

**ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

Α' Ομάδα Ασκήσεων "Λογικού Προγραμματισμού"
Ακαδημαϊκού Έτους 2016-17

Άσκηση 1

Μία ακολουθία από σύμβολα μπορεί να κωδικοποιηθεί σύμφωνα με τη λογική του *τρέχοντος μήκους* (run length encoding) με τον εξής τρόπο. Κάθε μέγιστη υπο-ακολουθία από το ίδιο σύμβολο S μήκους N (> 1) κωδικοποιείται σαν ένα ζευγάρι (S,N) . Υπο-ακολουθίες μήκους 1 δεν κωδικοποιούνται, αλλά απλά παραμένει στο αποτέλεσμα το σύμβολο S . Για παράδειγμα, η ακολουθία “a, a, a, b, b, c, d, d, d, d, e” κωδικοποιείται σαν “(a,3), (b,2), c, (d,4), e”. Ορίστε ένα κατηγορημα Prolog `decode_rl/2`, το οποίο όταν καλείται με πρώτο όρισμα μία λίστα που παριστάνει μία κωδικοποιημένη ακολουθία συμβόλων, σύμφωνα με τα παραπάνω, να την αποκωδικοποιεί και να επιστρέφει στο δεύτερο όρισμα το αποτέλεσμα. Τα σύμβολα μπορεί να είναι όροι Prolog, όχι όμως μεταβλητές. Μπορούν να είναι όμως δομές που περιέχουν μεταβλητές. Κάποια παραδείγματα εκτέλεσης:

```
?- decode_rl([(a,3),(b,2),c,(d,4),e], L).
L = [a,a,a,b,b,c,d,d,d,d,e]
```

```
?- decode_rl([(f(5,a),7)], L).
L = [f(5,a),f(5,a),f(5,a),f(5,a),f(5,a),f(5,a),f(5,a)]
```

```
?- decode_rl([g(X),(h(Y),3),k(Z),(m(W),4),n(U)], L).
L = [g(X),h(Y),h(Y),h(Y),k(Z),m(W),m(W),m(W),m(W),n(U)]
```

Στη συνέχεια, υλοποιήστε σε Prolog και την κωδικοποίηση *τρέχοντος μήκους*, ορίζοντας ένα κατηγορημα `encode_rl/2`, το οποίο να επιτελεί την αντίστροφη λειτουργία από αυτήν του `decode_rl/2`. Δηλαδή να παίρνει σαν πρώτο όρισμα μία λίστα συμβόλων (όρων Prolog πλην μεταβλητών) και να επιστρέφει στο δεύτερο όρισμα την κωδικοποιημένη εκδοχή της. Παραδείγματα:

```
?- encode_rl([a,a,a,b,b,c,d,d,d,d,e], L).
L = [(a,3),(b,2),c,(d,4),e]
```

```
?- encode_rl([f(5,a),f(5,a),f(5,a),f(5,a),
               f(5,a),f(5,a),f(5,a)], L).
L = [(f(5,a),7)]
```

```
?- encode_rl([g(X),h(Y),h(Y),h(Y),k(Z),
               m(W),m(W),m(W),m(W),n(U)], L).
L = [g(X),(h(Y),3),k(Z),(m(W),4),n(U)]
```

Τι απάντηση θα δοθεί, με βάση τον ορισμό που δώσατε, στην εξής ερώτηση;

```
?- encode_r1([p(3),p(X),q(X),q(Y),q(4)], L).
```

Εξηγήστε γιατί παίρνετε τη συγκεκριμένη απάντηση μέσω σχολίων στο αρχείο που θα παραδώσετε.

Παραδοτέο για την άσκηση είναι ένα **πηγαίο αρχείο Prolog** με όνομα **runlen.pl**, στο οποίο θα περιέχονται οι ορισμοί τόσο του `decode_r1/2` όσο και του `encode_r1/2`.

Άσκηση 2

Υπάρχει μία παρέα από φίλους, κάποιοι από τους οποίους λένε πάντα ψέματα, ενώ οι υπόλοιποι λένε πάντα την αλήθεια. Όλοι τους γνωρίζουν για τον καθένα άλλο, αν είναι ψεύτης ή όχι. Κάποια ημέρα, για να περάσουν την ώρα τους, παίζουν το εξής παιχνίδι. Το κάθε άτομο κάνει μία δήλωση της μορφής “στην παρέα υπάρχουν τουλάχιστον K ψεύτες”. Για παράδειγμα, έστω ότι υπάρχουν 5 άτομα και οι δηλώσεις τους είναι οι εξής:

- Ο 1ος δηλώνει: “Υπάρχουν τουλάχιστον 3 ψεύτες μεταξύ μας”
- Ο 2ος δηλώνει: “Υπάρχουν τουλάχιστον 2 ψεύτες μεταξύ μας”
- Ο 3ος δηλώνει: “Υπάρχει τουλάχιστον 1 ψεύτης μεταξύ μας”
- Ο 4ος δηλώνει: “Υπάρχουν τουλάχιστον 4 ψεύτες μεταξύ μας”
- Ο 5ος δηλώνει: “Υπάρχουν τουλάχιστον 2 ψεύτες μεταξύ μας”

Σχετικά εύκολα μπορούμε να διαπιστώσουμε ότι με βάση τα παραπάνω δεδομένα, στην παρέα πρέπει το 1ο και το 4ο άτομο να είναι ψεύτες και μόνο αυτοί.

