
Logistics and Management System

Name: Dasin Madanayake

Student ID: AS20240566

Course: University of Sri Jayawardanapura BSc Physical Science

Submission Date: 10/21/2025

Git hub Repository link:

<https://github.com/dinsaradasin/AS20240566-Logistics-Management-System-.git>

1. Introduction

The Logistics Management System is designed to automate and simplify the process of managing deliveries, cities, and vehicles. It allows managing city data, calculating distances, handling delivery requests, calculating delivery costs and fuel usage, and generating performance reports for efficient management. The system supports up to 30 cities and 50 delivery records and it is coded using the C programming language using arrays, pointers, functions and file handling.

2. Objectives

The main objectives of the Logistics Management System are:

1. Storing and managing city names.
2. Storing and managing distances between cities efficiently in a software system
3. Storing and managing data of vehicles and their properties like capacity and fuel consumption
4. Handle delivery requests based on starting and destination cities, cargo weight, and vehicle selection.
5. Calculating of delivery costs, operational costs, fuel consumption, profit, and final customer charges.

6. Storing and maintaining delivery records for future reference.
 7. Generate performance reports.
-

3. System Design

3.1 Data Structures

- **Cities:**

char cities[int MAX_CITIES][50];

- **Distance Matrix:**

float distance[MAX_CITIES][MAX_CITIES]

- **Vehicles:**

Vehicles are represented as arrays:

- int capacity[3] = {1000, 5000, 10000};
- float ratePerKm[3] = {30, 40, 80} ;
- float avgSpeed[3] = {60, 50, 45} ;
- float fuelEff[3] = {12, 6, 4} ;
- char types[3][20] = {"Van", "Truck", "Lorry"};

- **Deliveries:**

Multiple arrays store delivery details:

- Int deliverySrc[MAX_DEL]
- Int deliveryDes[MAX_DEL]
- Float deliveryWei[MAX_DEL]
- int deliveryVehicle[MAX_DEL]
- float deliveryCost, float deliveryTime, float fuelUsed, float fuelCost, float profit, float customerCharge

3.2 Functional Modules

1. City Management

- Add, rename, remove, and display cities.

2. Distance Management

- Enter, update, and display distances between city pairs.

3. Vehicle Management

- Displays all vehicles with their capacity, rate per km, average speed, and fuel efficiency.

4. Delivery Requests

- Users select source and destination cities, vehicle type, and cargo weight.
- Validates weight and vehicle data.

5. Cost, Time, and Fuel Calculation

For each delivery, these are calculated:

- **Delivery Cost:** $\text{distance} \times \text{rate} \times (1 + \text{weight} / 10000)$
- **Delivery Time:** $\text{distance} / \text{average speed}$
- **Fuel Used:** $\text{distance} / \text{fuel efficiency}$
- **Fuel Cost:** $\text{fuel used} \times \text{fuel price}$
- **Operational Cost:** $\text{delivery cost} + \text{fuel cost}$
- **Profit:** 25% of delivery cost
- **Customer Charge:** $\text{operational cost} + \text{profit}$

6. Delivery Records

- Display detailed information about each delivery

7. Performance Report

- Display a summary of total deliveries, total distance, average delivery time, total revenue, and total profit.

8. File Handling functions

- `saveRoutes` and `loadRoutes` store and retrieve city and distance data.
- `saveDeliveries` and `loadDeliveries` store and retrieve delivery records.

4. Algorithm Design

4.1 City Management

- Can add, rename and remove cities.

4.2 Distance Management

- Can enter and update distances between cities.
- Display formatted distance table.

4.3 Delivery Request Handling

1. User enters source and destination cities.
2. Then, enter the cargo weight and vehicle type.
3. Check weight limitations.
4. Validate city indices.
5. Record valid delivery into arrays.

4.4 Cost, Time, and Fuel Calculations

- Loop all deliveries and compute cost, time, fuel, profit, and final charge.

4.5 Minimum Distance Algorithm

- Calculates all possible routes and selects the one with the minimum distance (supports only for 2 to 4 cities).

5. File Handling

5.1 Routes File (routes.txt)

- Stores city names and the distance between cities.

