NIRMA UNIVERSITY

INSTITUTE OF TECHNOLOGY

Project : Package Delivery Optimization

Aditya Jain :-

- Roll no : 24ITK058
- Course : B.Tech CSE + MBA
- Semester : 1

Jiya Patel :-

- Roll no : 24ITK075
- Course : B.Tech CSE
- Semester : 1

Submission Date :-

Table of content

INTRODUCTION

Brief Overview:

The primary goal of this project is to find the shortest route for delivering packages across multiple cities. The solution ensures efficient travel, minimizes the total distance covered, and visits all designated cities exactly once before returning to the starting point. It also tells total distance needed to travel and at what location the refuel of tank is needed.

Objectives

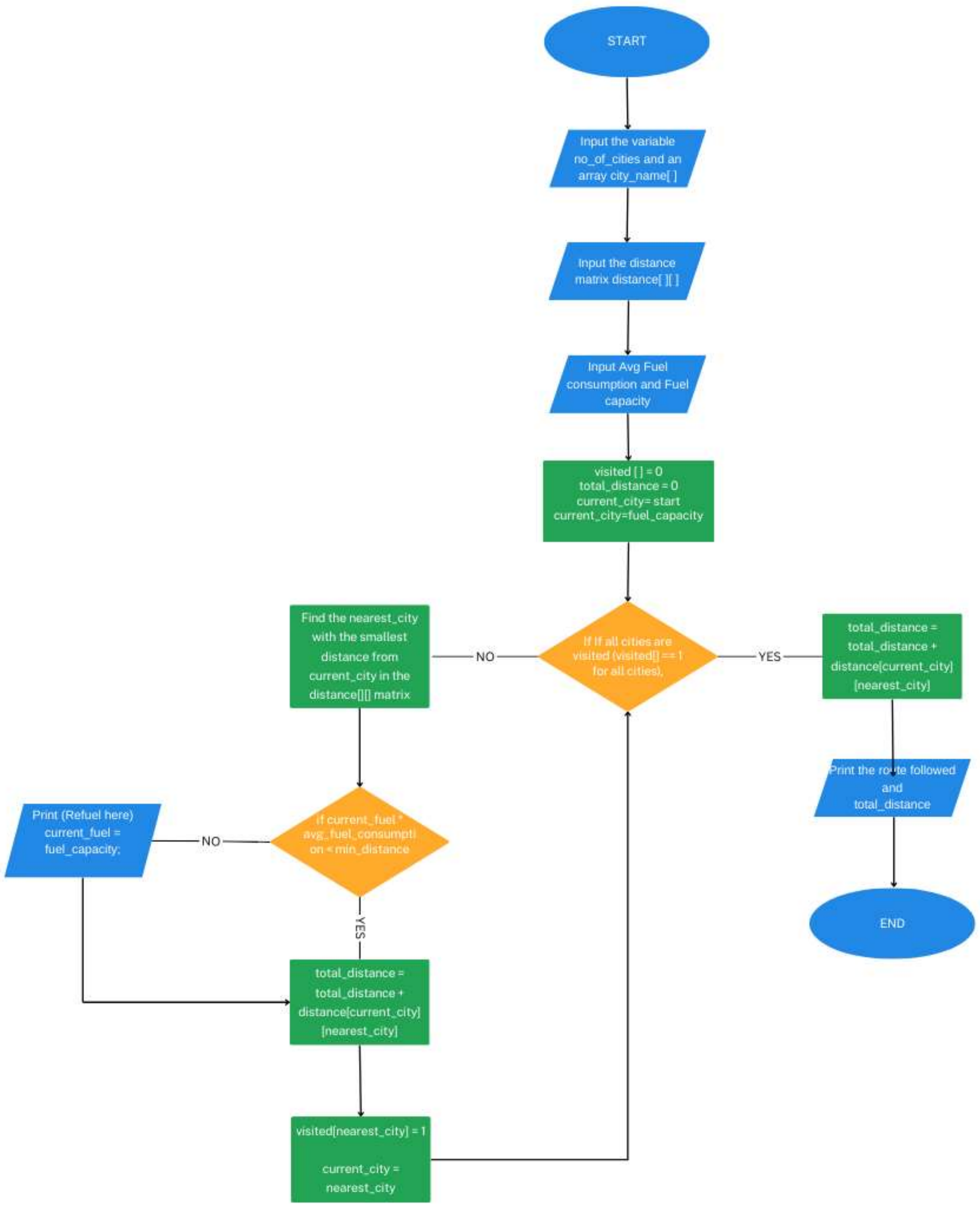
- Develop an efficient method to calculate the shortest distance between up to 10 cities or locations and advise for refueling at respective places.
- Minimize time and fuel usage while ensuring all cities are visited.
- Provide a user-friendly program that accepts:
    - **Input**: Number of cities, their names, the distance matrix, fuel capacity and average fuel consumption.
    - **Output**: The optimal route covering all places, the total distance and in which city refuel is required.
- Enhance the understanding of real-world optimization problems and demonstrate their solutions using the C programming language.

