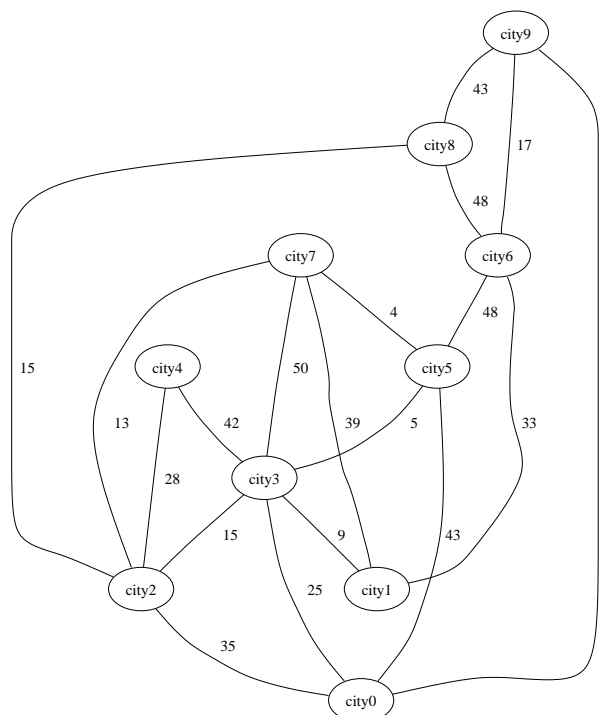


Άσκηση 3

Έστω ότι μας δίνεται ένα σύνολο από πόλεις και τα μήκη των δρόμων που συνδέουν κάποια ζευγάρια από αυτές, όπως φαίνεται στο διπλανό σχήμα. Μιλώντας με μαθηματικούς όρους, το σχήμα αυτό είναι ένας γράφος ή γράφημα (*graph*), όπου οι πόλεις είναι οι κόμβοι (*nodes*) του γράφου και οι δρόμοι είναι οι ακμές (*edges*) του. Στις ακμές του γράφου, μπορούμε να έχουμε και κάποιο κόστος (*cost*), εδώ το μήκος του κάθε δρόμου. Ένα κλασικό πρόβλημα στην Επιστήμη των Υπολογιστών, και ειδικότερα στη Θεωρητική Πληροφορική, είναι να βρεθεί μία συντομότερη διαδρομή (*shortest path*) μεταξύ δύο κόμβων ενός γράφου, δηλαδή κάποια που έχει το μικρότερο δυνατό συνολικό κόστος ακμών. Υπάρχουν διάφοροι αλγόριθμοι για την αντιμετώπιση αυτού του προβλήματος, άλλοι περισσότερο, άλλοι λιγότερο αποδοτικοί. Αντικείμενο της άσκησης αυτής είναι να υλοποιήσετε C, και μάλιστα με δύο τρόπους, έναν συγκεκριμένο αλγόριθμο εύρεσης συντομότερου μονοπατιού σε ένα γράφο, τον αλγόριθμο *Floyd-Warshall*.



Ο αλγόριθμος αυτός (http://en.wikipedia.org/wiki/Floyd-Warshall_algorithm) βρίσκει τα μήκη των συντομότερων μονοπατιών μεταξύ όλων των ζευγαριών κόμβων του δεδομένου γράφου. Έστω ότι έχουμε N κόμβους, που είναι αριθμημένοι από το 0 έως το $N-1$. Αν με $shortestPath(i, j, k)$ συμβολίσουμε το μήκος του συντομότερου μονοπατιού από τον κόμβο i προς τον κόμβο j , με ενδιάμεσους κόμβους μέσα από το σύνολο $\{0, 1, \dots, k\}$, τότε η ουσία του αλγορίθμου Floyd-Warshall βρίσκεται στην εξής αναδρομική διατύπωση:

$$shortestPath(i, j, k) = \min\{shortestPath(i, j, k - 1), shortestPath(i, k, k - 1) + shortestPath(k, j, k - 1)\}, \text{ αν } k \geq 0$$

$$shortestPath(i, j, -1) = edgeCost(i, j)$$

Η ερμηνεία του παραπάνω είναι ότι για να βρούμε το μήκος του ελάχιστου μονοπατιού από τον κόμβο i προς τον κόμβο j , με ενδιάμεσους κόμβους μέσα από το σύνολο $\{0, 1, \dots, k\}$, μπορούμε να βρούμε

