

Ασκήσεις Φυλλάδιο 8

ECLiPSe CLP

1. Το πρόβλημα των N-Βασιλισσών. Κατεβάστε το αρχείο `queens_ic.pl` από το `comprus`. Περιέχει τις υλοποιήσεις του προβλήματος των βασιλισσών σε απλή Prolog και σε CLP. Δοκιμάστε να τρέξετε το πρόβλημα με διαφορετικό αριθμό βασιλισσών και να δείτε την διαφορά ανάμεσα στις υλοποιήσεις. Για το σκοπό αυτό υπάρχει ένα κατηγορημα `measure_time(Goal)`, το οποίο δέχεται ένα ερώτημα της Prolog, το εκτελεί και τυπώνει στην οθόνη του συστήματος το χρόνο που απαιτήθηκε για την εκτέλεση (user time, system time, realtime). Για παράδειγμα

```
?- measure_time(solve(11, L)).
L = [11, 9, 7, 5, 3, 1, 10, 8, 6, 4, 2]
```

θα τυπώσει στο παράθυρο Output and Messages τα ακόλουθα

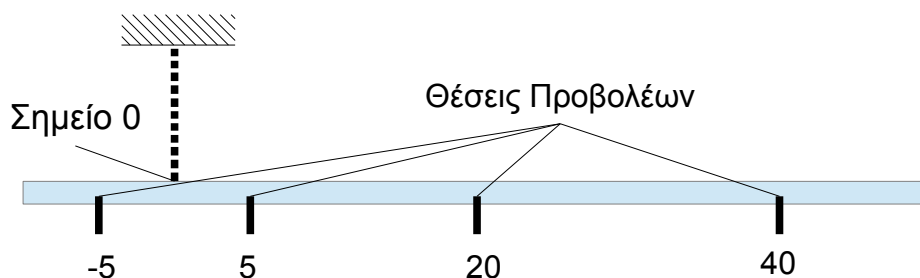
```
User Time (secs):: 1.23
Sys Time (secs):: 0.03
Clock (secs):: 1.26
```

2. Σε μια αρχιτεκτονική εταιρεία, πρέπει να ισορροπήσουν σε μια δοκό η οποία αναρτάται από ένα σημείο τέσσερα διαφορετικά φωτιστικά σώματα. Η δοκός φαίνεται στο ακόλουθο σχήμα (Σχήμα 1), όπου εμφανίζονται οι θέσεις των φωτιστικών σωμάτων και το σημείο ανάρτησης. Αν

- τα διαθέσιμα φωτιστικά έχουν βάρη που δίνονται από γεγονότα της μορφής που ακολουθεί και υπάρχει πεπερασμένος αριθμός φωτιστικών, διαφορετικού βάρους:

```
weight(10).
weight(20).
weight(30).
weight(50).
weight(60).
weight(90).
weight(100).
weight(150).
weight(250).
weight(500).
```

- οι θέσεις στη δοκό θα πρέπει έχουν φωτιστικά διαφορετικού βάρους (αρχιτεκτονική επιλογή)
- και η δοκός να ισορροπεί, δηλαδή η ροπή (βάρος * απόσταση) αριστερά του σημείου ανάρτησης να είναι ίση με την ροπή δεξιά του σημείου ανάρτησης (θεωρούμε ότι η δοκός δεν έχει βάρος),



Σχήμα 1: Δοκός Φώτων.

να αναπτύξετε ένα CLP κατηγορημα **balance_lights/2** το οποίο να βρίσκει τα βάρη των φωτιστικών που θα πρέπει να αναρτηθούν σε κάθε θέση, και θα δίνει στο δεύτερο όρισμα το συνολικό βάρος. Για παράδειγμα, η απάντηση που φαίνεται παρακάτω στην αντίστοιχη ερώτηση σημαίνει ότι στη θέση -5 θα μπει ο προβολέας με βάρος 500, στη θέση 5 εκείνος με βάρος 10, κοκ:

```
?- balance_lights(W, Total) .
W = [250, 50, 10, 20]
Total = 330
Yes
```

3. Έστω 2 αριθμοί πέντε ψηφίων ο καθένας, οι οποίοι είναι μερικώς γνωστοί, δηλαδή ξέρω μερικά από τα ψηφία και τις θέσεις τους, όπως φαίνεται παρακάτω:

[?,5,?,?,3] [?,?,0,?,1]

Γνωρίζω ότι κάθε ψηφίο από 0.. 9 εμφανίζεται μόνο μια φορά στους δύο αριθμούς.

