***Logistic-Management-system-Project Report***

GitHub Repository Link: <https://github.com/Sathirasugeesvara/Logistic-Management-System-.git>

* Name: Sathira Sugeesvara
* Index no: AS20240401
* University of Sri Jayewardhanapura - Faculty of Applied Science
* Course: CSC 1012 Introduction to Computer Programing
* Language: C
* Date: 26th of October 2025

**Introduction**

The purpose of the assignment is to design and implement a menu-driven logistics management system using the C programming language. Apply key programming concepts such as arrays, functions, loops, conditionals, to simulate a simple logistics and delivery management system that manages cities, delivery routes, vehicle types, fuel consumption, and order costs.

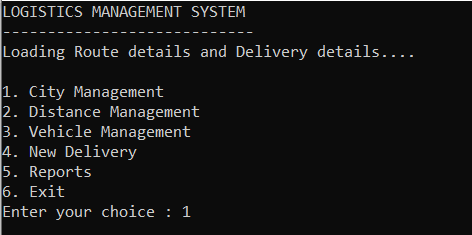
**Objectives**

* Manage cities and distances between them
* Handle customer delivery requests
* Estimate delivery time and cost
* Track completed deliveries and generate performance reports.

**describe all the functions**

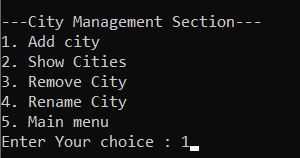
* main functions
* void mainMenuShow();

Show main menu of the Logistic Management System.



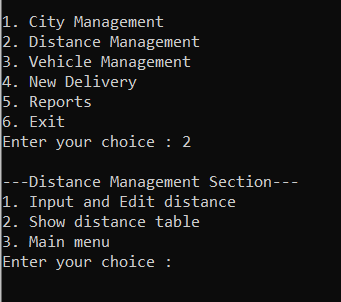
* void toHandleCities();

Main menu to add, rename, remove, and view cities.



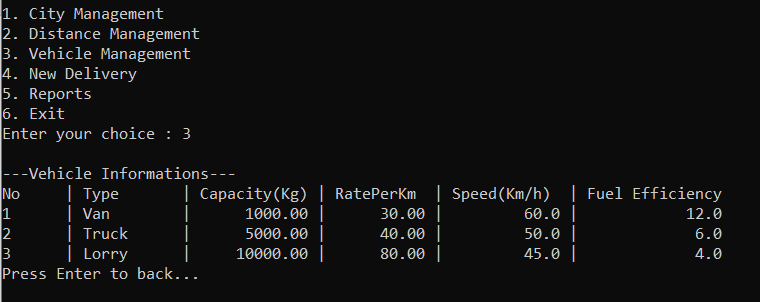
* void handleDistance();

Check are there have cities enter and show distance menu add and edit distance, show distance.



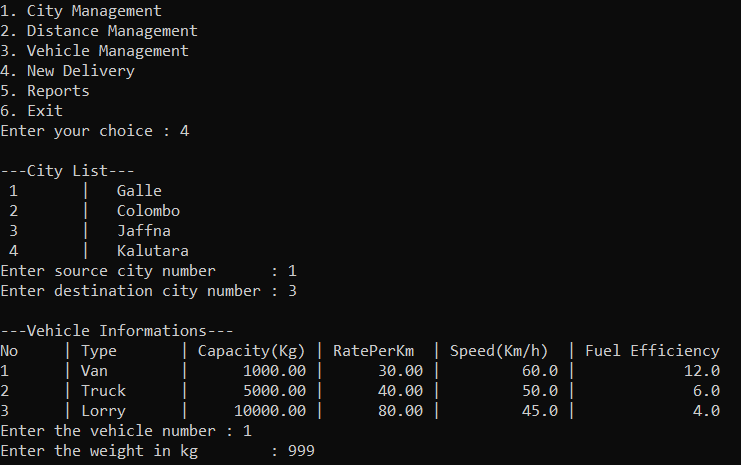
* void showVehicles();

Show vehicle information specified in the given project instructions and use special feature press enter to back.



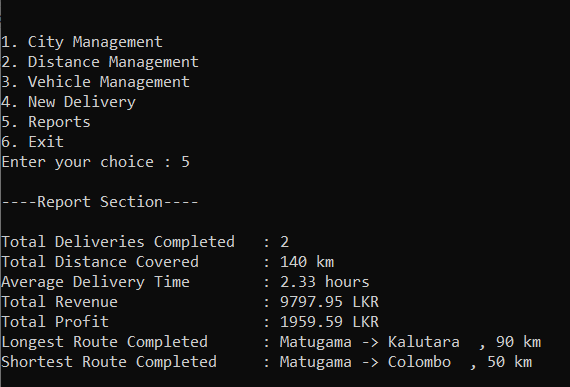
* void handleDelivery();

Show cities and ask get to source city index and destination city index. Then show vehicle list and choose one among them. After that get capacity you want to deliver. Then check is it over the limit.



