



ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



Intelligent Systems

Εργαστήριο Ευφύων Συστημάτων

ΜΑΘΗΜΑ:	Υπολογιστική Λογική & Λογικός Προγραμματισμός
ΕΡΓΑΣΙΑ:	Υποχρεωτική (30% της τελικής βαθμολογίας)
ΒΑΘΜΟΛΟΓΙΑ:	0 – 3
ΗΜΕΡΟΜΗΝΙΑ ΑΝΑΚΟΙΝΩΣΗΣ:	10/04/2020
ΗΜΕΡΟΜΗΝΙΑ ΠΑΡΑΔΟΣΗΣ:	10/06/2020
ΤΡΟΠΟΣ ΠΑΡΑΔΟΣΗΣ:	Upload στο https://elearning.auth.gr

Εκφώνηση

Έστω ότι υπάρχει το κατηγορήμα `attends/2`, στο οποίο είναι καταχωρημένα τα γεγονότα των δηλώσεων μεταπτυχιακών φοιτητών 8 μαθημάτων για ένα ακαδημαϊκό εξάμηνο ενός μεταπτυχιακού προγράμματος σπουδών. Κάθε φοιτητής δηλώνει 4 μαθήματα. Τα δύο ορίσματα του κατηγορήματος `attends` είναι το ΑΕΜ του φοιτητή και ο κωδικός του μαθήματος. Παρακάτω υπάρχουν μερικά παραδείγματα:

<code>attends(476, im216).</code>	<code>attends(478, im209).</code>	...
<code>attends(478, im216).</code>	<code>attends(480, im209).</code>	
<code>attends(484, im216).</code>	<code>attends(481, im209).</code>	
<code>attends(487, im216).</code>	<code>attends(484, im209).</code>	
<code>attends(491, im216).</code>	<code>attends(485, im209).</code>	
...	...	

Το αρχείο με όλα τα γεγονότα μπορεί να βρεθεί στην διεύθυνση:

<https://elearning.auth.gr/mod/resource/view.php?id=369463>

Το ζητούμενο είναι να κατασκευαστεί ένα πρόγραμμα εξετάσεων το οποίο να είναι βέλτιστο από πλευράς διευκόλυνσης προς τους φοιτητές, με τα εξής δεδομένα και παραδοχές:

- Τα 8 μαθήματα εξετάζονται στη διάρκεια 2,5 εβδομάδων, σύμφωνα με το εξής πρόγραμμα:
 - 1^η εβδομάδα: Δευτέρα, Τετάρτη, Παρασκευή
 - 2^η εβδομάδα: Δευτέρα, Τετάρτη, Παρασκευή
 - 3^η εβδομάδα: Δευτέρα, Τετάρτη
- Το πρόγραμμα θα πρέπει να εξασφαλίζει ότι κανείς φοιτητής δεν θα πρέπει να εξετάζεται σε πάνω από δύο μαθήματα την ίδια εβδομάδα. Αν δεν υπάρχει αυτή η δυνατότητα, τότε θα πρέπει να ελαχιστοποιείται ο αριθμός των φοιτητών για τους οποίους τυχαίνει να εξετάζονται σε παραπάνω από 2 μαθήματα την ίδια εβδομάδα. Παράδειγμα: Αν ένα πρόγραμμα εξετάσεων έχει 2 φοιτητές με πάνω από δύο μαθήματα την ίδια εβδομάδα, ενώ ένα άλλο έχει 5 τέτοιους φοιτητές, τότε το πρώτο πρόγραμμα είναι προτιμότερο από το δεύτερο.
- Αν περισσότερα του ενός προγράμματα είναι παρόμοια ως προς το παραπάνω κριτήριο, δηλαδή είναι ισοδύναμα ως προς τον αριθμό των φοιτητών που εξετάζονται σε πάνω από 2 μαθήματα σε μία εβδομάδα, τότε ένα δεύτερο κριτήριο σύγκρισης της προτίμησης μεταξύ των προγραμμάτων, είναι οι ημέρες που μεσολαβούν μεταξύ των εξεταζόμενων μαθημάτων. Παράδειγμα: Αν ένας φοιτητής εξετάζεται την ίδια εβδομάδα Δευτέρα-Τετάρτη, τότε το πρόγραμμα παίρνει σκορ +1 για αυτόν τον φοιτητή και για αυτήν την εβδομάδα. Αν την δεύτερη εβδομάδα δίνει Δευτέρα-Παρασκευή τότε το πρόγραμμα παίρνει σκορ +3 για αυτόν τον φοιτητή και για αυτήν την εβδομάδα. Αν κάποιος φοιτητής δίνει μόνο ένα μάθημα