Να γράψετε ένα CLP πρόγραμμα ώστε να υπολογίζει τα ψηφία που λείπουν στους δύο αριθμούς ώστε η διαφορά τους να είναι ακριβώς **12848**. Παράδειγμα εκτέλεσης του προγράμματος δίνεται παρακάτω:

```
?- num_gen(N1, N2) .
N1 = [6, 5, 2, 4, 3]
N2 = [7, 8, 0, 9, 1]
```

4. Κλασικά προβλήματα στο CLP είναι τα κρυπταριθμητικά παζλ. Ένα κλασσικό τέτοιο puzzle αφορά την ανάθεση τιμών από 0..9 στα γράμματα που υπάρχουν, ώστε κάθε γράμμα να πάρει μια διαφορετική τιμή και να ικανοποιείται η ακόλουθη σχέση:

```
  D O N A L D
+ G E R A L D
-----
= R O B E R T
```

Η λύση είναι η ακόλουθη:

```
[D,O,N,A,L,G,E,R,B,T]
[5,2,6,4,8,1,9,7,3,0]
```

Να υλοποιήσετε ένα κατηγορημα **donald/1**, που μοντελοποιεί το παραπάνω πρόβλημα και επιστρέφει στην μεταβλητή L, τη λύση στο puzzle. (υποθέτουμε ότι σε κάθε θέση της λίστας αντιστοιχεί ένα γράμμα, όπως φαίνεται παραπάνω). Για παράδειγμα:

```
?- donald(L) .
L = [5, 2, 6, 4, 8, 1, 9, 7, 3, 0]
Yes (0.00s cpu, solution 1, maybe more)
```

5. Σε ένα εστιατόριο οι τιμές των προσφερόμενων πιάτων έχουν διαμορφωθεί ως ακολούθως:

```
item(pizza,12) .
item(burger,14) .
item(kingburger,18) .
item(platSurprise,15) .
```

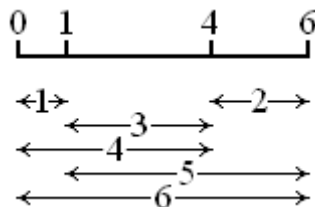
Ένας πελάτης που επισκέφτηκε το εστιατόριο ζήτησε να μάθει ποια πιάτα μπορεί να παραγγείλει ώστε το συνολικό ποσό που θα πληρώσει να είναι σε κάθε περίπτωση 50 ευρώ. Κάθε πιάτο μπορεί να παραγγελθεί περισσότερες από μια φορές, και ο πελάτης θα ήθελε να μάθει όλους τους δυνατούς συνδυασμούς. Γράψτε ένα Prolog πρόγραμμα το οποίο επιλύει το παραπάνω πρόβλημα. Στο πρόγραμμά σας θα πρέπει να έχετε ένα κατηγορημα menu(Amount, Order), το οποίο για δοθέν ποσό Amount, επιστρέφει στην λίστα Order τα πιάτα και τον αριθμό από κάθε πιάτο που μπορεί να παραγγελθεί. Το πρόγραμμα θα πρέπει να είναι γενικό με την έννοια ότι θα πρέπει να μπορεί να δουλεύει με οποιαδήποτε λίστα πιάτων, αλλάζοντας φυσικά τα αντίστοιχα γεγονότα της Prolog. Για παράδειγμα:

```
?- menu(50, L) .
L = [0 - pizza, 1 - burger, 2 - kingburger, 0 - platSurprise]
L = [3 - pizza, 1 - burger, 0 - kingburger, 0 - platSurprise]
No (0.01s cpu)

?- menu(49, L) .
No (0.01s cpu)
```

(Μπορείτε πρώτα να υπολογίσετε τις ποσότητες από κάθε πιάτο και μετά να συνδυάσετε την ποσότητα με το όνομα του πιάτου).

6. Μια σειρά από κεραίες πρέπει να τοποθετηθούν στη σειρά έτσι ώστε η απόσταση μεταξύ δύο κεραιών να είναι διαφορετική από την απόσταση μεταξύ δύο οποιονδήποτε άλλων κεραιών. Αν μετράμε η πρώτη κεραία είναι στη θέση 0, τότε ποια είναι η **ελάχιστη απόσταση** που μπορεί να τοποθετηθεί η τελευταία κεραία? Για παράδειγμα το ακόλουθο σχήμα δείχνει την τοποθέτηση 4 κεραιών σύμφωνα με τον παραπάνω κανόνα:



Να υλοποιήσετε ένα πρόγραμμα το οποίο βρίσκει τις θέσεις των κεραιών για διαφορετικούς αριθμούς κεραιών και επιστρέφει την ελάχιστη απόσταση. Για παράδειγμα

```
?- antennas(4, Max, L) .
Max = 6
L = [0, 1, 4, 6]
Yes
```

```
?- antennas(5, Max, L).
```

```
Max = 11
```

```
L = [0, 1, 4, 9, 11]
```

```
Yes
```

```
?- antennas(7, Max, L).
```

```
Max = 25
```

```
L = [0, 1, 4, 10, 18, 23, 25]
```

```
Yes
```

```
?- antennas(8, Max, L).
```

```
Max = 34
```

```
L = [0, 1, 4, 9, 15, 22, 32, 34]
```

```
Yes
```