
typedef union {
    MaintenanceTasks tasks;
    char notes[100];
} MaintenanceRecord;

// Array to store vehicle details
Vehicle vehicles[MAX_VEHICLES];

// Function prototypes
void addVehicle();
void removeVehicle();
void viewVehicles();

int main() {
    int choice;
    while (1) {
        printf("\nTransport Management System\n");
        printf("1. Add Vehicle\n");
        printf("2. Remove Vehicle\n");
        printf("3. View Vehicles\n");
        printf("4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
```

```
            case 1:
                addVehicle();
                break;
            case 2:
                removeVehicle();
                break;
            case 3:
                viewVehicles();
                break;
            case 4:
                printf("Exiting system.\n");
                return 0;
            default:
                printf("Invalid choice. Please try again.\n");
        }
    }
}

void addVehicle() {
    if (totalVehicles >= MAX_VEHICLES) {
        printf("Vehicle limit reached. Cannot add more vehicles.\n");
        return;
    }

    Vehicle v;
    printf("Enter Vehicle ID: ");
    scanf("%d", &v.id);
    printf("Enter Vehicle Type: ");
    scanf("%s", v.type);
```

```
    printf("Enter Capacity (max %d): ", MAX_CAPACITY);
    scanf("%d", &v.capacity);
    if (v.capacity > MAX_CAPACITY) {
        printf("Capacity exceeds maximum allowed limit.\n");
        return;
    }
    printf("Enter Availability (1 for available, 0 for not available): ");
    scanf("%d", &v.available);
    printf("Enter Driver Name: ");
    scanf("%s", v.driver.name);
    printf("Enter Driver License: ");
    scanf("%s", v.driver.license);

    vehicles[totalVehicles++] = v;
    printf("Vehicle added successfully!\n");
}

void removeVehicle() {
    if (totalVehicles == 0) {
        printf("No vehicles to remove.\n");
        return;
    }

    int id, i, found = 0;
    printf("Enter Vehicle ID to remove: ");
    scanf("%d", &id);

    for (i = 0; i < totalVehicles; i++) {
        if (vehicles[i].id == id) {
```

```
            found = 1;
            break;
        }
    }

    if (found) {
        for (int j = i; j < totalVehicles - 1; j++) {
            vehicles[j] = vehicles[j + 1];
        }
        totalVehicles--;
        printf("Vehicle removed successfully!\n");
    } else {
        printf("Vehicle with ID %d not found.\n", id);
    }
}

void viewVehicles() {
    if (totalVehicles == 0) {
        printf("No vehicles available.\n");
        return;
    }

    printf("\nVehicle List:\n");
    for (int i = 0; i < totalVehicles; i++) {
        printf("ID: %d, Type: %s, Capacity: %d, Availability: %s\n",
               vehicles[i].id, vehicles[i].type, vehicles[i].capacity,
               vehicles[i].available ? "Yes" : "No");
        printf("Driver Name: %s, License: %s\n",
               vehicles[i].driver.name, vehicles[i].driver.license);
```

```
    }
}
```