ALGORITHMS

1. **Start**
    - Initialize constants MAX = 10, distance matrix distance[][], city names city_name[], and fuel variables (avg_fuel_consumption, fuel_capacity).
2. **Input Number of Cities**
    - Take the number of cities (no_of_cities) and their names (city_name[]).
3. **Input Distance Matrix and fuel data**
    - Enter the symmetric distances between cities and store them in distance[][].
    - Input avg_fuel_consumption (distance per unit of fuel) and fuel_capacity.
4. **Input starting city**
    - Input the starting city.
5. **Initialize Variables**
    - Initialize visited[], total_distance = 0, current_city to the starting city, and calculate fuel range = avg_fuel_consumption * fuel_capacity.
6. **Find Nearest Unvisited City**
    - Loop through the cities to find the unvisited city with the shortest distance from current_city.
7. **Update visited[]**
    - Update the nearest unvisited city as vistied.

8. **Check Fuel Availability**
    - If the fuel is insufficient to reach the nearest city, refuel (update current_fuel = fuel_capacity after refueling).
9. **Update Travel Data**
    - Add the distance to total_distance, reduce fuel based on distance traveled, and mark the city as visited.
10. **Return to Starting City**
    - After visiting all cities, check fuel for the return trip, refuel if needed, and add the distance back to total_distance.
11. **Output Results**
    - Display the route taken and the total distance traveled.

FLOWCHART


START
Input the variable no_of_cities and an array city_name[ ]
Input the distance matrix distance[ ][ ]
Input Avg Fuel consumption and Fuel capacity
visited [ ] = 0
total_distance = 0
current_city= start
current_city=fuel_capacity
If If all cities are visited (visited[] == 1 for all cities),
NO
YES
Find the nearest_city with the smallest distance from current_city in the distance[][] matrix
total_distance = total_distance + distance[current_city] [nearest_city]
Print the route followed and total_distance
END
if current_fuel * avg_fuel_consumption < min_distance
NO
YES
Print (Refuel here) current_fuel = fuel_capacity;
total_distance = total_distance + distance[current_city] [nearest_city]
visited[nearest_city] = 1
current_city = nearest_city