- το μήκος του ελάχιστου μονοπατιού από τον κόμβο i προς τον κόμβο j , έστω d_1^{min} , με ενδιάμεσους κόμβους μέσα από το σύνολο $\{0, 1, \dots, k - 1\}$ και
- τα μήκη των ελάχιστων μονοπατιών από τον κόμβο i προς τον κόμβο k , έστω d_2^{min} , και από τον κόμβο k προς τον κόμβο j , έστω d_3^{min} , με ενδιάμεσους κόμβους μέσα από το σύνολο $\{0, 1, \dots, k - 1\}$

και να πάρουμε το ελάχιστο των d_1^{min} και $d_2^{min} + d_3^{min}$.

Με άλλα λόγια, για να βρούμε το μήκος του ελάχιστου μονοπατιού από τον κόμβο i προς τον κόμβο j , με ενδιάμεσους κόμβους μέσα από το σύνολο $\{0, 1, \dots, k\}$, είτε δεν θα χρησιμοποιήσουμε

σαν ενδιάμεσο κόμβο τον k , είτε θα τον χρησιμοποιήσουμε, αλλά τα μονοπάτια από τον i στον k και από τον k στον j δεν θα τον χρησιμοποιούν. Όποια από τις δύο εκδοχές είναι συντομότερη, αυτή δίνει το ελάχιστο μονοπάτι από τον i στο j , με κόμβους μέχρι τον k .

Φυσικά, αν $k = -1$, τότε δεν έχουμε ενδιάμεσους κόμβους, οπότε το μήκος του ελάχιστου μονοπατιού από τον κόμβο i προς τον κόμβο j είναι το μήκος της ακμής από τον i στον j .

Το μήκος του ελάχιστου μονοπατιού από οποιονδήποτε κόμβο i σε οποιονδήποτε κόμβο j σε γράφο με N κόμβους, χωρίς περιορισμό στους ενδιάμεσους κόμβους, είναι το $shortestPath(i, j, N - 1)$.

Αναδρομική υλοποίηση του αλγορίθμου Floyd-Warshall

Γράψτε ένα πρόγραμμα C το οποίο να διαβάζει από την είσοδο τα δεδομένα ενός γράφου και, χρησιμοποιώντας την αναδρομική διατύπωση του αλγορίθμου Floyd-Warshall, να υπολογίζει και να εκτυπώνει τα μήκη των ελάχιστων μονοπατιών μεταξύ όλων των ζευγαριών κόμβων στον γράφο. Όσον αφορά τη μορφή της εισόδου στο πρόγραμμα, αυτή πρέπει να είναι της μορφής:

```

<N>
<e1→0>
<e2→0> <e2→1>
<e3→0> <e3→1> <e3→2>
...
<e(N-1)→0> <e(N-1)→1> ... <e(N-1)→(N-2)>

```

Το $\langle N \rangle$ είναι το πλήθος των κόμβων του γράφου και το κάθε $\langle e_{i \rightarrow j} \rangle$ είναι το μήκος της ακμής από τον κόμβο i προς τον κόμβο j . Αν δεν υπάρχει ακμή που να συνδέει τους κόμβους i και j , τότε σαν $\langle e_{i \rightarrow j} \rangle$ να δίνεται το -1.