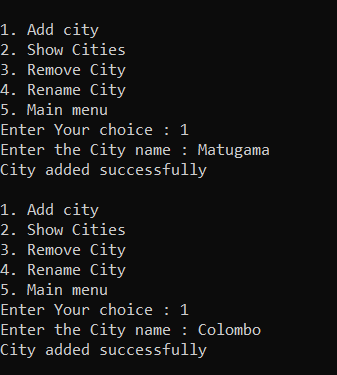
* void showReport();

Check are there have any deliveries and print Total Deliveries Completed, Total Distance Covered, Average Delivery Time, Total Revenue, Total Profit , Longest Route Completed , Shortest Route Completed.



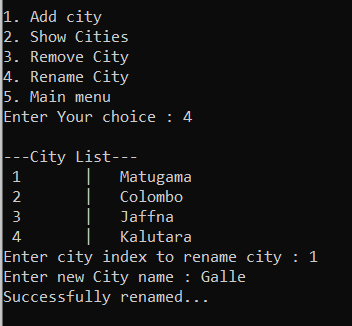
* for city management function
* void addCity();

Check the new city is available in the array if not available add the city name into the cityName [][] array. Else show city is available.



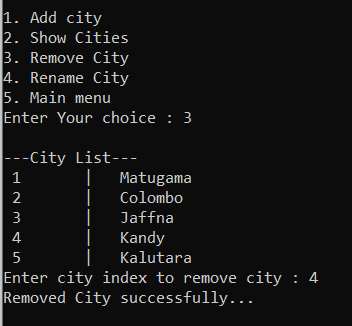
* void renameCity();

Check the index is it available and change the city's name if the new name isn't already taken.



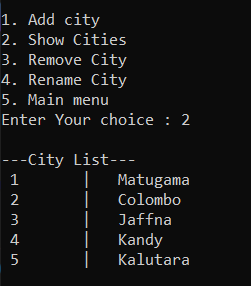
* void removeCity();

Check the index city is available and Deletes a city and shifts array , distance matrix.



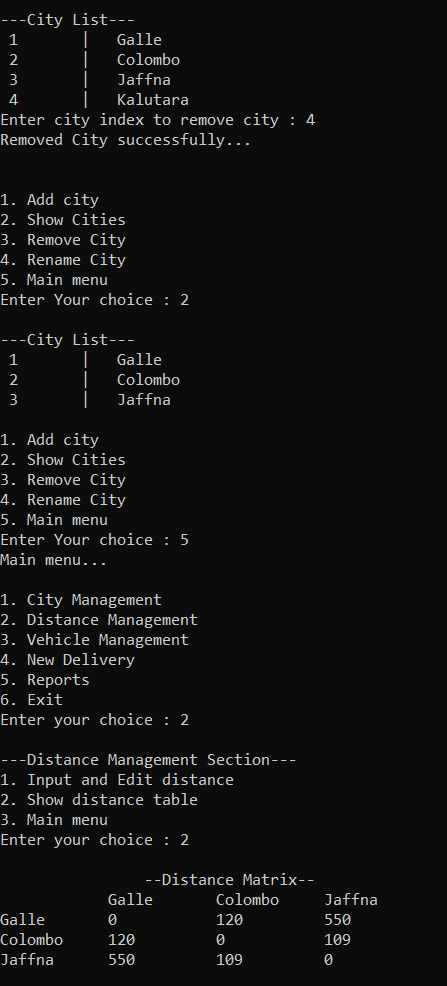
* void showCity();

Check cities are available and Displays all currently added cities with their numbers.



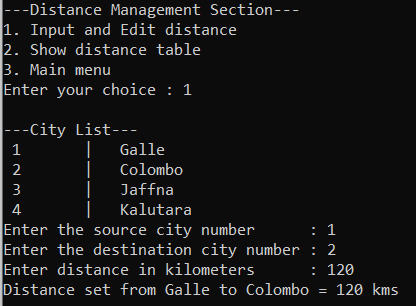
* void RemoveDistncsWhenCityRemove(int index);

Reorganizes the distance matrix after a city is removed , moves remaining distances up/left



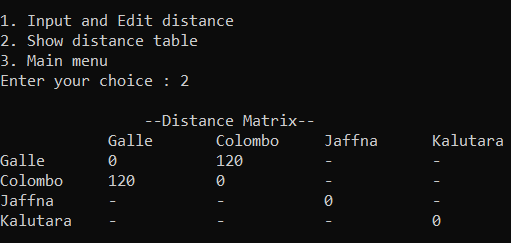
* for distance management function
* void inputAndEditDis();

