**Assignment 5**

**Consider the scenario of the supply chain management system. The supply chain involves multiple locations, such as suppliers, warehouses, distribution centers, and retail stores, connected by transportation networks. Find the shortest possible route using Dijkstra's algorithm to optimize transportation path and inventory management.**

#include <iostream>

#include <string>

using namespace std;

class SupplyChain {

int map[100][100];

int total;

string names[100];

int added;

public:

SupplyChain(int n) {

total = n;

added = 0;

for (int i = 0; i < total; i++) {

for (int j = 0; j < total; j++) {

map[i][j] = 9999;

}

}

}

void addname(int index, string n) {

if (added < total) {

names[index] = n;

added++;

}

}

void showindex() {

cout << "\nLocation Indexes:\n";

for (int i = 0; i < total; i++) {

cout << i << " : " << names[i] << endl;

}

}

void addroute(int a, int b, int w) {

map[a][b] = w;

map[b][a] = w;

}

void printPath(int prev[], int j) {

if (prev[j] == -1) {

cout << names[j];

return;

}

printPath(prev, prev[j]);

cout << " -> " << names[j];

}

void shortest(int start) {

int dist[100], visited[100], prev[100];

for (int i = 0; i < total; i++) {

dist[i] = 9999;

visited[i] = 0;

prev[i] = -1;

}

dist[start] = 0;

for (int i = 0; i < total - 1; i++) {

int min = 9999, index = -1;

for (int j = 0; j < total; j++) {

if (!visited[j] && dist[j] < min) {

min = dist[j];

index = j;

}

}

if (index == -1) break;

visited[index] = 1;

for (int j = 0; j < total; j++) {

if (!visited[j] && map[index][j] != 9999 && dist[index] + map[index][j] < dist[j]) {

dist[j] = dist[index] + map[index][j];

prev[j] = index;

}

}

}

cout << "\nShortest distance and path from " << names[start] << ":\n";

for (int i = 0; i < total; i++) {

if (dist[i] == 9999)

cout << names[i] << ": Unreachable\n";

else {

cout << names[i] << " (Distance: " << dist[i] << ") Path: ";

printPath(prev, i);

cout << "\n";

}

}

}

};

int main() {

int n;

cout << "Enter number of locations: ";

cin >> n;

SupplyChain sc(n);

cout << "\nEnter names of locations (one word each):\n";

for (int i = 0; i < n; i++) {

string temp;

cin >> temp;

sc.addname(i, temp);

}

int m;

cout << "\nEnter number of routes: ";

cin >> m;

cout << "Enter route details (from\_index to\_index distance):\n";

for (int i = 0; i < m; i++) {

int a, b, w;

cin >> a >> b >> w;

// Optional validation

if (a < 0 || b < 0 || a >= n || b >= n) {

cout << "Invalid index! Please enter again.\n";

i--;

continue;

}

sc.addroute(a, b, w);

}

sc.showindex();

int start;

cout << "\nEnter source index to find shortest routes: ";

cin >> start;

if (start < 0 || start >= n) {

cout << "Invalid source index!\n";

return 1;

}

sc.shortest(start);

return 0;

}