Αν το εκτελέσιμο πρόγραμμα που θα κατασκευάσετε τελικά ονομάζεται “flowar”, μία ενδεικτική εκτέλεσή του, για τον γράφο της πρώτης σελίδας, φαίνεται στη συνέχεια. Τα δεδομένα για τον συγκεκριμένο γράφο μπορείτε να τα βρείτε στο <http://www.di.uoa.gr/~ip/hwfiles/flowar/demo.txt>, οπότε μπορείτε να εκτελέσετε το πρόγραμμα σαν “./flowar < demo.txt”.

```

% ./flowar
10
-1
35 -1
25 9 15
-1 -1 28 42
43 -1 -1 5 -1
-1 33 -1 -1 -1 48
-1 39 13 50 -1 4 -1
-1 -1 15 -1 -1 -1 48 -1
8 -1 -1 -1 -1 -1 17 -1 43
^D
From node 1 to node 0: Length of shortest path is 34

From node 2 to node 0: Length of shortest path is 35
From node 2 to node 1: Length of shortest path is 24

From node 3 to node 0: Length of shortest path is 25
From node 3 to node 1: Length of shortest path is 9
From node 3 to node 2: Length of shortest path is 15

```

```

From node 4 to node 0: Length of shortest path is 63
From node 4 to node 1: Length of shortest path is 51
From node 4 to node 2: Length of shortest path is 28
From node 4 to node 3: Length of shortest path is 42

From node 5 to node 0: Length of shortest path is 30
From node 5 to node 1: Length of shortest path is 14
From node 5 to node 2: Length of shortest path is 17
From node 5 to node 3: Length of shortest path is 5
From node 5 to node 4: Length of shortest path is 45

From node 6 to node 0: Length of shortest path is 25
From node 6 to node 1: Length of shortest path is 33
From node 6 to node 2: Length of shortest path is 57
From node 6 to node 3: Length of shortest path is 42
From node 6 to node 4: Length of shortest path is 84
From node 6 to node 5: Length of shortest path is 47

From node 7 to node 0: Length of shortest path is 34
From node 7 to node 1: Length of shortest path is 18
From node 7 to node 2: Length of shortest path is 13
From node 7 to node 3: Length of shortest path is 9
From node 7 to node 4: Length of shortest path is 41
From node 7 to node 5: Length of shortest path is 4
From node 7 to node 6: Length of shortest path is 51

From node 8 to node 0: Length of shortest path is 50
From node 8 to node 1: Length of shortest path is 39
From node 8 to node 2: Length of shortest path is 15
From node 8 to node 3: Length of shortest path is 30
From node 8 to node 4: Length of shortest path is 43
From node 8 to node 5: Length of shortest path is 32
From node 8 to node 6: Length of shortest path is 48
From node 8 to node 7: Length of shortest path is 28

From node 9 to node 0: Length of shortest path is 8
From node 9 to node 1: Length of shortest path is 42
From node 9 to node 2: Length of shortest path is 43
From node 9 to node 3: Length of shortest path is 33
From node 9 to node 4: Length of shortest path is 71
From node 9 to node 5: Length of shortest path is 38
From node 9 to node 6: Length of shortest path is 17
From node 9 to node 7: Length of shortest path is 42
From node 9 to node 8: Length of shortest path is 43
%
```

Μπορείτε να δοκιμάσετε το πρόγραμμά σας και με άλλους γράφους, που γεννώνται τυχαία από το πρόγραμμα “randgr_<arch>”, όπου το <arch> είναι solaris, linux, windows.exe ή macosx, ανάλογα με το σύστημα που σας ενδιαφέρει. Τα εκτελέσιμα αυτού του προγράμματος για τις προη-

γούμενες αρχιτεκτονικές μπορείτε να τα βρείτε στο <http://www.di.uoa.gr/~ip/hwfiles/flowerar>. Το πρόγραμμα “randgr_<arch>” δέχεται τις εξής επιλογές:

