

CSC 1012 Introduction to Computer Programming

Assignment (Individual)

Logistics Management System

Index: AS20240515

Name: E.A.S.K.B.Edirisinghe

Github : UserName: KethmikaEdirisinghe

Link: <https://github.com/KethmikaEdirisinghe/Logistics-Management-System.git>

Introduction

This project implements a **menu-driven Logistics Management System** using the **C programming language** as a part of CSC 1012. It applies key programming concepts such as arrays, loops, conditionals, and functions to manage cities, delivery routes, and vehicle details in the system. The system calculates delivery time, cost, and fuel usage while maintaining a record of completed deliveries. It can also generate performance reports showing total deliveries, distances covered, and the shortest and longest routes.

- Objectives:

- To design a simple logistics and delivery management system using the C language.
- To apply programming concepts such as arrays, loops, conditionals, and functions.
- To manage cities and record distances between them.
- To handle customer delivery requests and vehicle selections.
- To calculate delivery time, cost, and fuel consumption automatically.
- To store and retrieve delivery data using file handling.
- To generate performance reports showing delivery summaries and route details.

- **Technologies Used :**
 - > Programming Language – C
 - > IDE – CodeBlocks
- **Github folder contains :**
- Codefiles:
 - > main.c
 - > cityManagement.c
 - > distanceManagement.c
 - > vehicleManagement.c
 - > deliveryRequestHandeling.c
 - > calculations.c
 - > deliveryRecords.c
 - > fileHandeling.c
 - > findingBestRoot.c
 - > InterfaceMainMenu.c
 - > performance.c
- README.md
- Project Report

- **System Features**

1. **City Management** – Add, rename and remove cities from system according to the user's perspective.
2. **Distance Management** – Enter, edit and view distances between cities.
3. **Vehicle Management** – Select vehicles with different capacities, speeds, and fuel efficiencies.
4. **Delivery Request Handling** – Place delivery orders by selecting source, destination, and package weight.
5. **Automatic Calculations** – The system automatically calculates delivery cost, fuel cost, time, and profit.
6. **Performance Reports** – Displays total deliveries, total distance covered, longest and shortest routes, and average delivery time.
7. **File Handling** – Saves all route and delivery data into text files and loads them back when the program restarts.

- Main Structural Functions

Function	Task	Location
cityManagement()	Controls adding, renaming, and removing cities.	cityManagement.c
distanceManagement()	Manages distances between cities (add, edit, and view).	distanceManagement.c
vehicleManagement()	Displays available vehicles and allows selection for deliveries.	vehicleManagement.c
diliveryRequestSection()	Handles user input for delivery requests such as cities and package weight.	deliveryRequest.c
Calculations()	Calculates delivery cost, fuel consumption, time, total cost, and profit.	calculations.c
recordDilivery()	Saves each delivery's data (source, destination, cost, etc.) into the records array.	deliveryRecords.c
performances()	Generates performance reports including totals, averages, and routes.	performance.c
saveRoutesToFile()	Saves city names and distance matrix to routes.txt.	fileHandling.c
loadRoutesFromFile()	Loads city and distance data from the routes.txt file.	fileHandling.c
loadDiliveriesFromFile()	Loads previous delivery records from the deliveries.txt file.	fileHandling.c
saveDiliveriesToFile()	Saves all delivery records and costs to deliveries.txt.	fileHandling.c
findShortestRouteIterative()	Finds the shortest (least-cost) route between two cities using an iterative search.	findingBestRoute.c

- **Calculations**

D = Distance (km)

R = Rate per km (LKR)

W = Weight (kg)

S = Speed (km/h)

Fuel Used = Fuel consumption (liters)

Fuel price (310 LKR per liter)

E = Fuel Efficiency (km/l)

Cost = $D \times R \times (1 + W \times 1/10000)$

Estimated Delivery Time (hours) = D / S

Fuel Consumption = D / E

Fuel Cost (LKR) = FuelUsed \times F

Total Operational Cost (LKR) = DeliveryCost + FuelCost

Profit Calculation (LKR) = Cost \times 0.25 (25% markup on base delivery cost)

Final Charge to Customer (LKR) = TotalCost + Profit

• System Manual Guide

Main Menu:

```
"C:\Users\ASUS\OneDrive\Do x + v
Routes loaded from routes.txt
Deliveries loaded from deliveries.txt

=====
LOGISTIC MANAGEMENT SYSTEM

1. Go to City management.
2. Go to Distance management.
3. Place a Delivery.
4. Check Records.
5. Exit from System.

=====
(*)Select what you want to do:
```