IMPLEMENTATION

Code

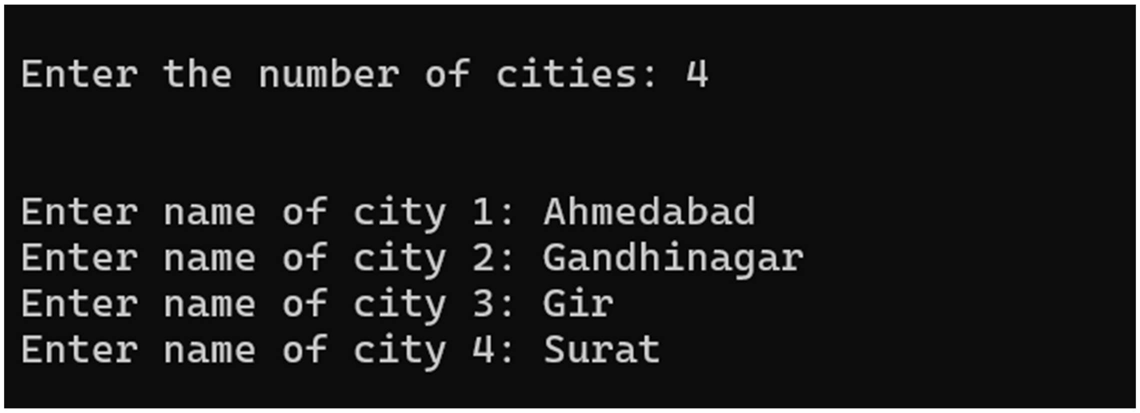
```
#include <stdio.h>
#include <string.h>

#define MAX 10

// Global variables
int distance[MAX][MAX] = {0};
int no_of_cities;
char city_name[MAX][50];
double avg_fuel_consumption;  // Distance per unit of fuel
double fuel_capacity;         // Total capacity of the fuel tank

void input(void);
void find_shortest_route(int start);

int main() {
   printf("----- Welcome to Package Delivery Optimization -----\n\n");
   printf("\t\t\t by Aditya Jain, Jiya Patel.\n\n");

   input();

   int start;
   printf("\nEnter the starting city (1 to %d): ", no_of_cities);
   scanf("%d", &start);

   find_shortest_route(start - 1);
   return 0;
```

```
}

void input() {
   printf("Enter the number of cities: ");
   scanf("%d", &no_of_cities);
   printf("\n\n");
   for (int i = 0; i < no_of_cities; i++) {
      printf("Enter name of city %d: ", i + 1);
      scanf(" %[^\n]", city_name[i]);
   }

   printf("\nEnter the distance matrix:\n");
   for (int i = 0; i < no_of_cities; i++) {
      for (int j = 0; j < no_of_cities; j++) {
         if (distance[i][j] != 0 || i == j) {
            continue;
         } else {
            printf("Enter distance between %s and %s: ", city_name[i],
city_name[j]);
            scanf("%d", &distance[i][j]);
            distance[j][i] = distance[i][j];
         }
      }
   }
   printf("\nEnter average fuel consumption (distance per unit of fuel): ");
   scanf("%lf", &avg_fuel_consumption);
   printf("Enter fuel tank capacity (in units): ");
   scanf("%lf", &fuel_capacity);
}

void find_shortest_route(int start) {
   int visited[MAX] = {0};
   int total_distance = 0;
   int current_city = start;
   double fuel_range = avg_fuel_consumption * fuel_capacity;  // Maximum
distance the vehicle can travel on a full tank
   double current_fuel = fuel_capacity;
```

```
    printf("\nThe route that should be followed is :--\n");
    for (int count = 0; count < no_of_cities; count++) {
        printf("%s", city_name[current_city]);

        // Mark the current city as visited
        visited[current_city] = 1;

        int nearest_city = -1;    // To store the index of the nearest city
        int min_distance = 10000;

        // Loop through all cities to find the nearest unvisited city
        for (int i = 0; i < no_of_cities; i++) {
            // Check if the city is unvisited and closer than the current nearest city
            if (!visited[i] && distance[current_city][i] < min_distance) {
                nearest_city = i;
                min_distance = distance[current_city][i];
            }
        }

        if (nearest_city == -1) break; // If no unvisited city, break the loop

        // Check if the vehicle can reach the nearest city with the current fuel
        if (current_fuel * avg_fuel_consumption < min_distance) {
            printf(" (Refuel here) ");
            current_fuel = fuel_capacity;  // Refuel to full capacity
        }

        total_distance += min_distance;
        current_fuel -= min_distance / avg_fuel_consumption;  // Update
remaining fuel
        printf(" --> ");
        current_city = nearest_city;
    }

    // Return to the starting city
    if (current_fuel * avg_fuel_consumption < distance[current_city][start]) {
            printf(" (Refuel here) ");
            current_fuel = fuel_capacity;  // Refuel to full capacity
        }

    printf("--> %s\n", city_name[start]);
    total_distance += distance[current_city][start];

    // Print the total distance of the route
    printf("\nTotal Distance: %d\n", total_distance);
}
```

