ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Εαρινό 2012

ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ Ι

Άσκηση 3

Καταληκτική ημερομηνία και ώρα ηλεκτρονικής υποβολής: 31/8/2012, 23:59:59

Ομιλίες (0.25 βαθμοί)

Το πρόβλημα με τις ομιλίες είναι γνωστό από την πρώτη σειρά ασκήσεων. Το ζητούμενο αυτής της άσκησης είναι να γραφεί η λύση του προβλήματος σε Prolog. Το πρόγραμμά σας πρέπει να περιέχει ένα κατηγόρημα **omilies/2** το οποίο, για το παράδειγμα της εκφώνησης της πρώτης άσκησης, θα πρέπει να συμπεριφέρεται όπως φαίνεται παρακάτω[1].

```
?- omilies('omilies.txt', Answers).
Answers = [true, true, true, true, true, true, false, false, true] ;
false.
```

Ψηφοφόροι στα άκρα (0.25 βαθμοί)

Το πρόβλημα της μετακίνησης ψηφοφόρων στα άκρα είναι γνωστό από τη δεύτερη σειρά ασκήσεων του μαθήματος. Το ζητούμενο είναι να γραφεί η λύση του προβλήματος σε Prolog. Κάποια παραδείγματα, αντίστοιχα αυτών της δεύτερης άσκησης, από τη χρήση του κυρίου κατηγορήματος **moves/2** που το πρόγραμμά σας πρέπει να περιέχει φαίνονται παρακάτω.

```
?- moves('voters1.txt', Moves).
Moves = 5 ;
false.
?- moves('voters2.txt', Moves).
Moves = 0 ;
false.
?- moves('voters3.txt', Moves).
false.
```

Προσέξτε ότι το κατηγόρημα πρέπει να επιτυγχάνει το πολύ μία φορά αν υπάρχει λύση ή να *αποτυγχάνει* αν δεν υπάρχει κάποια λύση (αντί να επιστρέφει -1 όπως αναφερόταν στην εκφώνηση της δεύτερης σειράς).

[1] Σε όλα τα παραδείγματα αυτής της σειράς ασκήσεων, ανάλογα με το σύστημα Prolog που θα χρησιμοποιήσετε, η γραμμή με το **false.** μπορεί να λέει **fail.** ή **no** ή μπορεί να μην τυπώνεται.

Νευρωζώνη (0.25+0.25 βαθμοί)

Πάνε οι εκλογές. Καιρός πλέον να εξετάσουμε τα προβλήματα της *Νευρωζώνης*. Στη Νευρωζώνη υπάρχουν χώρες με κατοίκους αλλά κανείς δε φαίνεται να ασχολείται με αυτούς. Αντιθέτως, όλοι ασχολούνται με τις τράπεζές της, που εμφανίζονται καταχρεωμένες. Υπάρχουν λοιπόν **N** τράπεζες στη Νευρωζώνη, αριθμημένες από 1 έως N. Κάθε τράπεζα εκδίδει ομόλογα τα οποία αγοράζουν οι άλλες τράπεζες. Όταν ένα ομόλογο λήξει, ή τράπεζα που το εξέδωσε πρέπει να πληρώσει ένα ποσό στην τράπεζα που το είχε αρχικά αγοράσει. Μία τέτοια κίνηση λέγεται «οφειλή». Στο χρονικό διάστημα που μελετάμε, υπάρχουν **M** οφειλές στη Νευρωζώνη. Ας δούμε μία από αυτές:

4 2 8 1 9

Ο πρώτος θετικός ακέραιος αριθμός είναι το οφειλόμενο ποσό (εν προκειμένω 4 βρισεκατομμύρια νευρώ, αλλά οι μονάδες δε θα μας απασχολήσουν περισσότερο). Το ποσό αυτό πρέπει να πληρώσει η τράπεζα 2 τη χρονική στιγμή 8 στην τράπεζα 1. Όμως, όπως όλοι ξέρουμε, τα τραπεζικά εμβάσματα ποτέ δεν εκτελούνται αυτοστιγμεί. Τη στιγμή 8 το ποσό εκταμιεύεται από την τράπεζα 2, όμως δε θα είναι διαθέσιμο στην τράπεζα 1 παρά μόνο τη χρονική στιγμή 9. Κάθε κίνηση δηλαδή είναι της μορφής «Π T_1 X_1 T_2 X_2» και σημαίνει ότι το ποσό Π εκταμιεύεται από την τράπεζα T_1 τη χρονική στιγμή X_1 και είναι διαθέσιμο στην τράπεζα T_2 τη χρονική στιγμή X_2.

