

3η Σειρά Εργαστηριακών Ασκήσεων

Οι απαντήσεις θα πρέπει να το υποβληθούν με **turnin**, το αργότερο μέχρι την **Πέμπτη 7 Μαΐου 2015, ώρα 20:00**.

- 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 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.**
- Μπορείτε να χρησιμοποιήσετε 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.

- Οι ασκήσεις 1 και 2 θεωρούν ως δεδομένο τον ορισμό κάποιων κατηγορημάτων. Τα κατηγορήματα αυτό ορίζονται με ένα πλήθος γεγονότων τα οποία συμπεριλαμβάνονται στο αρχείο πρότυπο Lab3.pro. Δεν θα πρέπει να σβήσετε ούτε να τροποποιήσετε τα γεγονότα αυτά.
- 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.
- 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.hs. 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. 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.
- Για υποβολή με turnin γράψτε (υποθέτοντας ότι έχετε γράψει το πρόγραμμα στο αρχείο Lab3.pro):

turnin Prolog@myy401 Lab3.pro

Ασκηση 1.

Έστω ότι τα παρακάτω κατηγορήματα έχουν οριστεί από ένα πλήθος γεγονότων:

- $\text{item}(C,X,G,P)$: ο κωδικός C αντιστοιχεί στη συσκευασία G γραμμαρίων του προϊόντος X και η τιμή αυτής της συσκευασίας είναι P .
- $\text{purchase}(I,A,C,N)$: ο πελάτης A αγόρασε N τεμάχια από τη συσκευασία με κωδικό C και ο αριθμός τιμολογίου αυτής της αγοράς είναι I .
- $\text{discount}(A,D)$: ο πελάτης A έχει έκπτωση $D\%$ στις αγορές οποιουδήποτε προϊόντος

Γράψτε κανόνες που να ορίζουν τα παρακάτω κατηγορήματα:

- $p1(I,J)$: τα I και J είναι αριθμοί τιμολογίων αγορών που έγιναν από το ίδιο πελάτη.
- $p2(X,Y)$: τα προϊόντα X και Y έχουν αγοραστεί από τον ίδιο πελάτη.
- $p3(C)$: Υπάρχει συσκευασία του ίδιου προϊόντος με αυτό της συσκευασίας με κωδικό C , που έχει μικρότερη τιμή ανά μονάδα βάρους.
- $p4(I,T)$: το συνολικό κόστος της αγοράς με αριθμό τιμολογίου I , μετά την έκπτωση που δικαιούται ο αντίστοιχος πελάτης, είναι T .

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

```
| ?- p1(inv001,inv008).  
yes  
| ?- p1(inv003,inv005).  
no  
| ?- p1(inv004,inv004).  
yes  
| ?- setof((I,J),p1(I,J),L).  
L = [(inv001,inv001),(inv001,inv002),(inv001,inv008),(inv002,inv001),  
(inv002,inv002),(inv002,inv008),(inv003,inv003),(inv003,inv007),  
(inv004,inv004),(inv004,inv005),(inv005,inv004),(inv005,inv005),  
(inv006,inv006),(inv007,inv003),(inv007,inv007),(inv008,inv001),  
(inv008,inv002),(inv008,inv008)]
```

```

| ?- p2(coffee,sugar).
yes
| ?- p2(coffee,bread).
no
| ?- p2(milk,milk).
yes
| ?- p2(tea,tea).
no
| ?- setof((X,Y),p2(X,Y),L).
L = [(bread,bread),(bread,milk),(coffee,coffee),(coffee,milk),
(coffee,sugar),(milk,bread),(milk,coffee),(milk,milk),(milk,sugar),
(sugar,coffee),(sugar,milk),(sugar,sugar),(water,water)]

| ?- p3(n3001).
yes
| ?- p3(n3002).
no
| ?- p3(k2105).
no
| ?- setof(C,p3(C),L).
L = [d1105,n3001,z1010]

| ?- p4(inv005,T).
T = 135.0
| ?- setof((I:T),p4(I,T),L).
L = [inv001:6.4799999999999995,inv002:1.6200000000000001,
inv003:0.6824999999999998,inv004:7.3125,inv005:135.0,
inv006:4.7999999999999998,inv007:1.5924999999999998,
inv008:2.1600000000000001]

```

Ασκηση 2.

