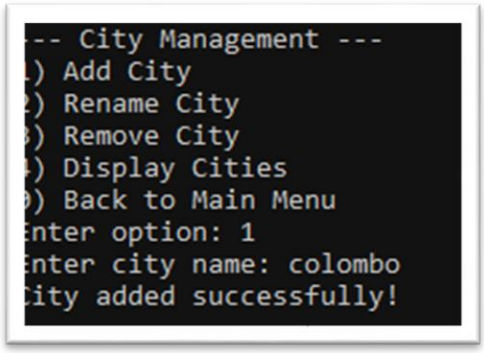
Git hub repository - https://github.com/minindulochana/computer-programming-as20240578.git

Index - as20240578

menu-driven logistics management system

Lest's see the output of the main 9 switch cases,

1) City Management




```
--- City Management ---
1) Add City
2) Rename City
3) Remove City
4) Display Cities
0) Back to Main Menu
Enter option: 2

Cities List:
[0] colombo
Enter city index to rename: 0
Enter new name: maharagama
City renamed!
```

```
--- City Management ---
1) Add City
2) Rename City
3) Remove City
4) Display Cities
0) Back to Main Menu
Enter option: 3

Cities List:
[0] maharagama
Enter city index to remove:
0
City removed!
```

```
--- City Management ---
1) Add City
2) Rename City
3) Remove City
4) Display Cities
0) Back to Main Menu
Enter option: 4

Cities List:
[0] colombo
```

2) Distance Management

```
---- Logistics Management System ----
1. City Management
2. Distance Management
3. Display vehcles
4. Delivery Request Handling
5. Cost, Time, and Fuel Calculations
6. Delivery Records
7. Finding The Least-Cost Route (Least-Distance)
8. Performance Reports
9. File Handling
10. Exit
Choose option: 2

Cities List:
[0] colombo
[1] maharagama
Enter first city index: 0
Enter second city index: 1
Enter distance in km: 34
 Distance updated successfully!

Distance Table (km)
          0     1
  0 |     0    34    colombo
  1 |    34     0    maharagama
```

3) Display vehcles

```
---- Logistics Management System ----
1. City Management
2. Distance Management
3. Display vehcles
4. Delivery Request Handling
5. Cost, Time, and Fuel Calculations
6. Delivery Records
7. Finding The Least-Cost Route (Least-Distance)
8. Performance Reports
9. File Handling
10. Exit
Choose option: 3

--- Vehicle Types ---
1) Van   | Capacity: 1000 kg | Rate: 30 LKR/km | Speed: 60 km/h | Fuel: 12 km/l
2) Truck | Capacity: 5000 kg | Rate: 40 LKR/km | Speed: 50 km/h | Fuel: 6 km/l
3) Lorry | Capacity: 10000 kg | Rate: 80 LKR/km | Speed: 45 km/h | Fuel: 4 km/l
```

4) Delivery Request Handling

```
---- Logistics Management System ----
1. City Management
2. Distance Management
3. Display vehcles
4. Delivery Request Handling
5. Cost, Time, and Fuel Calculations
6. Delivery Records
7. Finding The Least-Cost Route (Least-Distance)
8. Performance Reports
9. File Handling
10. Exit
Choose option: 4

Cities List:
[0] colombo
[1] maharagama

--- Vehicle Types ---
1) Van   | Capacity: 1000 kg | Rate: 30 LKR/km | Speed: 60 km/h | Fuel: 12 km/l
2) Truck | Capacity: 5000 kg | Rate: 40 LKR/km | Speed: 50 km/h | Fuel: 6 km/l
3) Lorry | Capacity: 10000 kg | Rate: 80 LKR/km | Speed: 45 km/h | Fuel: 4 km/l
Enter source city index: 0
Enter destination city index: 1
Enter weight (kg): 234

Select Vehicle Type:
Enter vehicle type: 1

Delivery Request Accepted
Source: colombo
Destination: maharagama
Weight: 234.00 kg
Vehicle: 1
```

5) Cost, Time, and Fuel Calculations

```
---- Logistics Management System ----
1. City Management
2. Distance Management
3. Display vehcles
4. Delivery Request Handling
5. Cost, Time, and Fuel Calculations
6. Delivery Records
7. Finding The Least-Cost Route (Least-Distance)
8. Performance Reports
9. File Handling
10. Exit
Choose option: 5

Cities List:
[0] colombo
[1] maharagama
Enter begining city index: 0
Enter End city index: 1
Enter weight (kg): 345

Vehicle Types:

--- Vehicle Types ---
1) Van   | Capacity: 1000 kg | Rate: 30 LKR/km | Speed: 60 km/h | Fuel: 12 km/l
2) Truck | Capacity: 5000 kg | Rate: 40 LKR/km | Speed: 50 km/h | Fuel: 6 km/l
3) Lorry | Capacity: 10000 kg | Rate: 80 LKR/km | Speed: 45 km/h | Fuel: 4 km/l
Enter vehicle type: 2

Delivery Cost Summary
Start: colombo
End: maharagama
Distance: 34.00 km
Weight: 345.00 kg
Base Delivery Cost: 1406.92 LKR
Fuel Used: 5.67 L
Fuel Cost: 1756.67 LKR
Operational Cost: 3163.59 LKR
Profit (25%): 351.73 LKR
 Final Customer Charge: 3515.32 LKR
```