5.2 Deliveries File (deliveries.txt)

- Stores all delivery records.

6. Sample Outputs

➤ cityManage Function

```
Data Loaded Successfully!

=====

logistics management system

=====

1.City Management
2.Distance Management between Cities
3.Vehicle Management
4.Delivery Request Handling
5.Cost,Time and Fuel Calculations
6.Delivery Records
7.Least Cost Route
8.Report
9.Exit

Enter the choice:1

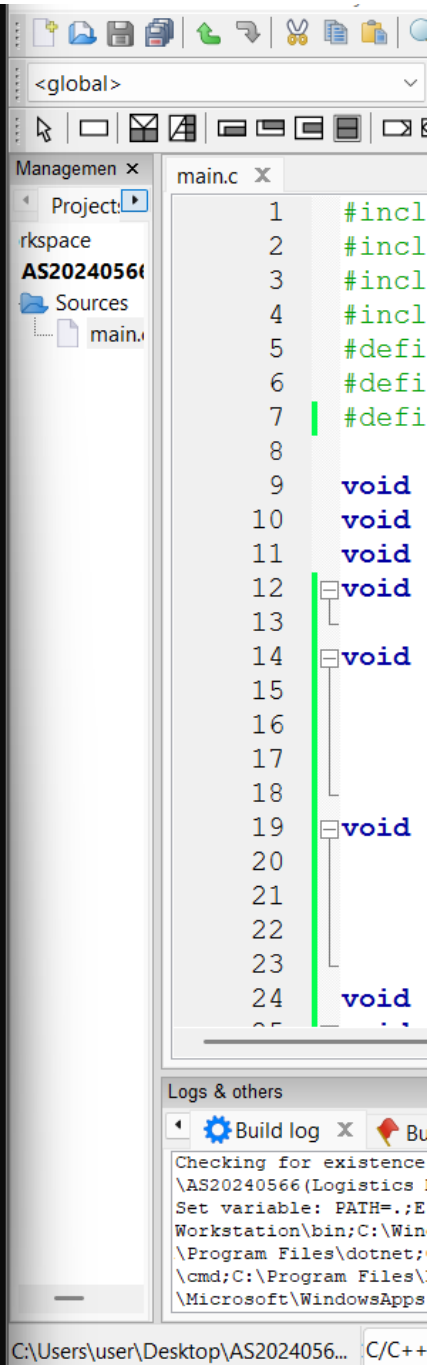
1.Add a new city
2.Rename a city
3.Remove a city
4.Display Cities
5.Exit

Enter the option:1
Enter the name of city 1:Horana

1.Add a new city
2.Rename a city
3.Remove a city
4.Display Cities
5.Exit

Enter the option:1
Enter the name of city 2:Colombo

1.Add a new city
2.Rename a city
```



The screenshot displays a C++ IDE with two main panels. The left panel shows the output of the 'cityManage' function, which is a menu-driven system for logistics management. The user has entered choice 1, then option 1, and entered 'Horana' as the name of city 1. The menu is repeated, and the user has entered option 1 again and 'Colombo' as the name of city 2. The right panel shows the 'main.c' source code, which includes preprocessor directives for file inclusion and definition, and function declarations for 'void' functions. The 'Build log' window at the bottom right shows the compilation process, including checking for existence and setting the PATH variable.

```
1 #incl
2 #incl
3 #incl
4 #incl
5 #defi
6 #defi
7 #defi
8
9 void
10 void
11 void
12 void
13
14 void
15
16
17
18
19 void
20
21
22
23
24 void
25
```

Build log X Bu

```
Checking for existence
\AS20240566(Logistics !
Set variable: PATH=.;E
Workstation\bin;C:\Win
\Program Files\dotnet;
\cmd;C:\Program Files\
\Microsoft\WindowsApps
```

C:\Users\user\Desktop\AS2024056... C/C++

```
Enter the option:1
Enter the name of city 3:Negam
```

- 1.Add a new city
- 2.Rename a city
- 3.Remove a city
- 4.Display Cities
- 5.Exit

```
Enter the option:2
Enter the current name of the city:Negam
Enter the new name of city Negam:Negambo
City renamed successfully!
```