Ένα δίκτυο μεταφοράς αποτελείται από ένα σύνολο κόμβων, από τους οποίους κάποιοι είναι αφετηρίες και κάποια άλλοι προορισμοί, και ένα σύνολο συνδέσεων μεταξύ κόμβων, οι οποίες έχουν κατεύθυνση. Στο δίκτυο αυτό δεν υπάρχουν κύκλοι, κάτι που σημαίνει ότι αν φύγουμε από κάποιον κόμβο δεν υπάρχει τρόπος να επιστρέψουμε ξανά στον κόμβο αυτόν.

Μπορούμε να αναπαραστήσουμε ένα δίκτυο μεταφοράς χρησιμοποιώντας τα παρακάτω κατηγορήματα:

- `source(S)`: ο `S` είναι κόμβος αφετηρία.
- `destination(D)`: ο `D` είναι κόμβος προορισμός.
- `link(X,Y)`: υπάρχει σύνδεση από τον κόμβο `X` προς τον κόμβο `Y`.

Έστω ότι τα κατηγορήματα `source`, `destination` και `link` έχουν οριστεί από ένα πλήθος γεγονότων. Ο ορισμός του κατηγορήματος `link` εξασφαλίζει ότι το δίκτυο μεταφοράς δεν περιέχει κύκλους ούτε βρόχους. Γράψτε κανόνες που να ορίζουν τα παρακάτω κατηγορήματα:

- `path(X,Y)`: μπορούμε να μεταβούμε από κόμβο `X` στον κόμβο `Y` ακολουθώντας μία ή περισσότερες συνδέσεις (δηλαδή υπάρχει μονοπάτι μήκους τουλάχιστον 1 από την `X` στον `Y`)
- `biconnected(X,Y)`: υπάρχει σύνδεση από τον κόμβο `X` στον κόμβο `Y` και επιπλέον υπάρχει ένα εναλλακτικό μονοπάτι μήκους τουλάχιστον 2 από τον `X` στον `Y`.
- `meetpoint(S1,D1,S2,D2,X)`: οι κόμβοι `S1` και `S2` είναι αφετηρίες, οι κόμβοι `D1` και `D2` είναι προορισμοί, και ο κόμβος `X` είναι κοινός εσωτερικός κόμβος ενός μονοπατιού που συνδέει τον `S1` με τον `D1` και ενός μονοπατιού που συνδέει τον `S2` με τον `D2`.

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

```
| ?- path(d,i).  
yes  
| ?- path(g,p).  
yes  
| ?- path(p,g).  
no  
| ?- path(u,f).  
no
```

```

| ?- setof(X,path(a,X),L).
L = [d,e,h,i,j,p,x,y,z]
| ?- setof(X,path(b,X),L).
L = [d,f,h,i,k,p,v,w,x,y,z]
| ?- setof(X,path(c,X),L).
L = [f,g,i,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z]
| ?- setof(X,path(z,X),L).
no

| ?- biconnected(r,u).
yes
| ?- biconnected(d,i).
yes
| ?- biconnected(c,y).
no
| ?- biconnected(u,r).
no
| ?- setof((X-Y),biconnected(X,Y),L).
L = [d-i,r-u,w-z]

| ?- meetpoint(a,x,b,y,h).
yes
| ?- meetpoint(b,y,c,y,k).
yes
| ?- meetpoint(b,y,c,y,m).
no
| ?- meetpoint(a,x,c,z,N).
no
| ?- setof(X,meetpoint(a,x,b,y,X),L).
L = [d,h,i]
| ?- setof(X,meetpoint(a,y,b,x,X),L).
L = [d,h,i]
| ?- setof(X,meetpoint(a,x,c,y,X),L).
L = [i]
| ?- setof(X,meetpoint(a,x,b,z,X),L).
no
| ?- setof(X,meetpoint(b,x,c,y,X),L).
L = [i]
| ?- setof(X,meetpoint(b,x,c,z,X),L).
no
| ?- setof(X,meetpoint(b,y,c,z,X),L).
L = [f]

```

Ασκηση 3.