6) Delivery Records

```
---- Logistics Management System ----
1. City Management
2. Distance Management
3. Display vehcles
4. Delivery Request Handling
5. Cost, Time, and Fuel Calculations
6. Delivery Records
7. Finding The Least-Cost Route (Least-Distance)
8. Performance Reports
9. File Handling
10. Exit
Choose option: 6

Cities List:
[0] colombo
[1] maharagama
Enter start city index: 0
Enter End city index: 1
Enter weight (kg): 234

--- Vehicle Types ---
1) Van   | Capacity: 1000 kg | Rate: 30 LKR/km | Speed: 60 km/h | Fuel: 12 km/l
2) Truck | Capacity: 5000 kg | Rate: 40 LKR/km | Speed: 50 km/h | Fuel: 6 km/l
3) Lorry | Capacity: 10000 kg | Rate: 80 LKR/km | Speed: 45 km/h | Fuel: 4 km/l

Select Vehicle Type:
Enter vehicle type: 2
 Delivery Recorded
Record No: 1
```

7) Finding The Least-Cost Route (Least-Distance)

```
---- Logistics Management System ----
1. City Management
2. Distance Management
3. Display vehcles
4. Delivery Request Handling
5. Cost, Time, and Fuel Calculations
6. Delivery Records
7. Finding The Least-Cost Route (Least-Distance)
8. Performance Reports
9. File Handling
10. Exit
Choose option: 7

Cities List:
[0] colombo
[1] maharagama
Enter START city index: 0
Enter END city index: 1
Enter Weight (kg): 23

--- Vehicle Types ---
1) Van   | Capacity: 1000 kg | Rate: 30 LKR/km | Speed: 60 km/h | Fuel: 12 km/l
2) Truck | Capacity: 5000 kg | Rate: 40 LKR/km | Speed: 50 km/h | Fuel: 6 km/l
3) Lorry | Capacity: 10000 kg | Rate: 80 LKR/km | Speed: 45 km/h | Fuel: 4 km/l
Select Vehicle:  2
DELIVERY COST ESTIMATION
------------------------------------------------------
From: colombo
To: maharagama
Minimum Distance: 9999.00 km
Vehicle: 2
Weight: 23.00 kg
------------------------------------------------------
Base Cost: 400879.91 LKR
Fuel Used: 1666.50 L
Fuel Cost: 516615.00 LKR
Operational Cost: 917494.88 LKR
Profit: 100219.98 LKR
Customer Charge: 1017714.88 LKR
Estimated Time: 199.98 hours
======================================================

Best Route:
```

Only used two cities that’s why there is no best route

```
10. Exit
Choose option: 7

Cities List:
[0] colombo
[1] maharagama
[2] kollupitiya
[3] rathnapura
Enter START city index: 2
Enter END city index: 3
Enter Weight (kg): 56

--- Vehicle Types ---
1) Van   | Capacity: 1000 kg | Rate: 30 LKR/km | Speed: 60 km/h | Fuel: 12 km/l
2) Truck | Capacity: 5000 kg | Rate: 40 LKR/km | Speed: 50 km/h | Fuel: 6 km/l
3) Lorry | Capacity: 10000 kg | Rate: 80 LKR/km | Speed: 45 km/h | Fuel: 4 km/l
Select Vehicle:  2
DELIVERY COST ESTIMATION
------------------------------------------------------
From: kollupitiya
To: rathnapura
Minimum Distance: 98.00 km
Vehicle: 2
Weight: 56.00 kg
------------------------------------------------------
Base Cost: 3941.95 LKR
Fuel Used: 16.33 L
Fuel Cost: 5063.33 LKR
Operational Cost: 9005.29 LKR
Profit: 985.49 LKR
Customer Charge: 9990.77 LKR
Estimated Time: 1.96 hours
======================================================

Best Route: kollupitiya - maharagama - colombo - rathnapura
```

