

4η Σειρά Εργαστηριακών Ασκήσεων

Οι εργαστηριακές ασκήσεις είναι **ατομικές**. Οι απαντήσεις θα πρέπει να υποβληθούν με **turnin**, το αργότερο μέχρι την **Δευτέρα 8 Ιουνίου 2020**. Πριν ξεκινήσετε να γράφεται τα προγράμματα που ζητούνται στις ασκήσεις της σειράς αυτής, **διαβάστε πολύ προσεκτικά τις αναλυτικές οδηγίες** που ακολουθούν.

Οδηγίες

- 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 Lab4.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 πο. 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**.
- 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 Lab4.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 υπερχειλίση στοίβας, ατέρμονος υπολογισμός ή κάποιο σφάλμα χρόνου εκτέλεσης) τότε ο βαθμός για την υλοποίηση του αντίστοιχου κατηγορήματος θα είναι μηδέν.**
4. Κατα 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-4”.
 - Για υποβολή με turnin γράψτε (υποθέτοντας ότι έχετε γράψει το πρόγραμμα στο αρχείο Lab4.pro):

turnin Prolog-4@myy401 Lab4.pro

Ασκηση 1.

Γράψτε ένα πρόγραμμα σε Prolog για την κατασκευή της λίστας με τους n πρώτους όρους μιας αριθμητικής προόδου. Συγκεκριμένα, ορίστε ένα κατηγορημα $\dots(N,M,K,L)$, το οποίο θα αληθεύει αν L είναι η λίστα $[m, m+k, \dots, m+(n-1) \cdot k]$, όπου n, m, k είναι αντίστοιχα οι τιμές των μεταβλητών N, M, K .

Για έλεγχο χρησιμοποιήστε τις παρακάτω ερωτήσεις:

| ?- ... (0,0,1,L) .

L = []

| ?- ... (1,1,4,L) .

L = [1]

| ?- ... (10,2,2,L) .

L = [2,4,6,8,10,12,14,16,18,20]

| ?- ... (20,5,32,L) .

L = [5,37,69,101,133,165,197,229,261,293,325,357,389,421,453,485,517,549,581,613]

| ?- ... (10,0,-1,L) .

L = [0,-1,-2,-3,-4,-5,-6,-7,-8,-9]

| ?- ... (5,1000,-100,L) .

L = [1000,900,800,700,600]

| ?- ... (12,735,125,L) .

L = [735,860,985,1110,1235,1360,1485,1610,1735,1860,1985,2110]

Ασκηση 2.

Με δεδομένες δύο λίστες ακεραίων αριθμών $[x_1, x_2, \dots, x_m]$ και $[y_1, y_2, \dots, y_n]$, θέλουμε να κατασκευάσουμε τη λίστα η οποία περιέχει σε αύξουσα τάξη όλες τις τιμές k για τις οποίες ισχύει $x_k = y_k$.

Γράψτε ένα πρόγραμμα σε Prolog το οποίο θα πραγματοποιεί την παραπάνω κατασκευή. Συγκεκριμένα, ορίστε ένα κατηγορημα `common(X,Y,L)` το οποίο θα αληθεύει αν X , Y και L είναι λίστες και η L περιέχει σε αύξουσα τάξη όλους τους ακέραιους που υποδεικνύουν τις θέσεις στις οποίες οι X και Y έχουν το ίδιο στοιχείο.

Για έλεγχο χρησιμοποιήστε τις παρακάτω ερωτήσεις:

```
| ?- common([], [3,5,8], L).  
L = []
```

```
| ?- common([1,2,3,4], [], L).  
L = []
```

```
| ?- common([2,4,7,10,12], [1,3,5,6,8], L).  
L = []
```

```
| ?- common([2,4,7,10,12], [2,4,7,10,12], L).  
L = [1,2,3,4,5]
```

```
| ?- common([2,4,7,10,12], [2,6,7,11,12], L).  
L = [1,3,5]
```

```
| ?- common([8,9,10,11,12], [8,9,10,11,12,13,14,15], L).  
L = [1,2,3,4,5]
```

```
| ?- common([11,12,13,14,15], [8,9,10,11,12,13,14,15], L).  
L = []
```

```
| ?- common([10,11,12,13], [8,10,12,14,16,18,20], L).  
L = [3]
```

```
| ?- common([100,150,200,250,275,300,400], [50,150,175,250], L).  
L = [2,4]
```

Ασκηση 3.