Βοηθήστε τη Νευρωζώνη να μην καταρρεύσει! Θέλουμε να βρούμε ποια είναι τα *ελάχιστα* ποσά που πρέπει να διαθέτουν οι τράπεζες στην αρχή του χρόνου, ώστε να είναι δυνατό να εξωφληθούν όλες οι οφειλές. Η άσκηση σας ζητάει να γράψετε προγράμματα σε Prolog και στη γλώσσα της αρεσκείας σας (μεταξύ των C, ML και Java) που να λύνουν το παραπάνω πρόβλημα.

Ας δούμε και ένα παράδειγμα. Έστω ότι έχουμε τρεις τράπεζες και πέντε οφειλές (N=3 και M=5) οι οποίες βρίσκονται σε ένα αρχείο εισόδου, έστω **`debts.txt`**, με περιεχόμενα που φαίνονται παρακάτω:

```
3 5
2 3 0 1 4
1 2 1 1 5
2 2 5 3 8
3 1 6 2 7
4 2 8 1 9
```

Η πρώτη γραμμή του αρχείου περιέχει τα N και M και οι υπόλοιπες τις πέντε οφειλές. Η τελευταία γραμμή είναι η οφειλή που εξηγήσαμε παραπάνω. Το πρόγραμμά σας σε Prolog πρέπει να ορίζει ένα κατηγόρημα που να δουλεύει ως εξής:

```
?- neurozone('debts.txt', Solution).
Solution = [0, 4, 2] ;
false.
```

Η λύση λέει ότι αν αρχικά οι τρεις τράπεζες της Νευρωζώνης έχουν αντίστοιχα 0, 4 και 2 βρισεκατομμύρια νευρώ, τότε μπορούν να εξωφληθούν όλες οι οφειλές, ενώ αντίθετα αν οποιαδήποτε από τις τράπεζες έχει αρχικά μικρότερο ποσό, τότε αυτό δεν είναι δυνατό.

Το πρόγραμμά σας σε SML/NJ πρέπει να έχει τη συμπεριφορά που φαίνεται παρακάτω:

```
- neurozone "debts.txt";
val it = [0, 4, 2] : int list
```

Το πρόγραμμά σας σε Java, C, MLton ή OCaml πρέπει να έχει τη συμπεριφορά που φαίνεται παρακάτω.

```
$ java Neurozone debts.txt
0 4 2
$ ./a.out debts.txt
0 4 2
```

Περαιτέρω οδηγίες για την άσκηση

- Μπορείτε να δουλέψετε σε ομάδες το πολύ 2 ατόμων (αλλά μπορείτε αν θέλετε να σχηματίσετε διαφορετική ομάδα σε σχέση με τις προηγούμενες ασκήσεις).
- Δεν επιτρέπεται να μοιράζεστε ασκήσεις με άλλους συμφοιτητές σας ή να βάλετε τις ασκήσεις σας σε μέρος που άλλοι μπορούν να τις βρουν εύκολα (π.χ. σε κάποια σελίδα στο διαδίκτυο, σε ιστοτόπους συζητήσεων, …).
- Τα προγράμματα σε Prolog πρέπει να είναι σε ένα αρχείο και να δουλεύουν σε κάποιο από τα παρακάτω συστήματα SWI Prolog, GNU Prolog ή YAP.
- Το πρόγραμμα στην άλλη γλώσσα μπορεί να είναι σε C, ML (SML/NJ, MLton, OCaml) ή Java.
  Η αποστολή των προγραμμάτων θα γίνει ηλεκτρονικά μέσω του moodle, όπως και στις προηγούμενες ασκήσεις. Θα υπάρξει σχετική ανακοίνωση μόλις το submission site καταστεί ενεργό για αυτήν την άσκηση. Τα προγράμματά σας πρέπει να διαβάζουν την είσοδο όπως αναφέρεται και δεν πρέπει να έχουν κάποιου άλλους είδους έξοδο διότι δε θα γίνουν δεκτά από το σύστημα στο οποίο θα υποβληθούν.