Υλοποιήστε σε Prolog ένα κατηγορημα `liars/2`, το οποίο όταν καλείται με πρώτο όρισμα τη λίστα των αριθμών που δηλώνονται από κάθε άτομο ως ελάχιστο πλήθος ψευτών στην παρέα, να επιστρέφει στο δεύτερο όρισμα μία λίστα που να δείχνει τι είναι το κάθε άτομο της παρέας, ψεύτης ή όχι, μέσω κατάλληλης τιμής, 1 ή 0. Κάποια παραδείγματα εκτέλεσης είναι τα εξής:

```
?- liars([3,2,1,4,2], Liars).
Liars = [1,0,0,1,0]
```

```
?- liars([9,1,7,1,8,3,8,9,1,3], Liars).
Liars = [1,0,1,0,1,0,1,1,0,0]
```

```
?- liars([12,3,9,15,8,9,0,15,9,6,14,6,3,3,9], Liars).
Liars = [1,0,1,1,0,1,0,1,1,0,1,0,0,0,1]
```

```
?- liars([2,3,3,4], Liars).
no
```

```
?- liars([4,9,1,12,14,8,1,17,3,6,5,6,18,20,0,8,7,9,4,16],
        Liars).
Liars = [0,1,0,1,1,0,0,1,0,0,0,0,1,1,0,0,0,1,0,1]
```

```
?- liars([13,2,2,14,24,7,25,19,10,14,16,3,24,12,9,16,16,0,15,
          16,5,19,2,3,16], Liars).
Liars = [0,0,0,1,1,0,1,1,0,1,1,0,1,0,0,1,1,0,1,1,0,1,0,0,1]

?- liars([11,15,29,17,20,30,25,15,14,24,26,21,8,21,28,8,5,28,
          9,6,28,8,20,18,10,29,28,16,0,5], Liars).
Liars = [0,0,1,1,1,1,1,0,0,1,1,1,0,1,1,0,0,1,0,0,1,0,1,1,0,1,
          1,0,0,0]
```

Παραδοτέο για την άσκηση είναι ένα πηγαίο αρχείο **Prolog** με όνομα **liars.pl**.

Άσκηση 3

Έστω ότι έχουμε στη διάθεσή μας τις παρακάτω 28 πλάκες ντόμινο:

0	0	0	1	0	2	0	3	0	4	0	5	0	6				
	1	1		1	2		1	3		1	4		1	5		1	6
				2	2		2	3		2	4		2	5		2	6
							3	3		3	4		3	5		3	6
										4	4		4	5		4	6
													5	5		5	6
																6	6

Γράψτε ένα πρόγραμμα Prolog που να μας πληροφορεί πώς πρέπει να τις τοποθετήσουμε σ' ένα πλαίσιο 7x8, έτσι ώστε η τελική διάταξη να είναι η εξής:

3	1	2	6	6	1	2	2
3	4	1	5	3	0	3	6
5	6	6	1	2	4	5	0
5	6	4	1	3	3	0	0
6	1	0	6	3	2	4	0
4	1	5	2	4	3	5	5
4	1	0	2	4	5	2	0

Κωδικοποιήστε τα δεδομένα του προβλήματος ως εξής:

```

dominos([ (0,0), (0,1), (0,2), (0,3), (0,4), (0,5), (0,6),
          (1,1), (1,2), (1,3), (1,4), (1,5), (1,6),
          (2,2), (2,3), (2,4), (2,5), (2,6),
          (3,3), (3,4), (3,5), (3,6),
          (4,4), (4,5), (4,6),
          (5,5), (5,6),
          (6,6)]).

frame([ [3,1,2,6,6,1,2,2],
        [3,4,1,5,3,0,3,6],
        [5,6,6,1,2,4,5,0],
        [5,6,4,1,3,3,0,0],
        [6,1,0,6,3,2,4,0],
        [4,1,5,2,4,3,5,5],
        [4,1,0,2,4,5,2,0]]).

```

Ως προς τον τρόπο εμφάνισης του αποτελέσματος, ορίστε ένα κατηγορημα `put_dominos/0`, το οποίο όταν καλείται, να εκτυπώνει το ζητούμενο. Για παράδειγμα:

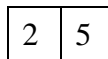
```

?- put_dominos.
3-1 2 6 6 1 2-2
   | | | |
3-4 1 5 3 0 3 6
           | |
5 6-6 1 2-4 5 0
   |       |
5 6 4 1 3 3-0 0
   | |       |
6 1 0 6 3 2 4 0
   |       | |
4 1-5 2 4 3 5 5
           | |
4-1 0-2 4 5-2 0

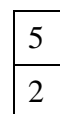
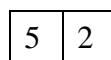
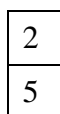
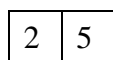
```

Φροντίστε, η υλοποίησή σας να μην είναι δεσμευτική για τα συγκεκριμένα δεδομένα, αλλά να μπορεί να εφαρμοστεί και για άλλο σύνολο από ντόμινο ή/και πλαίσιο.

Σημειώστε ότι ένα ντόμινο μπορεί να τοποθετηθεί στο πλαίσιο με οποιοδήποτε από τους τέσσερις δυνατούς τρόπους. Δηλαδή το ντόμινο



μπορεί να τοποθετηθεί με κάποιον από τους εξής τρόπους:



Παραδοτέο για την άσκηση είναι ένα πηγαίο αρχείο Prolog με όνομα **domino.pl**.