

3η Σειρά Εργαστηριακών Ασκήσεων

Οι εργαστηριακές ασκήσεις είναι **ατομικές**. Οι απαντήσεις θα πρέπει να υποβληθούν με **turnin**, το αργότερο μέχρι την **Πέμπτη 28 Μαΐου 2020**. Πριν ξεκινήσετε να γράφεται τα προγράμματα που ζητούνται στις ασκήσεις της σειράς αυτής, **διαβάστε πολύ προσεκτικά τις αναλυτικές οδηγίες** που ακολουθούν.

Οδηγίες

- Για μεγαλύτερη ευκολία στην εκπόνηση αυτής και της επόμενης εργαστηριακής άσκησης σας συνιστώ:
 1. να δημιουργήσετε έναν κατάλογο Prolog κάτω από το home directory σας.
 2. να χρησιμοποιήτε αυτόν τον κατάλογο για αποθήκευση των αρχείων με τα προγράμματα Prolog που θα γράψετε.
 3. να μεταβαίνετε σε αυτόν τον κατάλογο πριν εκτελέσετε τον διερμηνέα της Prolog.
- Αν θέλετε να χρησιμοποιήσετε τη Prolog στο δικό σας υπολογιστή, μπορείτε να κατεβάσετε το διερμηνέα GNU Prolog από το σύνδεσμο

<http://www.gprolog.org/>

Συνοπτικές οδηγίες για τη χρήση του GNU Prolog υπάρχουν στις σημειώσεις.

- Πριν ξεκινήσετε να γράφετε τα προγράμματα που ζητούνται στις παρακάτω ασκήσεις, θα ήταν χρήσιμο να γράψετε σε ένα αρχείο ορισμένα από τα κατηγορήματα των σημειώσεων, να φορτώσετε το αρχείο στον διερμηνέα της Prolog και να κάνετε ερωτήσεις χρησιμοποιώντας τα κατηγορήματα αυτά, έτσι ώστε να εξοικειωθείτε με την γλώσσα Prolog και το διερμηνέα της.
- Για τη συγγραφή των προγραμμάτων επιτρέπεται να χρησιμοποιήσετε προκαθορισμένα κατηγορήματα και προκαθορισμένους τελεστές **μόνο εφόσον αναφέρονται στις σημειώσεις του μαθήματος**.
- Για τη συγγραφή των προγραμμάτων θα πρέπει να χρησιμοποιήσετε το αρχείο πρότυπο Lab3.pro (που υπάρχει στην ιστοσελίδα του μαθήματος), στο οποίο για κάθε κατηγορήμα που ζητείτε να ορίσετε στις παρακάτω ασκήσεις, υπάρχει ένας κανόνας ο οποίος το ορίζει έτσι ώστε να επιστρέφει πάντα την απάντηση no. Για να απαντήσετε στις ασκήσεις αντικαταστήστε τους παραπάνω κανόνες με ένα κατάλληλο σύνολο προτάσεων που να ορίζει το κάθε κατηγορήμα. **Δεν θα πρέπει να τροποποιήσετε το όνομα κανενός κατηγορήματος ούτε το πλήθος των ορισμάτων του.**
- Οι ασκήσεις 1 και 2 θεωρούν ως δεδομένο τον ορισμό κάποιων κατηγορημάτων. Τα κατηγορήματα αυτά ορίζονται με ένα πλήθος γεγονότων τα οποία συμπεριλαμβάνονται στο αρχείο πρότυπο Lab3.pro. **Δεν θα πρέπει να σβήσετε ούτε να τροποποιήσετε τα γεγονότα αυτά.**

