#include<iostream>

#include<string.h>

using namespace std;

const int MAX\_CITIES = 10;

class Flight {

private:

int adjacencyMatrix[MAX\_CITIES][MAX\_CITIES];

char cityIndex[MAX\_CITIES][10];

int cityCount;

public:

Flight() {

cityCount = 0;

for (int i = 0; i < MAX\_CITIES; i++) {

strcpy(cityIndex[i], "xx");

for (int j = 0; j < MAX\_CITIES; j++) {

adjacencyMatrix[i][j] = 0;

}

}

}

void create();

void display();

bool isConnected();

};

void Flight::create() {

int si, di, wt;

char s[10], d[10], c;

do {

cout << "\n\tEnter Source City : ";

cin >> s;

cout << "\n\tEnter Destination City : ";

cin >> d;

si = -1;

di = -1;

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

if (strcmp(cityIndex[i], s) == 0) {

si = i;

break;

}

}

if (si == -1) {

strcpy(cityIndex[cityCount], s);

si = cityCount++;

}

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

if (strcmp(cityIndex[i], d) == 0) {

di = i;

break;

}

}

if (di == -1) {

strcpy(cityIndex[cityCount], d);

di = cityCount++;

}

cout << "\n\t Enter Distance From " << s << " And " << d << ": ";

cin >> wt;

adjacencyMatrix[si][di] = wt;

cout << "\n\t Do you want to add more cities.....(y/n) : ";

cin >> c;

} while (c == 'y' || c == 'Y');

}

void Flight::display() {

cout << "\n\t Displaying Adjacency Matrix :\n\t";

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

cout << "\t" << cityIndex[i];

}

cout << "\n";

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

cout << "\t" << cityIndex[i];

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

cout << "\t" << adjacencyMatrix[i][j];

}

cout << "\n";

}

}

bool Flight::isConnected() {

bool visited[MAX\_CITIES];

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

visited[i] = false;

}

// Perform DFS from the first city

visited[0] = true;

int stack[MAX\_CITIES];

int top = 0;

stack[top++] = 0;

while (top != 0) {

int current = stack[--top];

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

if (adjacencyMatrix[current][i] != 0 && !visited[i]) {

visited[i] = true;

stack[top++] = i;

}

}

}

// Check if all cities are visited

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

if (!visited[i]) {

return false;

}

}

return true;

}

int main() {

Flight f;

int n;

char c;

do {

cout << "\n\t\*\*\*\*\* Flight Main Menu \*\*\*\*\*";

cout << "\n\t1. Create \n\t2. Adjacency Matrix\n\t3. Check Connectivity\n\t4. Exit";

cout << "\n\t.....Enter your choice : ";

cin >> n;

switch (n) {

case 1:

f.create();

break;

case 2:

f.display();

break;

case 3:

if (f.isConnected()) {

cout << "\nThe graph is connected.\n";

} else {

cout << "\nThe graph is not connected.\n";

}

break;

case 4:

return 0;

}

cout << "\n\t Do you Want to Continue in Main Menu....(y/n) : ";

cin >> c;

} while (c == 'y' || c == 'Y');

return 0;

}