- n <nodes> : Το πλήθος των κόμβων του γράφου είναι <nodes> (default τιμή: 10).
- w <maxCost> : Το μέγιστο κόστος ακμής είναι <maxCost> (default τιμή: 10).
- p <prob> : Η πιθανότητα ύπαρξης ακμής (από 0 έως 100) μεταξύ δύο κόμβων είναι <prob> (default τιμή: 100).
- o <outputFile> : Τα δεδομένα του γράφου αποθηκεύονται, στη μορφή που πρέπει να διαβάζονται από το πρόγραμμα “flowerar” που θα γράψετε, στο αρχείο <outputFile> (αν δεν δοθεί η επιλογή, εκτυπώνονται στην έξοδο).
- s <seed> : Το φύτρο της γεννήτριας τυχαίων αριθμών είναι <seed> (αν δεν δοθεί η επιλογή, σαν φύτρο χρησιμοποιείται ο τρέχων χρόνος).
- d <dotFile> : Τα δεδομένα του γράφου αποθηκεύονται στο αρχείο <dotFile> σε “dot” μορφή. Η μορφή αυτή μπορεί να χρησιμοποιηθεί από το πρόγραμμα Graphviz, που μπορείτε να κατεβάσετε από το <http://www.graphviz.org>, για την οπτικοποίηση γράφων που αναπαρίστανται σε “dot” μορφή, όπως ο γράφος στην πρώτη σελίδα. Περισσότερες λεπτομέρειες για τις διαδικασίες οπτικοποίησης γράφων μπορείτε να βρείτε στην τεκμηρίωση του προγράμματος Graphviz.
- v <nodesFile> : Αν δοθεί η επιλογή αυτή, σε συνδυασμό με την προηγούμενη, τα ονόματα των κόμβων για την οπτικοποίηση του γράφου παίρνονται κατά σειρά από το αρχείο <nodesFile>. Αλλιώς, σαν ονόματα χρησιμοποιούνται τα city0, city1, κλπ.

Κάποιες επιπλέον εκτελέσεις του προγράμματος “flowerar” (σε περιβάλλον Linux), με τυχαίες εισόδους που παράγονται από το πρόγραμμα “randgr_<arch>”, φαίνονται στη συνέχεια.

```
% ./randgr_linux -n 5 -p 20 -s 1 -o test.txt
% cat test.txt
5
 7
-1 -1
 6  3 -1
-1 -1 -1 10
% ./flowerar < test.txt
```

```
From node 1 to node 0: Length of shortest path is 7
```

```
From node 2 to node 0: There is no path
```

```
From node 2 to node 1: There is no path
```

```
From node 3 to node 0: Length of shortest path is 6
```

```
From node 3 to node 1: Length of shortest path is 3
```

```
From node 3 to node 2: There is no path
```

```
From node 4 to node 0: Length of shortest path is 16
```

```
From node 4 to node 1: Length of shortest path is 13
```

```
From node 4 to node 2: There is no path
```

```

From node 4 to node 3: Length of shortest path is 10
% ./randgr_linux -n 12 -s 2010 | ./flower

From node 1 to node 0: Length of shortest path is 6

From node 2 to node 0: Length of shortest path is 4
From node 2 to node 1: Length of shortest path is 2

From node 3 to node 0: Length of shortest path is 7
.....
From node 11 to node 8: Length of shortest path is 4
From node 11 to node 9: Length of shortest path is 3
From node 11 to node 10: Length of shortest path is 8
% ./randgr_linux -n 15 -s 2011 | ./flower

From node 1 to node 0: Length of shortest path is 6

From node 2 to node 0: Length of shortest path is 3
.....
From node 14 to node 13: Length of shortest path is 2
%