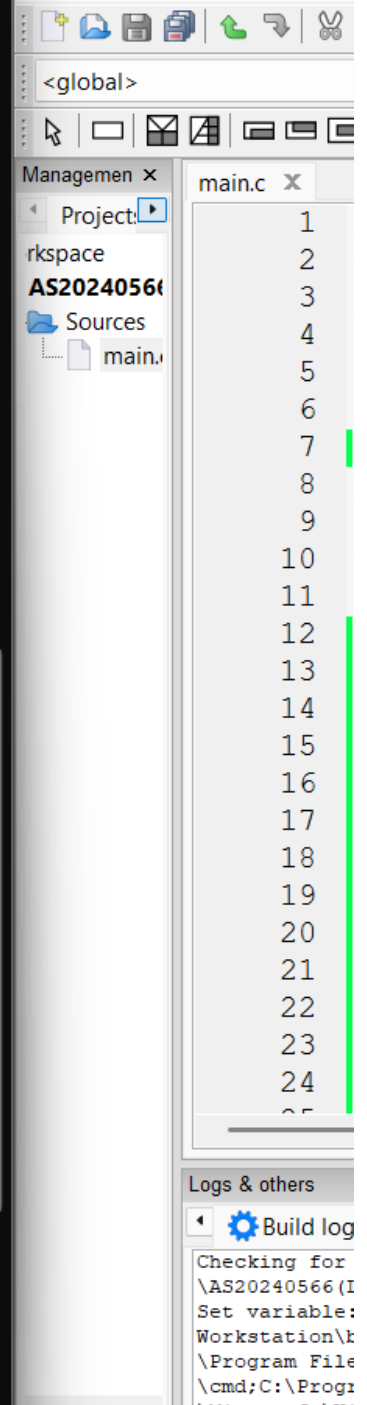
- 1.Add a new city
- 2.Rename a city
- 3.Remove a city
- 4.Display Cities
- 5.Exit

```
Enter the option:3
Enter the current name of the city:Horana
City removed successfully
```

- ```
1.Add a new city
2.Rename a city
3.Remove a city
4.Display Cities
5.Exit
```

```
Enter the option:4
City 1:Colombo

City 2:Negambo
```



## ➤ distanceManage Function

```

Enter the choice:2
1.Enter the Distance between cities
2.Update the distance between cities
3.Display the distances
4.Exit

Enter the option:1
Enter the Distance between city Colombo and Negambo:15
Enter the Distance between city Colombo and Horana:25
Enter the Distance between city Colombo and Kandy:150
Enter the Distance between city Negambo and Horana:30
Enter the Distance between city Negambo and Kandy:130
Enter the Distance between city Horana and Kandy:100
1.Enter the Distance between cities
2.Update the distance between cities
3.Display the distances
4.Exit

Enter the option:2
Enter the name city 1 :Colombo
Enter the name city 2 :Horana
Enter the distance between cities Colombo and Horana:19
1.Enter the Distance between cities
2.Update the distance between cities
3.Display the distances
4.Exit

Enter the option:3

Distance Table

City Colombo Negambo Horana Kandy

Colombo 0.00 15.00 19.00 150.00
Negambo 15.00 0.00 30.00 130.00
Horana 19.00 30.00 0.00 100.00
Kandy 150.00 130.00 100.00 0.00

```

AS20240566 (Logistics System) (bin\Debug\AS20240566 (Logistics System).exe)  
Set variable: PATH=C:\CodeBlocks\MinGW\bin

## ➤ vehicleManage Function

```

=====
1.City Management
2.Distance Management between Cities
3.Vehicle Management
4.Delivery Request Handling
5.Cost,Time and Fuel Calculations
6.Delivery Records
7.Least Cost Route
8.Report
9.Exit

Enter the choice:3

|-----|
| Type | Capacity(kg) | Rate per km(LKR) | Avg Speed(km/h) | Fuel Efficiency(km/L) |
|-----|
| Van | 1000 | 30.00 | 60.00 | 12.00 |
|-----|
| Truck | 5000 | 40.00 | 50.00 | 6.00 |
|-----|
| Lorry | 10000 | 80.00 | 45.00 | 4.00 |
|-----|
=====
1.City Management