- Μπορείτε να ορίσετε όσα βοηθητικά κατηγορήματα θέλετε, τα οποία θα χρησιμοποιούνται για τον ορισμό των κατηγορημάτων που σας ζητείται να υλοποιήσετε. Σε καμία περίπτωση δεν θα πρέπει να προσθέσετε άλλα ορίσματα στα κατηγορήματα που σας ζητούνται.
- Αν χρησιμοποιήσετε προκαθορισμένα κατηγορήματα ή τελεστές που δεν αναφέρονται στις σημειώσεις του μαθήματος, η αντίστοιχη άσκηση δεν θα βαθμολογηθεί.
- Ο έλεγχος της ορθότητας των απαντήσεων θα γίνει με ημι-αυτόματο τρόπο. Σε καμία περίπτωση δεν θα πρέπει ο βαθμολογητής να χρειάζεται να κάνει παρεμβάσεις στο αρχείο που θα υποβάλετε. Συνεπώς θα πρέπει να λάβετε υπόψη τα παρακάτω:
 1. Κάθε ένα από τα κατηγορήματα που σας ζητείται να υλοποιήσετε θα πρέπει να έχει το συγκεκριμένο όνομα και το συγκεκριμένο πλήθος ορισμάτων που περιγράφεται στην εκφώνηση της αντίστοιχης άσκησης και που υπάρχει στο αρχείο πρότυπο Lab3.pro. **Αν σε κάποια άσκηση το όνομα ή το πλήθος των ορισμάτων δεν συμφωνεί με αυτόν που δίνεται στην εκφώνηση, η άσκηση δεν θα βαθμολογηθεί.**
 2. Το αρχείο που θα παραδώσετε δεν θα πρέπει να περιέχει συντακτικά λάθη. Αν υπάρχουν τμήματα κώδικα που περιέχουν συντακτικά λάθη, τότε θα πρέπει να τα διορθώσετε ή να τα αφαιρέσετε πριν από την παράδοση. **Αν το αρχείο που θα υποβάλετε περιέχει συντακτικά λάθη, τότε ολόκληρη η εργαστηριακή άσκηση θα μηδενιστεί.**
 3. Οι ερωτήσεις που δίνονται στο τέλος κάθε άσκησης θα πρέπει να επιστρέφουν απάντηση. Αν κάποιες από τις επιστρεφόμενες απαντήσεις δεν είναι σωστές, αυτό θα ληφθεί υπόψη στη βαθμολογία, ωστόσο η άσκηση θα βαθμολογηθεί κανονικά. **Αν ωστόσο κάποια από τις παραπάνω ερωτήσεις δεν επιστρέφει απάντηση, (π.χ. προκαλείται υπερχείλιση στοίβας, ατέρμονος υπολογισμός ή κάποιο σφάλμα χρόνου εκτέλεσης) τότε ο βαθμός για την υλοποίηση του αντίστοιχου κατηγορήματος θα είναι μηδέν.**
 4. Κατά τη διόρθωση των ασκήσεων οι βαθμολογητές δεν θα κάνουν χρησιμοποιήσουν ερωτήσεις που εμπεριέχουν τα βοηθητικά κατηγορήματα τα οποία ενδεχομένως θα έχετε ορίσει. Η χρήση των βοηθητικών κατηγορημάτων θα πρέπει να γίνεται μέσα από τα κατηγορήματα που σας ζητείται να υλοποιήσετε.
- Μετά το τέλος της εκφώνησης κάθε άσκησης δίνονται παραδείγματα ερωτήσεων με τις αντίστοιχες αναμενόμενες απαντήσεις, που μπορείτε να χρησιμοποιήσετε για έλεγχο της ορθότητας των προγραμμάτων σας.
- Για όποιο πρόβλημα συναντήσετε κατά τη συγγραφή των προγραμμάτων στο πλαίσιο των εργαστηριακών ασκήσεων μπορείτε να στέλνετε email στην κ. Βίκυ Σταμάτη (vstamati@cs.uoi.gr) με θέμα “MYY401 - LAB-3”.
- Για υποβολή με turnin γράψτε (υποθέτοντας ότι έχετε γράψει το πρόγραμμα στο αρχείο Lab3.pro):

turnin Prolog-3@myy401 Lab3.pro

Ασκηση 1.

Τα παρακάτω κατηγορήματα ορίζονται από ένα πλήθος γεγονότων:

- `hosted(I,Y,C)`: οι I -οστοί σύγχρονοι Ολυμπιακοί Αγώνες διοργανώθηκαν στην πόλη C το έτος Y .
- `country(C,D)`: η πόλη C βρίσκεται στην χώρα D .
- `continent(D,E)`: η χώρα D βρίσκεται στην ήπειρο E .
- `dif(T1,T2)`: ο όρος $T1$ είναι διαφορετικός από τον $T2$ (προϋποθέτει ότι κανένας από τους όρους $T1$ και $T2$ δεν περιέχει μεταβλητή χωρίς δέσμευση).