```

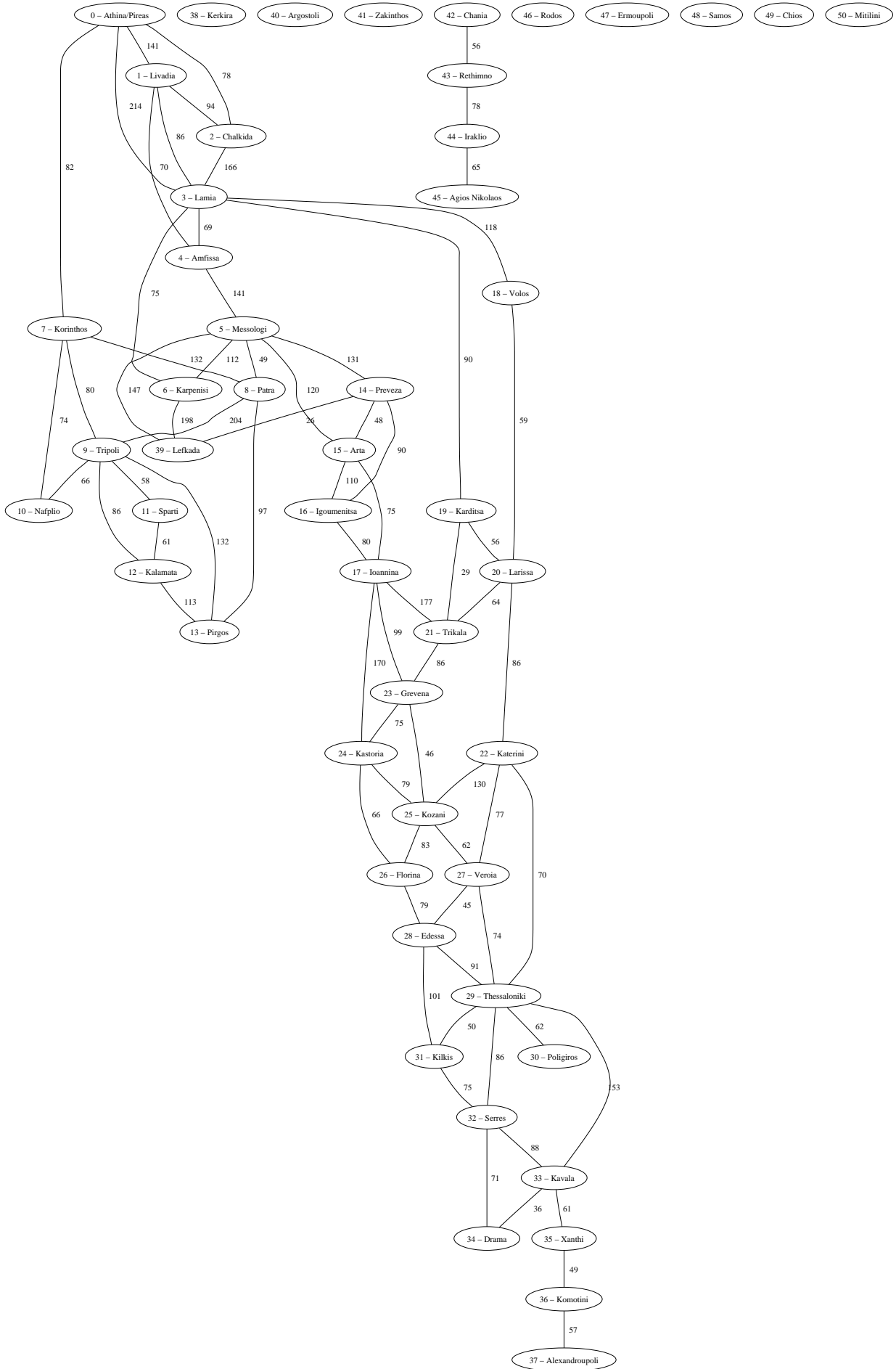
Επαναληπτική υλοποίηση του αλγορίθμου Floyd-Warshall

Παρατηρείτε ότι όσο μεγαλώνει ο γράφος, τόσο πιο πολύ αργεί η εκτέλεση του προγράμματός σας; Και μάλιστα, η αύξηση του χρόνου εκτέλεσης γίνεται με μεγάλο ρυθμό, που ονομάζεται εκθετικός, και είναι τέτοιος που από κάποιο μέγεθος γράφου και πάνω, το πρόγραμμά σας πρακτικά δεν δίνει καθόλου αποτελέσματα. Όπως ίσως μπορείτε να καταλάβετε, αυτό οφείλεται στο ότι στην αναδρομική συνάρτηση που έχετε γράψει, καλείται ο εαυτός της τρεις φορές. Για τη διόρθωση αυτού του προβλήματος, η λύση είναι η συνάρτηση αυτή να υλοποιηθεί με επαναληπτικό τρόπο, αντί για αναδρομικό.