```



➤ **deliveryRequest Function**

```
Enter the choice:4

Cities
1. Colombo
2. Negambo
3. Horana
4. Kandy
Enter the source city index:1
Enter the destination city index:3
Enter the weight of the cargo:50000

1.Van
2.Truck
3.Lorry

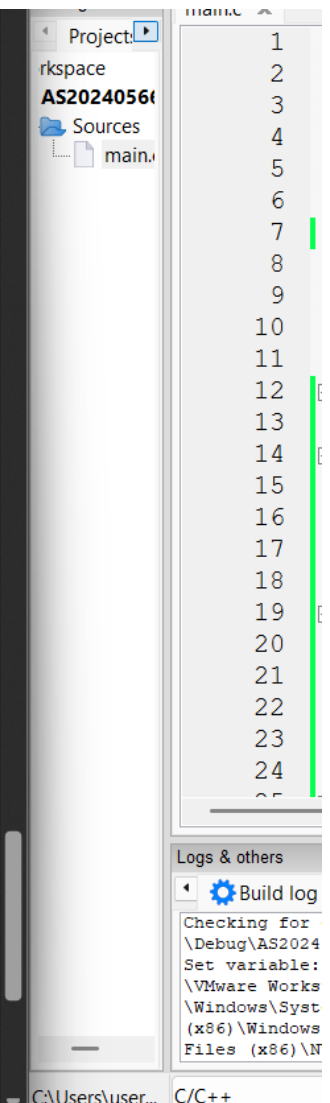
Enter the vehicle Type:3
Weight capacity of the lorry is 10000kg

To exit the loop enter 10
Enter the source city index:1
Enter the destination city index:3
Enter the weight of the cargo:1000

1.Van
2.Truck
3.Lorry

Enter the vehicle Type:1

Delivery Request Accepted!
From: Colombo
To: Horana
Weight: 1000.00 kg
Vehicle: Van
```



The screenshot shows a C++ IDE with a dark-themed console window on the left and a right-hand sidebar. The console displays the execution of a program that prompts for city choices, source/destination indices, cargo weight, and vehicle type. It shows two iterations of the program, with the first iteration using a lorry and the second using a van. The right sidebar contains a 'Project' view showing a file named 'main.cpp' and a 'Sources' folder. Below that is a 'Logs & others' panel with a 'Build log' section. The build log shows the following output: 'Checking for', '\Debug\AS2024', 'Set variable:', '\VMware Works', '\Windows\Syst', '(x86)\Windows', 'Files (x86)\N'. The status bar at the bottom indicates the file path 'C:\Users\user...' and the file type 'C/C++'.

➤ calcCTF Function

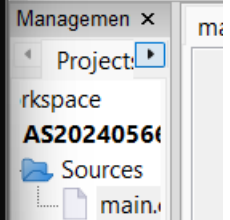
```
=====
1.City Management
2.Distance Management between Cities
3.Vehicle Management
4.Delivery Request Handling
5.Cost,Time and Fuel Calculations
6.Delivery Records
7.Least Cost Route
8.Report
9.Exit

Enter the choice:5

Cities
1.Colombo
2.Negambo
3.Horana
4.Kandy
Delivery 1 sucessfully Calculated

Delivery 2 sucessfully Calculated

Delivery 3 sucessfully Calculated
```