Με δεδομένη μία λίστα, θέλουμε να μετρήσουμε το πλήθος των εμφανίσεων κάθε στοιχείου σε αυτή και να σχηματίσουμε μία νέα λίστα, η οποία θα περιέχει ένα όρο της μορφής $N*X$, για κάθε στοιχείο X που εμφανίζεται N φορές στην αρχική λίστα.

Γράψτε ένα πρόγραμμα σε Prolog που θα υλοποιεί την παραπάνω κατασκευή. Συγκεκριμένα, ορίστε ένα κατηγορημα $\text{freq}(L,S)$, το οποίο θα αληθεύει αν η S είναι λίστα όρων της μορφής $N*X$, στην οποία ο όρος $N*X$ περιέχεται ακριβώς μία φορά αν και μόνο αν το στοιχείο X περιέχεται ακριβώς N φορές στη λίστα L .

Για έλεγχο χρησιμοποιήστε τις παρακάτω ερωτήσεις. Η σειρά των όρων στην επιστρεφόμενη λίστα S δεν παίζει ρόλο.

```
| ?- freq([],S).  
S = []
```

```
| ?- freq([a,b,c],S).  
S = [1*a,1*b,1*c]
```

```
| ?- freq([0,0,0,0,0,a,a,a],S).  
S = [5*0,3*a]
```

```
| ?- freq([5,5,5],S).  
S = [3*5]
```

```
| ?- freq([a,a,0,a,0,0,0,b,a,0],S).  
S = [5*0,4*a,1*b]
```

```
| ?- freq([0,1,0,1,0,1,0,1,0,1,0,1,0,1],S).  
S = [8*0,8*1]
```

```
| ?- freq([5,2,4,1,7,8,3,4,2,3,5,1,2,3,4],S).  
S = [2*1,3*2,3*3,3*4,2*5,1*7,1*8]
```

Ασκηση 4.

Με δεδομένη μία λίστα θετικών ακεραίων αριθμών $L = [x_1, x_2, \dots, x_n]$ μήκους n θέλουμε να μετρήσουμε το πλήθος των ζευγών (x_i, x_j) , με $1 \leq i < j \leq n$, για τα οποία ισχύει $x_i > 2x_j$ ή $x_j > 2x_i$. Γράψτε ένα πρόγραμμα σε Prolog το οποίο θα πραγματοποιεί τον παραπάνω υπολογισμό.

Συγκεκριμένα, ορίστε ένα κατηγορημα $\text{count}(L, C)$, το οποίο θα αληθεύει αν L είναι μία λίστα αποτελούμενη από θετικούς ακέραιους $[x_1, x_2, \dots, x_n]$ και η τιμή της C είναι $|\{(x_i, x_j) \mid 1 \leq i < j \leq n \text{ και } (x_i > 2x_j \text{ ή } x_j > 2x_i)\}|$.

Για έλεγχο χρησιμοποιήστε τις παρακάτω ερωτήσεις:

```
| ?- count([],C).  
C = 0
```

```
| ?- count([5],C).  
C = 0
```

```
| ?- count([5,10],C).  
C = 0
```

```
| ?- count([5,11],C).  
C = 1
```

```
| ?- count([7,2,84,40,18],C).  
C = 10
```

```
| ?- count([24,17,21,30,34],C).  
C = 0
```

```
| ?- count([12,15,8,20,16],C).  
C = 1
```

```
| ?- count([64,16,8,12,9,10],C).  
C = 5
```

```
| ?- count([88,97,64,108,55,12],C).  
C = 5
```

```
| ?- count([1,2,3,4,5,6,7,8,9,10],C).  
C = 20
```