

**ANDREAS ANDREOU**

**LEONIDAS LEONIDA**

**LOUIZA SAVVA**

Contents

[**Program 1:** 2](#_Toc3152)

[**Program 2:** 2](#_Toc3153)

[**Program 3:** 3](#_Toc3154)

*We confirm that the work is entirely our own and that no unauthorized programs were used, or we collaborated with other people, in a way that violates academic policies.*

# **Program 1:**

**Program 2:**

**Problem 3**

This exercise models the aircraft routes in the figure as an undirected, weighted, simple graph whose vertices are countries (0–8 → UK, Germany, Poland, Austria, France, Spain, Italy, Greece, Cyprus) and whose edge weights equal the number of red grid rectangles crossed by each yellow route; the edges and weights used are: UK–Germany 3, UK–France 3, Germany–France 5, Germany–Poland 3, Germany–Austria 4, Poland–Austria 4, France–Spain 4, France–Italy 6, Spain–Italy 5, Spain–Cyprus 11, Austria–Italy 3, Austria–Greece 6, Italy–Greece 4, Greece–Cyprus 4. The C++ program builds this same graph in two representations—an adjacency matrix pinakas[v][v] initialized to 0 with add\_edge setting symmetric weights, and an adjacency list vector<pair<int,int>> graph[v] with addedge inserting both directions—then prints both via print\_array and printgraph

Pseudocode:

PROCEDURE AddEdgeMatrix(M, i, j, w):

M[i][j] ← w

M[j][i] ← w

PROCEDURE PrintMatrix(M, V, Countries):

print " " + Countries as header row

FOR i ← 0 TO V-1:

row ← Countries[i]

FOR j ← 0 TO V-1:

row ← row + " " + M[i][j]

print row

PROCEDURE AddEdgeList(Adj, i, j, w):

append (j, w) to Adj[i]

append (i, w) to Adj[j]

PROCEDURE PrintAdjLists(Adj, V, Countries):

FOR i ← 0 TO V-1:

print Countries[i] + " →"

FOR each (nbr, w) IN Adj[i]:

print " " + Countries[nbr] + " (" + w + ")"

print newline

MAIN:

V ← 9

Countries ← ["UK","Germany","Poland","Austria","France","Spain","Italy","Greece","Cyprus"]

create V×V matrix M filled with 0

create array Adj[0..V-1] of empty lists

// index map: 0:UK, 1:Germany, 2:Poland, 3:Austria, 4:France, 5:Spain, 6:Italy, 7:Greece, 8:Cyprus

Edges ← [

(0,1,3),(0,4,3),

(1,4,5),(1,2,3),(1,3,4),

(2,3,4),

(4,5,4),(4,6,6),

(5,6,5),(5,8,11),

(3,6,3),(3,7,6),

(6,7,4),

(7,8,4)

]

FOR each (a,b,w) IN Edges:

AddEdgeMatrix(M, a, b, w)

AddEdgeList(Adj, a, b, w)

print "Adjacency Matrix"

PrintMatrix(M, V, Countries)

print newline + "Adjacency Lists"

PrintAdjLists(Adj, V, Countries)

END