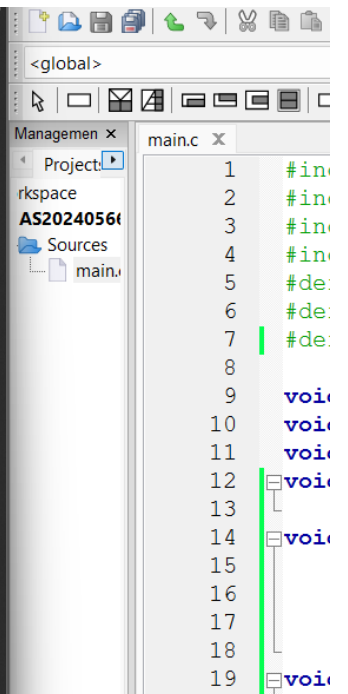
➤ **deliveryRecords Function**

```
Enter the choice:6
Delivery 1 sucessfully Calculated

Delivery 2 sucessfully Calculated

Delivery 3 sucessfully Calculated

=====
DELIVERY COST ESTIMATION
=====
From: Colombo
To: Horana
Minimum Distance is from Colombo to Horana:19.00
Vehicle: Van
Weight: 1000.00 kg
=====
Base Cost: 19.00 x 30.00(1+1000.00/10000)=627.00 LKR
Fuel Used: 1.58 liters
Fuel Cost: 490.83 LKR
Operational Cost: 1117.83 LKR
Profit: 156.75 LKR
Customer Charge: 1274.58 LKR
Estimated Time: 0.32 hours
=====
```



```
<global>
Management x main.c x
Project
rkspace
AS2024056
Sources
main.c
1 #include <stdio.h>
2 #include <math.h>
3 #include <string.h>
4 #include <stdlib.h>
5 #define MAX_DISTANCE 10000
6 #define MAX_WEIGHT 10000
7 #define MAX_FUEL 10000
8
9 void main()
10 {
11 void menu();
12 void calculateCost();
13 void displayCost();
14 void calculateTime();
15 void calculateProfit();
16 void calculateCustomerCharge();
17 void calculateEstimatedTime();
18
19 void menu();
```

=====

DELIVERY COST ESTIMATION

=====

From: Horana  
To: Kandy  
Minimum Distance is from Horana to Kandy:100.00  
Vehicle: Truck  
Weight: 5000.00 kg

-----

Base Cost:  $100.00 \times 40.00(1+5000.00/10000)=6000.00$  LKR  
Fuel Used: 16.67 liters  
Fuel Cost: 5166.67 LKR  
Operational Cost: 11166.67 LKR  
Profit: 1500.00 LKR  
Customer Charge: 12666.67 LKR  
Estimated Time: 2.00 hours

=====

=====

DELIVERY COST ESTIMATION

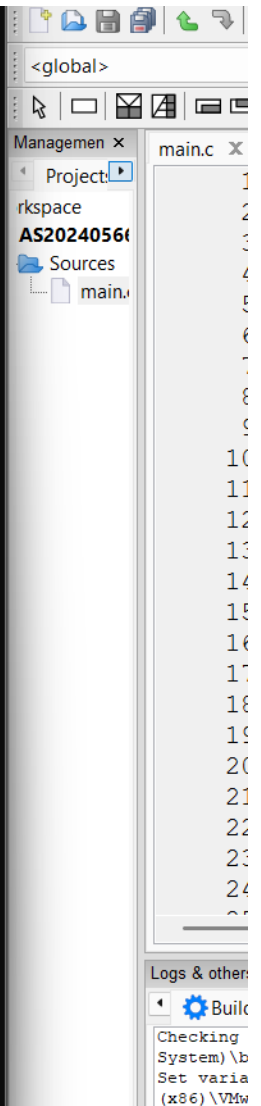
=====

From: Colombo  
To: Kandy  
Minimum Distance is from Colombo to Horana to Kandy:119.00  
Vehicle: Lorry  
Weight: 10000.00 kg

-----

Base Cost:  $150.00 \times 80.00(1+10000.00/10000)=24000.00$  LKR  
Fuel Used: 37.50 liters  
Fuel Cost: 11625.00 LKR  
Operational Cost: 35625.00 LKR  
Profit: 6000.00 LKR  
Customer Charge: 41625.00 LKR  
Estimated Time: 3.33 hours

=====



### ➤ leastDistance Function

```
Enter the choice:7
Enter the source city index:1
Enter the destination city index:2
Minimum Distance is from Colombo to Negambo:15.00

=====
1.City Management
2.Distance Management between Cities
3.Vehicle Management
4.Delivery Request Handling
5.Cost,Time and Fuel Calculations
6.Delivery Records
7.Least Cost Route
8.Report
9.Exit

Enter the choice:7
Enter the source city index:1
Enter the destination city index:4
Minimum Distance is from Colombo to Horana to Kandy:119.00

=====
```