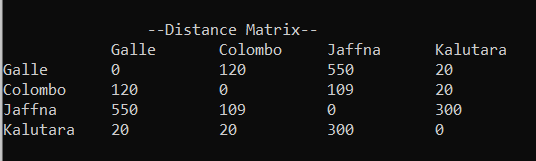
Check are there have any cities you entered and then add or edit distance



* void displyDisMatrx();

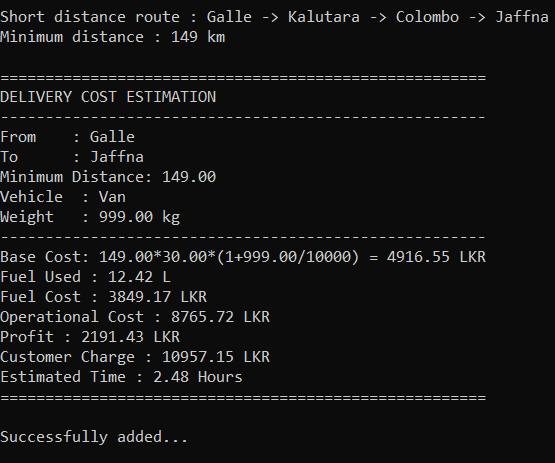
You entered data show like matrix here





* for handle delivery function
* void calculAndMiniStat(int sourceIndex,int destiIndex,float weight,int vehiType,int distance,int shortDis[],int shortLen);

Calculate cost, fuel used, fuel cost, total cost, profit, time. Then print a short report.

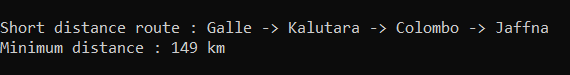


* int findLeastCostRoute(int sourceIndex,int destiIndex,int shortDis[],int \*shortLen);

Find least cost route between you entered city and assign it to distance

* void printShortDis(int shortDis[],int shortLen);

Print least cost route using arrows.

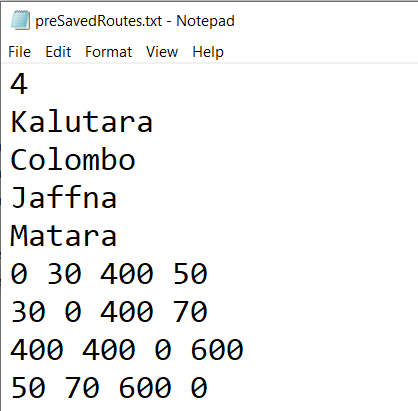


* int totalDistanceCovered(int way[],int n);

Find and store the sum of distance for find total distance and show in main report.

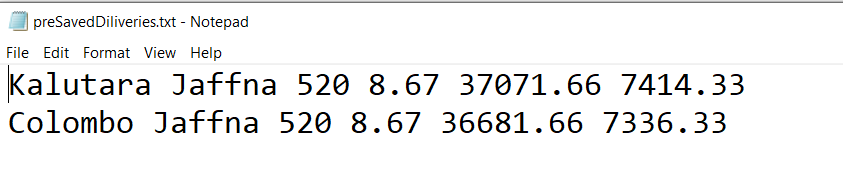
* for file handling
* void saveWaysToFile(const char \*filename);

Saves the current list of routes (ways between cities) to a file specified by filename. It usually appends or overwrites the file with route data.



* void saveDeliveriesToFile(const char \*filename);

Saves all delivery records (source, destination, distance, time, cost, profit) to the given file. Ensures the delivery data is persisted for later use.



* void getWaysFromFile(const char \*filename);

Reads route data from the specified file and loads it into the program’s memory, allowing the program to use previously saved routes

* void getDeliveriesFromFile(const char \*filename);

Reads delivery records from the file and populates the program’s arrays or structures, restoring saved delivery information for processing.

**Define**

* #define MAX\_CITI 30

Store Maximumly 30 cities in the array.

* #define MAX\_NAME\_SIZE 20

The maximum city name size (20 characters can store each city name)

* #define MAX\_DELI 50

The maximum deliveries 50 can store the array. Can’t add more than 50 deliveries.

* #define FUEL\_PRIZ 310.0f

The 310 was used for fuel price as it was specified in the given project instructions.

* #define INF 1000000000

This used for represent a very large value , like infinity.

**References**

* W3Schools C Programming Tutorial – <https://www.w3schools.com/c>
* GeeksforGeeks: File Handling in C – <https://www.geeksforgeeks.org/file-handling-c/>
* University of Sri Jayewardenepura – CSC1031 Course Materials