Σε ένα μάθημα που διδάσκεται σε κάποιο πανεπιστήμιο ο τελικός βαθμός προκύπτει από τους βαθμούς a, b, c τριών γραπτών εργασιών και τον βαθμό e της τελικής εξέτασης. Όλοι οι παραπάνω βαθμοί είναι πραγματικοί αριθμοί μεταξύ 1 και 10.

Για τον υπολογισμό του τελικού βαθμού, ο οποίος είναι επίσης ένας πραγματικός αριθμός μεταξύ 1 και 10, πρώτα υπολογίζεται ο αρμονικός μέσος m των τριών εργασιών. Αν το m είναι μεγαλύτερο από τον βαθμό e , τότε ο συνολικός βαθμός είναι ο μέσος όρος των δύο αυτών αριθμών (του m και του e). Σε αντίθετη περίπτωση, ο συνολικός βαθμός ισούται με τον βαθμό της τελικής εξέτασης e . Σε κάθε περίπτωση, αν οι βαθμοί και των τριών γραπτών εργασιών είναι μεγαλύτεροι ή ίσοι του 5, τότε στο συνολικό βαθμό προστίθεται bonus 20% ώστε να προκύψει ο τελικός βαθμός. Προφανώς, αν η βαθμολογία μετά την προσθεση του bonus είναι μεγαλύτερη από 10, τότε αποκόπτεται στο 10.

Ο αρμονικός μέσος τριών αριθμών x, y, z δίνεται από τον τύπο $m = \frac{3}{\frac{1}{x} + \frac{1}{y} + \frac{1}{z}}$.

Γράψτε ένα πρόγραμμα σε Prolog για τον υπολογισμό της βαθμολογίας του παραπάνω μαθήματος. Συγκεκριμένα ορίστε ένα κατηγορήμα `grade(A,B,C,E,G)`, το οποίο θα αληθεύει αν η τιμή της G είναι ο τελικός βαθμός του μαθήματος όταν οι τιμές των A, B, C είναι οι βαθμοί των γραπτών εργασιών και η τιμή της E είναι ο βαθμός της τελικής εξέτασης.

Αν κάποιο από τα ορίσματα έχει αριθμητική τιμή μικρότερη του 1 ή μεγαλύτερη του 10, τότε η επιστρεφόμενη τιμή θα πρέπει να είναι 0. Δεν μας ενδιαφέρει η συμπεριφορά του κατηγορήματος, όταν κάποιο από τα τέσσερα πρώτα ορίσματα δεν έχει αριθμητική τιμή.

Για έλεγχο χρησιμοποιήστε τις παρακάτω τιμές:

```
| ?- grade(3,7,5,8,G) .  
G = 8  
| ?- grade(9,8,4.5,4,G) .  
G = 5.2727272727272734  
| ?- grade(6,7,6,8,G) .  
G = 9.5999999999999996  
| ?- grade(10,7,5.5,2,G) .  
G = 5.4385321100917432  
| ?- grade(7,8,9,9,G) .  
G = 10  
| ?- grade(2,10,6,5,G) .  
G = 5  
| ?- grade(5,6,5,7,G) .  
G = 8.4000000000000004  
| ?- grade(1,10,2,1,G) .  
G = 1.4375  
| ?- grade(0,5,6,9,G) .  
G = 0  
| ?- grade(4,8,7,11,G) .  
G = 0
```

Ασκηση 4.