Για την επεξεργασία των παραπάνω δεδομένων, σας ζητείται να γράψετε κανόνες που να ορίζουν τα παρακάτω κατηγορήματα:

- `twice(C,N)`: η πόλη C έχει διοργανώσει δύο φορές Ολυμπιακούς Αγώνες με διαφορά τουλάχιστον N χρόνια.
- `in2Cities(D)`: η χώρα D έχει διοργανώσει Ολυμπιακούς Αγώνες σε δύο διαφορετικές πόλεις.
- `in3Continents(I)`: οι I -οστοί, οι $(I+1)$ -οστοί και οι $(I+2)$ -οστοί Ολυμπιακοί Αγώνες έχουν γίνει σε τρεις διαφορετικές ηπείρους.

Για έλεγχο χρησιμοποιήστε τις παρακάτω ερωτήσεις. Οι απαντήσεις προϋποθέτουν ότι τα κατηγορήματα `hosted`, `country`, και `continent` έχουν οριστεί από τα γεγονότα που περιέχονται στο αρχείο `Lab3.pro`. Σημειώνεται ωστόσο ότι τα κατηγορήματα που θα ορίσετε θα πρέπει να δουλεύουν σωστά για οποιοδήποτε εναλλακτικό ορισμό των κατηγορημάτων `hosted`, `country`, και `continent`.

```
| ?- twice('Athens',108).  
yes  
| ?- twice('London',108).  
no  
| ?- twice('London',80).  
yes  
| ?- twice('Paris',60).  
no  
| ?- twice('Los Angeles',20).  
yes  
| ?- twice('Rome',4).  
no
```

```
| ?- in2Cities('Australia').  
yes  
| ?- in2Cities('France').  
no  
| ?- in2Cities('United Kingdom').  
no  
| ?- in2Cities('United States of America').  
yes
```

```
| ?- in3Continents(3).  
no  
| ?- in3Continents(14).  
no  
| ?- in3Continents(16).  
yes  
| ?- in3Continents(24).  
yes  
| ?- in3Continents(28).  
no  
| ?- in3Continents(29).  
yes
```

Ασκηση 2.

Μία πόλη απέκτησε πρόσφατα ένα υπερσύγχρονο μετρό για τις μετακινήσεις των κατοίκων της. Το μετρό διαθέτει περισσότερους από πενήντα σταθμούς, οι οποίοι συνδέονται με τέσσερεις διαφορετικές γραμμές: την μπλε, την πράσινη, την κόκκινη και την κίτρινη.

Οι επιβάτες χρεώνονται κάθε φορά που εισέρχονται ή μετεπιβιβάζονται σε μία γραμμή του μετρό και επιπλέον χρεώνονται για κάθε μετακίνηση από έναν σταθμό στον επόμενο. Το κόστος εισόδου σε μία γραμμή και το κόστος μετακίνησης από έναν σταθμό στον επόμενο εξαρτάται από τη συγκεκριμένη γραμμή.

Για παράδειγμα, έστω ότι ένας επιβάτης εισέρχεται στο μετρό στο σταθμό c1, μετακινείται με την κίτρινη γραμμή, περνάει από τους σταθμούς d1, e1, φτάνει στον σταθμό d1 στον οποίο μετεπιβιβάζεται στη μπλε γραμμή. Στη συνέχεια μετακινείται με τη μπλε γραμμή, περνάει από το σταθμό f1 και φτάνει στον προορισμό του που είναι ο σταθμός g1. Αν τα κόστος εισόδου και μετακίνησης από έναν σταθμό στον επόμενο στην κίτρινη γραμμή είναι αντίστοιχα 20 λεπτά και 8 λεπτά και το κόστος εισόδου και μετακίνησης από έναν σταθμό στον επόμενο στην μπλε γραμμή είναι αντίστοιχα 10 λεπτά και 15 λεπτά τότε θα πληρώσει συνολικά $20 + 3 \cdot 8 + 10 + 2 \cdot 15 = 84$ λεπτά.

Τα δεδομένα για τις συνδέσεις μεταξύ των σταθμών και το κόστος εισόδου και μετακίνησης για κάθε γραμμή περιγράφονται από τα παρακάτω κατηγορήματα Prolog:

- `nextStation(L,A,B)`: ο σταθμός B είναι ο επόμενος σταθμός μετά τον A στη γραμμή L.
- `entranceFare(L,F)`: το κόστος εισόδου ή μετεπιβίβασης στη γραμμή L είναι F.
- `travelFare(L,F)`: το κόστος μετακίνησης από έναν σταθμό στον επόμενο στη γραμμή L είναι F.