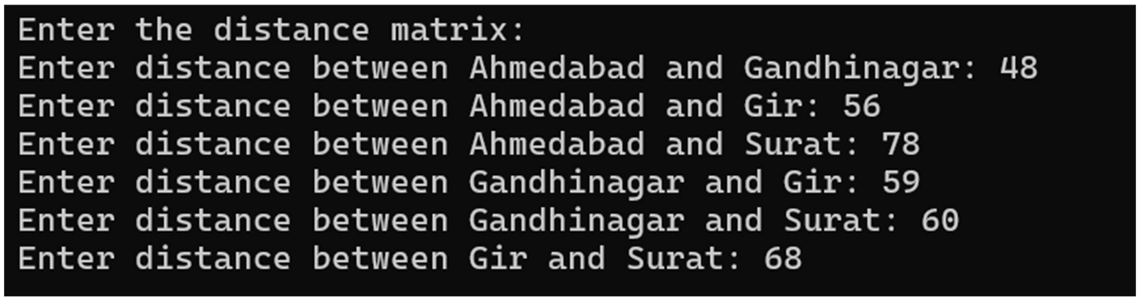
RESULTS

1. Input the Number of cities and add the name of cities.



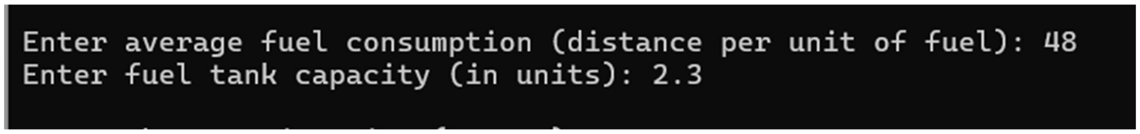
```
Enter the number of cities: 4


Enter name of city 1: Ahmedabad
Enter name of city 2: Gandhinagar
Enter name of city 3: Gir
Enter name of city 4: Surat
```

2. Adding of the Distance between the cities.



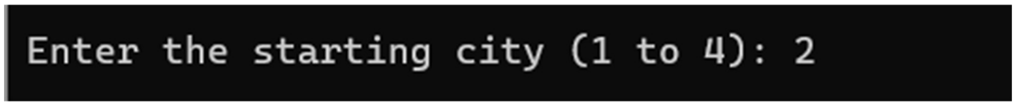
```
Enter the distance matrix:
Enter distance between Ahmedabad and Gandhinagar: 48
Enter distance between Ahmedabad and Gir: 56
Enter distance between Ahmedabad and Surat: 78
Enter distance between Gandhinagar and Gir: 59
Enter distance between Gandhinagar and Surat: 60
Enter distance between Gir and Surat: 68
```

3. Enter the Avg. Fuel consumption and Tank capacity



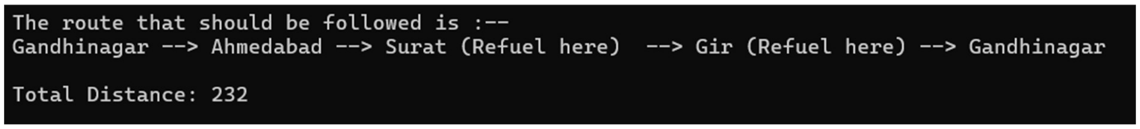
```
Enter average fuel consumption (distance per unit of fuel): 48
Enter fuel tank capacity (in units): 2.3
```

4. Enter the Starting City From you want to Start



```
Enter the starting city (1 to 4): 2
```

5. Display the shortest path and the were the refuel is needed with Total Distance



```
The route that should be followed is :--
Gandhinagar --> Ahmedabad --> Surat (Refuel here)  --> Gir (Refuel here) --> Gandhinagar

Total Distance: 232
```