8) Performance Reports

```
---- Logistics Management System ----
1. City Management
2. Distance Management
3. Display vehcles
4. Delivery Request Handling
5. Cost, Time, and Fuel Calculations
6. Delivery Records
7. Finding The Least-Cost Route (Least-Distance)
8. Performance Reports
9. File Handling
10. Exit
Choose option: 8

-----------PERFORMANCE REPORTS--------------
Total Deliveries Completed: 1
Total Distance Covered: 19.00 km
Average Delivery Time: 0.32 hours
Total Revenue: 1206.54 LKR
Total Profit: 143.14 LKR

Longest Route: 19.00 km (colombo - maharagama)
Shortest Route: 19.00 km (colombo - maharagama)
```

9)File handling

```
---- Logistics Management System ----
1. City Management
2. Distance Management
3. Display vehcles
4. Delivery Request Handling
5. Cost, Time, and Fuel Calculations
6. Delivery Records
7. Finding The Least-Cost Route (Least-Distance)
8. Performance Reports
9. File Handling
10. Exit
Choose option: 9
Routes Saved
Deliveries Saved
```

void removeNewline(char *str);

Purpose:
Removes the unwanted newline (`'\n'`) character at the end of strings read using `fgets()`.

How it works:
Checks the last character of the string and replaces newline with a string terminator (`'\0'`).

void addCity(char city[][SIZE], int *citycount);

Purpose:
Allows the user to add a new city name to the city list.

Inputs:
User types a city name.

Updates:
Increases `cityCount`.

void displayCities(char city[][SIZE], int citycount);

Purpose:
Prints all saved cities with their index numbers for easy selection.

void renameCity(char city[][SIZE], int citycount);

Purpose:
Allows the user to modify an existing city name.

void removeCity(char city[][SIZE], int *citycount);

Purpose:
Deletes a selected city from the list and shifts remaining elements.

void cityManagement(char city[][SIZE], int *citycount);

Purpose:
Main menu for city operations: add / rename / remove / display.

int distance(int dist[][CITIES], int cityCount, int a, int b, int km);

Purpose:
Stores the distance (in km) between two specific cities (symmetric update).

Returns:
0 on error, 1 on success.

void inputDistance(int dist[][CITIES], char cities[][SIZE], int cityCount);

Purpose:
Allows the user to enter and update the **distance matrix** values.

void displayDistanceTable(int dist[][CITIES], char cities[][SIZE], int cityCount);

Purpose:
Displays the full distance table in a formatted matrix.

void displayVehicleTypes();

Purpose:
Shows the available vehicle categories and their parameters.

(Used as reference during delivery request)

void handleDeliveryRequest(int dist[][CITIES], char cities[][SIZE], int cityCount, int capacity[]);

Purpose:
Collects user input for a delivery order: source, destination, weight and vehicle type.
Validates: city selection, weight, distance availability.

void calculateDeliveryCost(int dist[][CITIES], char cities[][SIZE], int cityCount,

int capacity[], int ratePerKm[], int avgSpeed[], int fuelEff[], float fuelPrice);`
Purpose:
Calculates and displays:

Base cost
Fuel consumption & cost
Total operational cost
Profit
Final customer charge
Estimated delivery time

void storeDeliveryRecord(int dist[][CITIES], char cities[][SIZE], int citycount,

int deliveryStart[], int deliveryEnd[], float deliveryWeight[],
int deliveryVehicle[], int *deliveryCount, int capacity[]);`
Purpose:
Stores each completed delivery into delivery record arrays for future analysis/ reports.

Ensures maximum deliveries ≤ 50

void findLeastCostRoute(int dist[][CITIES], char cities[][SIZE], int citycount,

int capacity[], int ratePerKm[], int avgSpeed[], int fuelEff[], float fuelPrice);`

Purpose:

Finds the **minimum distance route** (direct or indirect) using **brute-force search**.
Then calculates cost & time for the selected shortest path.

Helps find cheaper alternative delivery routes
Displays detailed estimation report

void showReports(int deliveryCount, int deliveryStart[], int deliveryEnd[],

float deliveryWeight[], int deliveryVehicle[], int dist[][CITIES],
int capacity[], int ratePerKm[], int avgSpeed[],
int fuelEff[], float fuelprice, char cities[][SIZE]);`

Purpose:

Shows performance statistics:

Total deliveries
Total distance travelled
Longest & shortest completed routes

void loadSaveData(int mode, char cities[][SIZE], int *cityCount, int dist[][CITIES],

int deliveryStart[], int deliveryEnd[], float deliveryWeight[],
int deliveryVehicle[], int *deliverycount);`

Purpose:

Handles **file saving & loading**:

Mode	Function

0	Load `routes.txt` + `deliveries.txt` at startup
1	Save city, distance & delivery data before exit