Χωρίς να ακυρώσετε την αναδρομική υλοποίησή σας, γράψτε πάλι τη συνάρτηση υπολογισμού των μηκών των ελάχιστων μονοπατιών μεταξύ όλων των ζευγαριών κόμβων του γράφου, αλλά με επαναληπτική λογική αυτή τη φορά. Οι δύο συναρτήσεις σας (αναδρομική και επαναληπτική) να έχουν ακριβώς τις ίδιες προδιαγραφές και πρωτότυπο, αλλά το ποια θα καλείται τελικά να καθορίζεται από το αν θα έχετε ορίσει ή όχι μία συμβολική σταθερά μέσα στο πρόγραμμά σας (δείτε τα σχετικά με `#ifdef/#else/#endif` στη σελίδα 159 των σημειώσεων/διαφανειών του μαθήματος).

Πλέον, με την επαναληπτική υλοποίηση, μπορείτε να εφαρμόζετε τον αλγόριθμο Floyd-Warshall για να βρίσκετε τα μήκη των ελάχιστων μονοπατιών μεταξύ όλων των ζευγαριών κόμβων του γράφου, ακόμα και για πολύ μεγάλους γράφους. Εκτελέστε το πρόγραμμά σας, στην επαναληπτική εκδοχή του, για τον γράφο που φαίνεται στην επόμενη σελίδα. Ο γράφος αυτός έχει σαν κόμβους τις πρωτεύουσες των νομών της Ελλάδας και περιλαμβάνει ένα μεγάλο πλήθος από δρόμους που τις συνδέουν, μαζί με τα αντίστοιχα μήκη των δρόμων. Τον γράφο αυτό, στη μορφή που απαιτεί το πρόγραμμα “flower” μπορείτε να τον βρείτε στο <http://www.di.uoa.gr/~ip/hwfiles/flower/greece.txt>. Κάτω από το <http://www.di.uoa.gr/~ip/hwfiles/flower> μπορείτε να βρείτε και τα αρχεία `greece07.txt`, `greece14.txt`, `greece18.txt`, `greece22.txt` και `greece38.txt`, που αναπαριστούν υποσύνολα του πλήρους γράφου (οι πρώτες 7, 14, 18, 22 και 38 πρωτεύουσες, αντίστοιχα — όλες είναι 51). Οπότε, μπορείτε να δοκιμάσετε το πρόγραμμά σας και για αυτές τις ενδιάμεσου μεγέθους εισόδους.

Ένα παράδειγμα εκτέλεσης φαίνεται από τη μεθεπόμενη σελίδα και μετά.



```
% ./flowar < greece14.txt
```

```
From node 1 to node 0: Length of shortest path is 141
```

```
From node 2 to node 0: Length of shortest path is 78
```

```
From node 2 to node 1: Length of shortest path is 94
```

```
From node 3 to node 0: Length of shortest path is 214
```

```
From node 3 to node 1: Length of shortest path is 86
```

```
From node 3 to node 2: Length of shortest path is 166
```

```
From node 4 to node 0: Length of shortest path is 211
```

```
From node 4 to node 1: Length of shortest path is 70
```

```
From node 4 to node 2: Length of shortest path is 164
```

```
From node 4 to node 3: Length of shortest path is 69
```

```
From node 5 to node 0: Length of shortest path is 263
```

```
From node 5 to node 1: Length of shortest path is 211
```

```
From node 5 to node 2: Length of shortest path is 305
```

```
From node 5 to node 3: Length of shortest path is 187
```

```
From node 5 to node 4: Length of shortest path is 141
```

```
From node 6 to node 0: Length of shortest path is 289
```

```
From node 6 to node 1: Length of shortest path is 161
```

```
From node 6 to node 2: Length of shortest path is 241
```

```
From node 6 to node 3: Length of shortest path is 75
```

```
From node 6 to node 4: Length of shortest path is 144
```

```
From node 6 to node 5: Length of shortest path is 112
```

```
From node 7 to node 0: Length of shortest path is 82
```

```
From node 7 to node 1: Length of shortest path is 223
```

```
From node 7 to node 2: Length of shortest path is 160
```

```
From node 7 to node 3: Length of shortest path is 296
```

```
From node 7 to node 4: Length of shortest path is 293
```

```
From node 7 to node 5: Length of shortest path is 181
```

```
From node 7 to node 6: Length of shortest path is 293
```

```
From node 8 to node 0: Length of shortest path is 214
```

```
From node 8 to node 1: Length of shortest path is 260
```

```
From node 8 to node 2: Length of shortest path is 292
```

```
From node 8 to node 3: Length of shortest path is 236
```

```
From node 8 to node 4: Length of shortest path is 190
```

```
From node 8 to node 5: Length of shortest path is 49
```

```
From node 8 to node 6: Length of shortest path is 161
```

```
From node 8 to node 7: Length of shortest path is 132
```

```
From node 9 to node 0: Length of shortest path is 162
```

```
From node 9 to node 1: Length of shortest path is 303
```

```
From node 9 to node 2: Length of shortest path is 240
```