City Management Section:

```
"C:\Users\ASUS\OneDrive\Do x + v

=====
LOGISTIC MANAGEMENT SYSTEM

1. Go to City management.
2. Go to Distance management.
3. Place a Delivery.
4. Check Records.
5. Exit from System.

=====
(*)Select what you want to do: 1

-----City Management Section-----

1. Add City.
2. Rename City.
3. Remove City.
4. Return to Main Menu

-----
:->Select what you want to do: 1

-----Add cities-----
Enter the name of the city: Ambalangoda
Do you want to add another City?(Y/N): Y
```

```
"C:\Users\ASUS\OneDrive\Do x + v

-----City Management Section-----

1. Add City.
2. Rename City.
3. Remove City.
4. Return to Main Menu

-----
:->Select what you want to do: 1

-----Add cities-----
Enter the name of the city: Ambalangoda
Do you want to add another City?(Y/N): Y
Enter the name of the city: Jaffna
Do you want to add another City?(Y/N): N

-----City Management Section-----

1. Add City.
2. Rename City.
3. Remove City.
4. Return to Main Menu

-----
:->Select what you want to do: |
```


Distance Management:

```
"C:\Users\ASUS\OneDrive\Do x + v
5. Exit from System.
=====
(*)Select what you want to do: 2
-----
---Distance Management Section---
1: Add Distances.
2: Edit Distances.
3: View Distances.
4: Exit to Main Menu.
-----
:->Select the option: 1
-----
Add Distances
Distance from Ambalangoda to Jaffna (km): 479
-----
---Distance Management Section---
1: Add Distances.
2: Edit Distances.
3: View Distances.
4: Exit to Main Menu.
-----
:->Select the option:
```

Delivery Requesting:

```
"C:\Users\ASUS\OneDrive\Do x + v
4. Check Records.
5. Exit from System.
=====
(*)Select what you want to do: 3
-----
---Delivery Requesting Section---
1. Ambalangoda
2. Jaffna
|*|Enter where you put your package: 1
|*|Enter where you want to send your package: 2
|*|Your Package will be transferred from Ambalangoda -----> Jaffna
|*|Enter the weight of the package(kg): 150
-----
---Vehicle Management Section---
-----
Type Capacity(kg) Rate per km(LKR) Avg Speed(kmh) Fuel Efficiency(km/L)
Van 1000 20 60 12
```

Vehicle Management Section:

```
"C:\Users\ASUS\OneDrive\Do x + v
|*|Your Package will be transfered from Ambalangoda -----(to)----> Jaffna
|*|Enter the weight of the package(kg): 150

-----
---Vehicle Management Section---
-----
Type Capacity(kg) Rate per km(LKR) Avg Speed(kmh) Fuel Efficiency(km/L)
Van 1000 20 60 12
Truck 5000 40 50 6
Lorry 10000 80 45 4
-----

-*****-
1-> Van.
2-> Truck.
3-> Lorry.

-*****-

=>Please Select your Vehicle: 1
* You have selected Van.

-----
-----
```

Performance and Reports:

```
"C:\Users\ASUS\OneDrive\Do x + v
=====
(*)Select what you want to do: 4
=====
DELIVERY PERFORMANCE REPORT
=====
Total Deliveries Completed : 1
Total Distance Covered : 479.00 km
Average Delivery Time : 7.98 hours
Total Revenue : 27622.33 LKR
Total Profit : 5524.47 LKR
=====
Longest and Shortest Routes Completed:
Shortest Route:
From Ambalangoda -> To Jaffna : 479.00 km
Longest Route:
From Ambalangoda -> To Jaffna : 479.00 km
=====
*** Best Route Optimization (Based on City Network) ***
--- Least-Cost Route Found ---
Minimum Distance: 479.00 km
Optimal Route: Ambalangoda -> Jaffna
=====
=====
```

Quit from System:

```
=====
LOGISTIC MANAGEMENT SYSTEM
1. Go to City management.
2. Go to Distance management.
3. Place a delivery.
4. Check Records.
5. Exit from System.
=====
(*)Select what you want to do: 5
-----

Thank You for Using our system.
...Exit Completed...

***
-----
Routes saved to routes.txt.
Diliveries save to dilivery.txt
***
-----

Process returned 0 (0x0)   execution time : 0.609 s
Press any key to continue.
```

Reference:

- geek for geeks
 - <https://www.geeksforgeeks.org/c/basics-file-handling-c/>
- Tutorials point
 - https://www.tutorialspoint.com/cprogramming/c_pointers.html
- w3schools.com
 - https://www.w3schools.com/c/c_strings.php
- Youtube : CSE GURU : Travelling Salesman Problem using(TSP) Brute Force Approach | Lec77 | Design & Analysis of Algorithm
 - <https://www.youtube.com/watch?v=gsYmZdZuFCw&t=118s>
- ChatGTP /Gemini support for Exhaustive Search algorithm,file handeling tips