κάποια εβδομάδα τότε αυτή η εβδομάδα παίρνει σκορ +7 για αυτόν τον φοιτητή. Αν κάποιος φοιτητής δεν δίνει κανένα μάθημα κάποια εβδομάδα, τότε αυτή η εβδομάδα δεν προσθέτει στο σκορ. Αν κάποιος φοιτητής δίνει τρία μαθήματα κάποια εβδομάδα τότε αυτή η εβδομάδα παίρνει σκορ -7 για αυτόν τον φοιτητή. Το συνολικό σκορ ενός προγράμματος προκύπτει από το άθροισμα των σκορ του κάθε φοιτητή και της κάθε εβδομάδας.

Να δημιουργηθούν τα ακόλουθα κατηγορήματα:

- **schedule(A,B,C)**, το οποίο να επιστρέφει στις μεταβλητές A, B, C ένα τυχαίο πρόγραμμα εξετάσεων για τις τρεις εβδομάδες. Μέσω backtracking το κατηγορήμα θα πρέπει να επιστρέφει εναλλακτικά προγράμματα εξετάσεων (όλους τους συνδυασμούς). Π.χ.

```
?- schedule(A,B,C).  
A = [im204,im209,im210],  
B = [im212,im214,im216],  
C = [im217,im218] ;  
A = [im204,im209,im210],  
B = [im212,im214,im216],  
C = [im218,im217] ;  
...
```

- **schedule_errors(A,B,C,E)**, στο οποίο να του δίνουμε στις μεταβλητές A, B, C ένα πρόγραμμα εξετάσεων για τις τρεις εβδομάδες και να επιστρέφει στην μεταβλητή E τον αριθμό των φοιτητών που είναι «δυσασαρεστημένοι», δηλαδή δίνουν παραπάνω από δύο μαθήματα την ίδια εβδομάδα. Π.χ.

```
?- schedule_errors([im204, im209, im210], [im212, im214, im216],  
[im217, im218], E).  
E = 1  
  
?- schedule_errors([im204, im210, im217], [im216, im212, im214],  
[im218, im209], E).  
E = 1  
  
?- schedule_errors([im204, im209, im210], [im212, im216, im218],  
[im214, im217], E).  
E = 2  
  
?- schedule_errors([im204, im209, im210], [im212, im216, im217],  
[im214, im218], E).  
E = 3  
  
?- schedule_errors([im204, im209, im210], [im212, im214, im217],  
[im216, im218], E).  
E = 4  
  
?- schedule_errors([im204, im209, im210], [im214, im217, im218],  
[im212, im216], E).  
E = 5  
  
?- schedule_errors([im204, im209, im217], [im212, im214, im218],  
[im210, im216], E).  
E = 6  
  
?- schedule_errors([im204, im209, im212], [im210, im216, im214],  
[im218, im217], E).  
E = 0
```

- **minimal_schedule_errors(A,B,C,E)**, το οποίο να επιστρέφει στις μεταβλητές A, B, C ένα πρόγραμμα εξετάσεων για τις τρεις εβδομάδες, στο οποίο ο αριθμός «δυσανεστημένων» φοιτητών να είναι ο μικρότερος δυνατός (ιδανικά 0). Αν υπάρχουν περισσότερα του ενός προγράμματα εξετάσεων με τον βέλτιστο αυτό αριθμό, τότε να επιστρέφονται μέσω backtracking. Π.χ.

```
?- minimal_schedule_errors(A,B,C,E) .
A = [im204,im209,im212] ,
B = [im210,im214,im216] ,
C = [im217,im218] ,
E = 0 ;
A = [im204,im209,im212] ,
B = [im210,im214,im216] ,
C = [im218,im217] ,
E = 0 ;
...
```