Ορίζουμε την παρακάτω αναδρομική συνάρτηση $\zeta : \mathbb{N}^4 \rightarrow \mathbb{N}$:

$$\zeta(k, n, x, y) = \begin{cases} y + k & \text{αν } k = 0 \text{ ή } n < k \\ x & \text{αν } y < k \text{ και } n = k \\ n - 2 & \text{αν } y < k \text{ και } n = k + 1 \\ k & \text{αν } y < k \text{ και } n \geq k + 2 \\ \zeta(k, n - k, x, \zeta(k, n, x, y - k)) & \text{αν } y \geq k \text{ και } n \geq k \geq 1 \end{cases}$$

Γράψτε ένα πρόγραμμα σε Prolog για τον υπολογισμό της συνάρτησης ζ . Συγκεκριμένα ορίστε ένα κατηγορημα $\text{zeta}(K, N, X, Y, F)$, το οποίο θα αληθεύει αν η τιμή της F ισούται με την τιμή της συνάρτησης ζ με ορίσματα τις τιμές των K, N, X, Y .

Για έλεγχο χρησιμοποιήστε τις παρακάτω τιμές:

```
| ?- zeta(0,21,31,41,F).  
F = 41  
| ?- zeta(1,0,10,20,F).  
F = 21  
| ?- zeta(1,1,3,5,F).  
F = 8  
| ?- zeta(1,2,4,8,F).  
F = 32  
| ?- zeta(1,3,3,4,F).  
F = 81  
| ?- zeta(1,4,2,3,F).  
F = 16  
| ?- zeta(2,1,10,12,F).  
F = 14  
| ?- zeta(2,2,10,12,F).  
F = 22  
| ?- zeta(80,2,1000,120,F).  
F = 200  
| ?- zeta(80,2,10000,1200,F).  
F = 1280
```


Ασκηση 5.

Μπορούμε να αναπαραστήσουμε φυσικούς αριθμούς στην Prolog χρησιμοποιώντας το ατομικό όρο 0 και ένα συναρτησιακό σύμβολο με ένα όρισμα, π.χ. το s . Ο φυσικός αριθμός n αναπαρίσταται με τον όρο $\underbrace{s(s(\dots s(0)\dots))}_n$ τον οποίο στη συνέχεια

θα συμβολίζουμε για συντομία με $s^n(0)$ (η μοναδιαία αναπαράσταση περιγράφεται με περισσότερες λεπτομέρειες στις σημειώσεις, σελ 38).

Η πράξη της διαίρεσης σε μοναδιαία αναπαράσταση περιγράφεται από την παρακάτω συνάρτηση (όπου το `undefined` είναι ατομικός όρος):

$$udiv(s^n(0), s^m(0)) = \begin{cases} s^{\lfloor \frac{n}{m} \rfloor} & \text{αν } m > 0 \\ undefined & \text{αν } m = 0 \end{cases}$$

Γράψτε ένα πρόγραμμα σε Prolog για υλοποίηση της διαίρεσης φυσικών αριθμών σε μοναδιαία αναπαράσταση. Συγκεκριμένα ορίστε ένα κατηγορημα `divide(X,Y,D)`, το οποίο θα αληθεύει αν η τιμή της D ισούται με την τιμή της συνάρτησης `udiv` με ορίσματα τις τιμές των X,Y .

Για έλεγχο χρησιμοποιήστε τις παρακάτω τιμές:

```
| ?- divide(s(0),s(0),D).  
D = s(0) ?  
| ?- divide(s(s(s(s(0)))) ,s(0),D).  
D = s(s(s(s(0))))  
| ?- divide(s(s(s(s(s(s(s(s(0))))))))) ,s(s(0)),D).  
D = s(s(s(s(0))))  
| ?- divide(s(s(s(s(s(s(s(s(s(s(0))))))))) ,s(s(s(0))),D).  
D = s(s(0))  
| ?- divide(s(s(s(s(s(s(s(s(s(s(s(s(0))))))))) ,s(s(s(s(0)))) ,D).  
D = s(s(0))  
| ?- divide(s(s(s(s(0)))) ,s(s(s(s(0)))) ,D).  
D = s(0)  
| ?- divide(s(s(s(s(s(0))))),s(s(s(s(0)))) ,D).  
D = s(0)  
| ?- divide(s(s(s((0)))) ,s(s(s(s(0)))) ,D).  
D = 0  
| ?- divide(s(s(s((0)))) ,0,D).  
D = undefined  
| ?- divide(0,s(s(s((0)))) ,D).  
D = 0
```