From node 9 to node 3: Length of shortest path is 376
From node 9 to node 4: Length of shortest path is 373
From node 9 to node 5: Length of shortest path is 253
From node 9 to node 6: Length of shortest path is 365
From node 9 to node 7: Length of shortest path is 80
From node 9 to node 8: Length of shortest path is 204

From node 10 to node 0: Length of shortest path is 156
From node 10 to node 1: Length of shortest path is 297
From node 10 to node 2: Length of shortest path is 234
From node 10 to node 3: Length of shortest path is 370
From node 10 to node 4: Length of shortest path is 367
From node 10 to node 5: Length of shortest path is 255
From node 10 to node 6: Length of shortest path is 367
From node 10 to node 7: Length of shortest path is 74
From node 10 to node 8: Length of shortest path is 206
From node 10 to node 9: Length of shortest path is 66

From node 11 to node 0: Length of shortest path is 220
From node 11 to node 1: Length of shortest path is 361
From node 11 to node 2: Length of shortest path is 298
From node 11 to node 3: Length of shortest path is 434
From node 11 to node 4: Length of shortest path is 431
From node 11 to node 5: Length of shortest path is 311
From node 11 to node 6: Length of shortest path is 423
From node 11 to node 7: Length of shortest path is 138
From node 11 to node 8: Length of shortest path is 262
From node 11 to node 9: Length of shortest path is 58
From node 11 to node 10: Length of shortest path is 124

From node 12 to node 0: Length of shortest path is 248
From node 12 to node 1: Length of shortest path is 389
From node 12 to node 2: Length of shortest path is 326
From node 12 to node 3: Length of shortest path is 446
From node 12 to node 4: Length of shortest path is 400
From node 12 to node 5: Length of shortest path is 259
From node 12 to node 6: Length of shortest path is 371
From node 12 to node 7: Length of shortest path is 166
From node 12 to node 8: Length of shortest path is 210
From node 12 to node 9: Length of shortest path is 86
From node 12 to node 10: Length of shortest path is 152
From node 12 to node 11: Length of shortest path is 61

From node 13 to node 0: Length of shortest path is 294
From node 13 to node 1: Length of shortest path is 357
From node 13 to node 2: Length of shortest path is 372
From node 13 to node 3: Length of shortest path is 333
From node 13 to node 4: Length of shortest path is 287
From node 13 to node 5: Length of shortest path is 146


```
From node 13 to node 6: Length of shortest path is 258
From node 13 to node 7: Length of shortest path is 212
From node 13 to node 8: Length of shortest path is 97
From node 13 to node 9: Length of shortest path is 132
From node 13 to node 10: Length of shortest path is 198
From node 13 to node 11: Length of shortest path is 174
From node 13 to node 12: Length of shortest path is 113
%
```

Υπολογισμός ελάχιστων μονοπατιών (BONUS)