- **score_schedule(A,B,C,S)**, στο οποίο να του δίνουμε στις μεταβλητές A, B, C ένα πρόγραμμα εξετάσεων για τις τρεις εβδομάδες και να επιστρέφει στην μεταβλητή S το σκορ του προγράμματος σύμφωνα με την εκφώνηση. Προφανώς θα πρέπει να δημιουργήσετε κάποιο βοηθητικό κατηγορήμα το οποίο θα υπολογίζει το σκορ για την μία μόνο εβδομάδα των εξετάσεων ή/και κάποιο άλλο το οποίο θα υπολογίζει το σκορ για έναν φοιτητή, ανάλογα με τον τρόπο τον οποίο θα επιλέξετε να το υλοποιήσετε. Π.χ.

```
?- score_schedule([im204,im209,im212],[im210,im214,im216],[im217,im218],S) .
S = 600.
```

- **maximum_score_schedule(A,B,C,E,S)**, το οποίο να επιστρέφει στις μεταβλητές A, B, C ένα πρόγραμμα εξετάσεων για τις τρεις εβδομάδες, στο οποίο ο αριθμός E «δυσανεστημένων» φοιτητών να είναι ο μικρότερος δυνατός (ιδανικά 0) και μεταξύ αυτών των προγραμμάτων που έχουν τον ελάχιστο αριθμό «δυσανεστημένων» φοιτητών να επιστρέφει αυτά που το σκορ τους να είναι μέγιστο. Αν υπάρχουν περισσότερα του ενός προγράμματα εξετάσεων με το μέγιστο σκορ, τότε να επιστρέφονται μέσω backtracking. **Σημείωση:** Θα συνδυάσετε τα κατηγορήματα **minimal_schedule_errors** και **score_schedule**. Π.χ.

```
?- maximum_score_schedule(A,B,C,E,S) .
A = [im214,im216,im210] ,
B = [im209,im212,im204] ,
C = [im218,im217] ,
E = 0 ,
S = 622 ;
A = [im214,im216,im210] ,
B = [im209,im212,im204] ,
C = [im217,im218] ,
E = 0 ,
S = 622 ;
...
```

Παρατηρήσεις

- Η εργασία είναι υποχρεωτική και καλύπτει το 100% των 3 μονάδων του τελικού βαθμού.
 - ✓ Σε περίπτωση που δεν παραδοθεί ο φοιτητής θα έχει ως άριστα το 7 στις εξετάσεις και ως βάση το 4,5/7.
- Η εργασία μπορεί να εκπονηθεί ατομικά ή σε ομάδες το πολύ 3 ατόμων.
 - ✓ Οι εργασίες θα ελεγχθούν για αντιγραφές μέσω ειδικού λογισμικού που έχει αναπτυχθεί σε γλώσσα Prolog από τον διδάσκοντα.

- Στην αρχή του αρχείου υποχρεωτικά να περιλαμβάνονται σε σχόλια ο αριθμός μητρώου και το ονοματεπώνυμο του φοιτητή ή των φοιτητών της ομάδας.
- Ο πηγαίος κώδικας του προγράμματος να γίνει upload από έναν από τους φοιτητές της ομάδας στο σύστημα **elearning.auth.gr**.
 - ✓ Το όνομα του αρχείου να είναι '**AEM.p1**', π.χ. **1234.p1**, σε περίπτωση ομάδας που αποτελείται από έναν μόνο φοιτητή, ή '**AEM1-AEM2.p1**', π.χ. **1234-678.p1**, σε περίπτωση ομάδας που αποτελείται από δύο φοιτητές, κ.ο.κ.
- Ο κώδικας θα πρέπει να είναι επαρκώς σχολιασμένος.
- Ο κώδικας θα πρέπει να είναι καλά δομημένος, δηλαδή να γίνει χρήση βοηθητικών κατηγορημάτων όπου κρίνεται απαραίτητο.
 - ✓ Θα βαθμολογηθεί και το στυλ προγραμματισμού, όχι μόνο το σωστό αποτέλεσμα.
 - ✓ Οποιοσδήποτε τροποποιήσεις πραγματοποιηθούν στις απαιτήσεις της εργασίας, να επισημανθούν με σχόλια στην αρχή του κώδικα, ώστε να αξιολογηθούν επιπλέον.