### ➤ performanceReport Function

```
=====
1.City Management
2.Distance Management between Cities
3.Vehicle Management
4.Delivery Request Handling
5.Cost,Time and Fuel Calculations
6.Delivery Records
7.Least Cost Route
8.Report
9.Exit

Enter the choice:8

===== PERFORMANCE REPORT =====
Total Deliveries Completed : 3
Total Distance Covered : 269.00 km
Average Delivery Time : 1.88 hours
Total Revenue : 55566.25 LKR
Total Profit : 7656.75 LKR
Longest Route Completed : Colombo to Kandy (150.00 km)
Shortest Route Completed : Colombo to Horana (19.00 km)
=====
```

➤ **File Handling Functions**

▪ **Data Load**

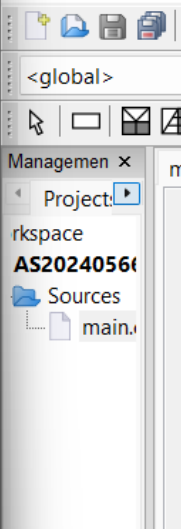
```
Data Loaded Successfully!

=====

logistics management system

=====

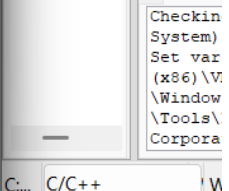
1.City Management
2.Distance Management between Cities
3.Vehicle Management
4.Delivery Request Handling
5.Cost,Time and Fuel Calculations
6.Delivery Records
7.Least Cost Route
8.Report
9.Exit
```



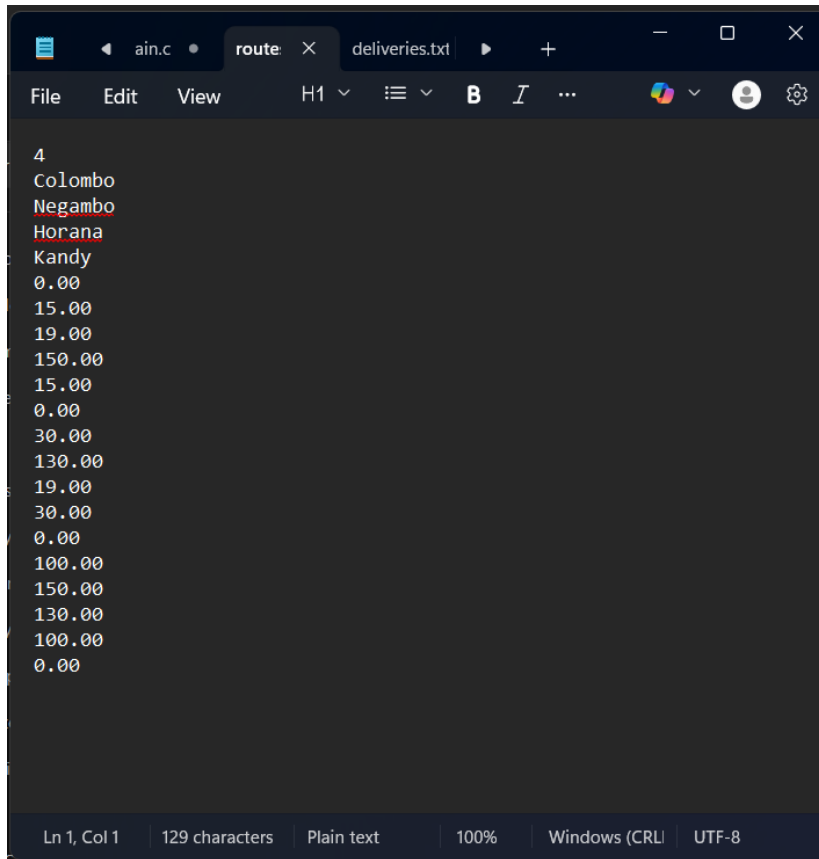
▪ **Data Saved**

```
Enter the choice:9
Data Saved Successfully!

Process returned 0 (0x0) execution time : 1255.267 s
Press any key to continue.
|
```