Αν επεκτείνετε την επαναληπτική υλοποίηση του αλγορίθμου Floyd-Warshall ώστε να υπολογίζονται και τα ελάχιστα μονοπάτια (εκτός από τα ελάχιστα κόστη) μεταξύ όλων των ζευγαριών κόμβων του γράφου, αυτό θα αξιολογηθεί με επιπλέον 20% στη βαθμολογία της άσκησης. Ένα παράδειγμα εκτέλεσης και με υπολογισμό μονοπατιών είναι το εξής:

```
% ./flower < greece07.txt
```

```
From node 1 to node 0: Length of shortest path is 141
Shortest path is: 1 -> 0
```

```
From node 2 to node 0: Length of shortest path is 78
Shortest path is: 2 -> 0
```

```
From node 2 to node 1: Length of shortest path is 94
Shortest path is: 2 -> 1
```

```
From node 3 to node 0: Length of shortest path is 214
Shortest path is: 3 -> 0
```

```
From node 3 to node 1: Length of shortest path is 86
Shortest path is: 3 -> 1
```

```
From node 3 to node 2: Length of shortest path is 166
Shortest path is: 3 -> 2
```

```
From node 4 to node 0: Length of shortest path is 211
Shortest path is: 4 -> 1 -> 0
```

```
From node 4 to node 1: Length of shortest path is 70
Shortest path is: 4 -> 1
```

```
From node 4 to node 2: Length of shortest path is 164
Shortest path is: 4 -> 1 -> 2
```

```
From node 4 to node 3: Length of shortest path is 69
Shortest path is: 4 -> 3
```

```
From node 5 to node 0: Length of shortest path is 352
Shortest path is: 5 -> 4 -> 1 -> 0
```

```
From node 5 to node 1: Length of shortest path is 211
Shortest path is: 5 -> 4 -> 1
```

```
From node 5 to node 2: Length of shortest path is 305
Shortest path is: 5 -> 4 -> 1 -> 2
```

```
From node 5 to node 3: Length of shortest path is 187
```

```

Shortest path is: 5 -> 6 -> 3
From node 5 to node 4: Length of shortest path is 141
Shortest path is: 5 -> 4

From node 6 to node 0: Length of shortest path is 289
Shortest path is: 6 -> 3 -> 0
From node 6 to node 1: Length of shortest path is 161
Shortest path is: 6 -> 3 -> 1
From node 6 to node 2: Length of shortest path is 241
Shortest path is: 6 -> 3 -> 2
From node 6 to node 3: Length of shortest path is 75
Shortest path is: 6 -> 3
From node 6 to node 4: Length of shortest path is 144
Shortest path is: 6 -> 3 -> 4
From node 6 to node 5: Length of shortest path is 112
Shortest path is: 6 -> 5
%
```

Παραδοτέο

Θα πρέπει να δομήσετε το πρόγραμμά σας σε ένα σύνολο από **τουλάχιστον δύο πηγαία αρχεία C** (με κατάληξη `.c`) και **τουλάχιστον ένα αρχείο επικεφαλίδας** (με κατάληξη `.h`).

Για να παραδώσετε το σύνολο των αρχείων που θα έχετε δημιουργήσει για την άσκηση αυτή, ακολουθήστε την εξής διαδικασία. Τοποθετήστε όλα τα αρχεία μέσα σ' ένα κατάλογο που θα δημιουργήσετε, έστω με όνομα `flower`, στους σταθμούς εργασίας Suns ή Linux του Τμήματος. Χρησιμοποιώντας την εντολή `zip` ως εξής

```
zip -r flower.zip flower
```

δημιουργείτε ένα συμπιεσμένο (σε μορφή `zip`) αρχείο, με όνομα `flower.zip`, στο οποίο περιέχεται ο κατάλογος `flower` μαζί με όλα τα περιεχόμενά του.¹ Το αρχείο αυτό είναι που θα πρέπει να υποβάλετε, με διαδικασία που θα ανακοινωθεί σύντομα.

Σημείωση: Μην υποβάλετε ασυμπίεστα αρχεία ή αρχεία που είναι συμπιεσμένα σε άλλη μορφή εκτός από `zip` (π.χ. `rar`, `7z`, `tar`, `gz`, κλπ.), γιατί το σύστημα υποβολής θα τα απορρίπτει “σιωπηρά”, δηλαδή χωρίς κάποιο μήνυμα λάθους.

¹Αρχεία `zip` μπορείτε να δημιουργήσετε και στα Windows, με διάφορα προγράμματα, όπως, για παράδειγμα, το WinZip.