Τα κατηγορήματα `nextStation`, `entranceFare` και `travelFare` έχουν οριστεί από ένα πλήθος γεγονότων και κανόνων. Επίσης έχει οριστεί το κατηγορήμα

- `dif(T1,T2)`: ο όρος T1 είναι διαφορετικός από τον T2 (προϋποθέτει ότι κανένας από τους όρους T1 και T2 δεν περιέχει μεταβλητή χωρίς δέσμευση).

Ορίστε σε Prolog το παρακάτω κατηγορήμα:

- `move(A,B,C)`: μπορούμε να μετακινηθούμε με το μετρό από το σταθμό A στο σταθμό B, με συνολικό κόστος το πολύ C.

Για έλεγχο χρησιμοποιήστε τις παρακάτω ερωτήσεις. Οι απαντήσεις προϋποθέτουν ότι τα κατηγορήματα `nextStation`, `entranceFare` και `travelFare` έχουν οριστεί από τα γεγονότα που περιέχονται στο αρχείο `Lab3.pro`. Σημειώνεται ωστόσο ότι τα κατηγορήματα που θα ορίσετε θα πρέπει να δουλεύουν σωστά για οποιοδήποτε εναλλακτικό ορισμό των κατηγορημάτων `nextStation`, `entranceFare` και `travelFare`.

```
| ?- move(b4,b8,84) .  
no  
| ?- move(b4,b8,85) .  
yes
```

```
| ?- move(e1,f1,52).  
no  
| ?- move(e1,f1,53).  
yes  
| ?- move(d2,f5,59).  
no  
| ?- move(d2,f5,60).  
yes  
| ?- move(d2,f5,60).  
yes  
| ?- move(a7,h2,124).  
no  
| ?- move(a7,h2,125).  
yes  
| ?- move(b1,c8,131).  
no  
| ?- move(b1,c8,132).  
yes  
| ?- move(d2,f5,59).  
no  
| ?- move(d2,f5,60).  
yes  
| ?- move(d3,f6,76).  
no  
| ?- move(d3,f6,77).  
yes  
| ?- move(a1,g3,144).  
no  
| ?- move(a1,g3,145).  
yes
```

Ασκηση 3.

Γράψτε ένα πρόγραμμα σε Prolog για τον υπολογισμό της συνάρτησης f , η οποία ορίζεται για ζεύγη θετικών ακεραίων από τον παρακάτω αναδρομικό τύπο:

$$f(n, m) = \begin{cases} 0 & \text{αν } n \leq 0 \text{ ή } m \leq 0 \text{ ή } m > n \\ 1 & \text{αλλιώς, αν } m = 1 \text{ ή } m = n \\ \sum_{i=1}^m f(n-m, i) & \text{αλλιώς} \end{cases}$$

Συγκεκριμένα, ορίστε ένα κατηγορημα $f(N, M, Y)$, το οποίο θα αληθεύει αν οι τιμές n, m, y των μεταβλητών N, M, Y ικανοποιούν την ισότητα $y = f(n, m)$.

Για έλεγχο χρησιμοποιήστε τις παρακάτω τιμές:

| ?- f(4,5,Y).
Y = 0

| ?- f(3,3,Y).
Y = 1

| ?- f(5,1,Y).
Y = 1

| ?- f(6,3,Y).
Y = 3

| ?- f(7,3,Y).
Y = 4

| ?- f(8,4,Y).
Y = 5

| ?- f(10,2,Y).
Y = 5

| ?- f(10,3,Y).
Y = 8

| ?- f(10,4,Y).
Y = 9

| ?- f(12,5,Y).
Y = 13

Ασκηση 4.

Ορίστε ένα κατηγορημα $p(N)$ σε Prolog το οποίο θα αληθεύει αν και μόνο αν ο N είναι θετικός ακέραιος αριθμός, ο οποίος δεν διαιρείται από το τετράγωνο κανενός πρώτου αριθμού.

Για έλεγχο χρησιμοποιήστε τις παρακάτω τιμές:

```
| ?- p(23).  
yes
```

```
| ?- p(24).  
no
```

```
| ?- p(25).  
no
```

```
| ?- p(26).  
yes
```

```
| ?- p(121).  
no
```

```
| ?- p(122).  
yes
```

```
| ?- p(1017).  
no
```

```
| ?- p(1018).  
yes
```

```
| ?- p(10201).  
no
```

```
| ?- p(10202).  
yes
```