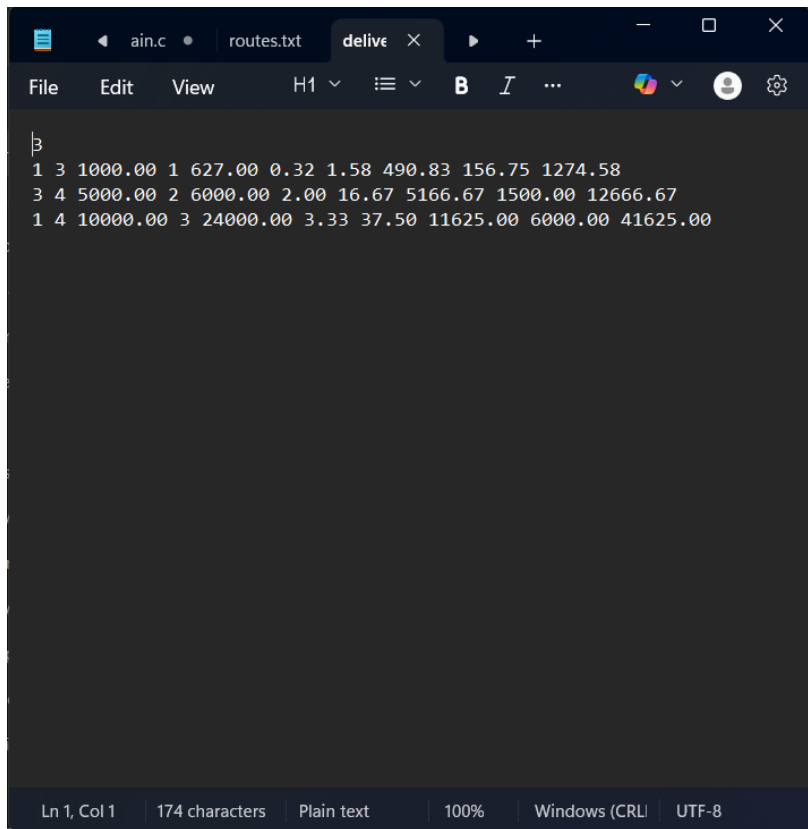
▪ route.txt



The image shows a code editor window with two tabs: 'ain.c' and 'route.txt'. The 'route.txt' tab is active, displaying a list of locations and associated numerical values. The locations are Colombo, Negambo, Horana, and Kandy. Each location is followed by a series of numerical values, some of which are formatted with two decimal places. The editor's interface includes a menu bar with 'File', 'Edit', and 'View', a toolbar with various icons, and a status bar at the bottom indicating the current line and column (Ln 1, Col 1), the total number of characters (129), the text encoding (UTF-8), and the window title (route.txt).

```
4
Colombo
Negambo
Horana
Kandy
0.00
15.00
19.00
150.00
15.00
0.00
30.00
130.00
19.00
30.00
0.00
100.00
150.00
130.00
100.00
0.00
```

▪ **deliveries.txt**



A screenshot of a code editor window with a dark theme. The window has three tabs: 'ain.c', 'routes.txt', and 'delive'. The 'delive' tab is active. The editor shows three lines of text in a monospaced font. The status bar at the bottom indicates 'Ln 1, Col 1', '174 characters', 'Plain text', '100%', 'Windows (CRLF)', and 'UTF-8'.

```
1 3 1000.00 1 627.00 0.32 1.58 490.83 156.75 1274.58
3 4 5000.00 2 6000.00 2.00 16.67 5166.67 1500.00 12666.67
1 4 10000.00 3 24000.00 3.33 37.50 11625.00 6000.00 41625.00
```

---

## 7. Limitations

- Maximum of 30 cities and 50 deliveries.
- Minimum distance calculation supports only on 2 to 4 cities.
- Vehicle types and rates are fixed.



---

## 8. Conclusion

The Logistics Management System is a program for managing cargo deliveries between different cities. It accurately calculates delivery costs, fuel consumption, and profits while maintaining detailed delivery records. It gives a final performance report of the overall deliveries and helps to save the data in .txt files. The system's modular design and file persistence make it extendable for future enhancements.