Test Plan

Nelson Villatoro

CMSC115

Chapter 8, Project 5

December 03, 2024

**Program Goals & Objectives**

This program finds the central city among user-specified cities using their coordinates on a two-dimensional plane. It verifies the user inputs at least two cities and guides them to enter each city’s x and y coordinates. The program computes the Euclidean distance between each pair of cities and finds the city with the minimum total distance. It presents this central city’s coordinates and total distance, helping the user understand the most optimally located city.

**Program Functional Requirements**

1. The program must prompt and validate the number of cities (must be at least two).
2. The program must prompt and validate the x,y coordinates for each city, ensuring numeric input.
3. The program must compute Euclidean distances between each city and every other city.
4. The program must identify the city with the smallest sum of distances to all other cities.
5. The program must display the coordinates of the central city.
6. Lastly, the program must display the total distance from the central city to all other cities.

**Program Pseudocode**

START

Display introduction and explain what the program does

Prompt the user to enter the number of cities

While the number of cities is less than 2

Display an error message

Prompt the user again until a valid number of cities (>=2) is entered

Create a structure to store coordinates for all cities

For each city

Prompt for the x-coordinate until a valid number is entered

Prompt for the y-coordinate until a valid number is entered

Store these coordinates

Set a variable for the central city index and set shortest total distance to a very large number

For each city as a candidate

Calculate the total distance from this city to all other cities

If this total distance is less than the current shortest total distance

Update the central city index with this city’s index

Update the shortest total distance with this city’s total distance

Display the central city’s coordinates

Display the shortest total distance

END

**Program Flowchart**

**A diagram of a flowchart

Description automatically generated**

**Table 1 – Traceability Matrix**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case | Input/Output | Expected Result | Actual Result | Outcome  (Pass/Fail) |
| 1a | ===============================================  Central City Calculator  ===============================================  This program finds the central city among a set of cities.  The central city has the shortest total distance to all other cities.  Enter the number of cities (must be at least 2): 1 | Error: You must enter at least 2 cities. | Error: You must enter at least 2 cities. | Pass |
| 2a | ===============================================  Central City Calculator  ===============================================  This program finds the central city among a set of cities.  The central city has the shortest total distance to all other cities.  Enter the number of cities (must be at least 2): 3  For each city, you will need to enter two numbers:  1. The x-coordinate (horizontal position)  2. The y-coordinate (vertical position)  Example:  For a city at position x=3.5, y=-2.0  When prompted, enter: 3.5 [Enter] -2.0 [Enter]  City #1:  Enter the x-coordinate: 4  Enter the y-coordinate: 9  City #2:  Enter the x-coordinate: -100  Enter the y-coordinate: 67  City #3:  Enter the x-coordinate: 45  Enter the y-coordinate: 88 | Results:  The central city (city #1) is at (4.0, 9.0)  The total distance to all other cities is 208.09 units | Results:  The central city (city #1) is at (4.0, 9.0)  The total distance to all other cities is 208.09 units | Pass |
| 3a | ===============================================  Central City Calculator  ===============================================  This program finds the central city among a set of cities.  The central city has the shortest total distance to all other cities.  Enter the number of cities (must be at least 2): four | Invalid input! Please enter a number. | Invalid input! Please enter a number. | Pass |