

**Ημερομηνία Ανάρτησης: 12/11/2018**  
**Ημερομηνία Παράδοσης: 5/12/2018, 23:59μμ**  
**Αρχές Γλωσσών Προγραμματισμού**

1. (10%) Ο μικρός Μήτσος έχει  $K \geq 100$  μπισκότα και  $N \geq 1$  φίλους. Στο Μήτσο δεν αρέσουν οι δίκαιες μοιρασιές (γιατί τότε όλα θα ήταν ήρεμα και ωραία στο σχολείο). Αποφασίζει να μοιράσει τα μπισκότα στους φίλους του με τέτοιο τρόπο ώστε κάθε φίλος να πάρει από ένα τουλάχιστον μπισκότο και όλοι οι φίλοι να πάρουν ένα διαφορετικό αριθμό μπισκότων ο καθένας. Ορίστε σε Prolog τη σχέση `biscuits(K,N,L)` η οποία παίρνει ως είσοδο τον αριθμό των μπισκότων  $K$ , τον αριθμό των φίλων  $N$  και επιστρέφει στη λίστα  $L$  μια κατανομή των μπισκότων που να ικανοποιεί τις παραπάνω προϋποθέσεις. Για παράδειγμα, `biscuits(100,5,L)` μπορεί να επιστρέψει  $L=[10,30,20,5,35]$  (υπάρχουν προφανώς και άλλες λύσεις). Το πρόγραμμά σας θα ελεγχθεί με τιμές του  $K$  και  $N$  για τις οποίες θα υπάρχει πάντα λύση.
2. (15%) Δίνονται δύο λίστες  $L$  και  $M$  ίδιου μήκους οι οποίες περιέχουν φυσικούς αριθμούς και ακολουθίες από μικρά γράμματα του Αγγλικού αλφαβήτου (για παράδειγμα,  $L=[x,3,x,y,3]$  και  $M=[x,y,2,z,3]$ ). Μας ενδιαφέρει αν υπάρχει τρόπος να αντικαταστήσουμε τις μεταβλητές με φυσικούς αριθμούς έτσι ώστε οι δύο λίστες να γίνουν ταυτόσημες. Ορίστε σε Prolog τη σχέση `solvelists(L,M,S)` η οποία δεδομένων των λιστών  $L$  και  $M$  επιστρέφει μια λίστα από ζεύγη που δίνει μια πιθανή λύση στο πρόβλημά μας. Για παράδειγμα, `solvelists([x,3,x,y,3],[x,y,2,z,3],S)` επιστρέφει  $S=[(x,2),(y,3),(z,3)]$ , ενώ η ερώτηση `solvelists([4,5,xx,yy],[1,xx,3,xx],S)` αποτυγχάνει. Σε κάποιες περιπτώσεις μπορεί να υπάρχουν περισσότερες από μία λύσεις. Σε αυτή την περίπτωση αρκεί να επιστρέψετε μία από αυτές.
3. (25%) Ο μικρός Μήτσος αποφασίζει να ταιριάζει κάποια ζευγάρια από κάλτσες που υπάρχουν πεταμένες στο δωμάτιό του. Είναι γνωστός νόμος του σύμπαντος ότι το τέλει ταιριασμα καλτσών είναι αδύνατο: (σχεδόν) πάντα κάποια κάλτσα θα μένει αταίριαστη. Θέλουμε να βοηθήσουμε το Μήτσο να πετύχει το ακατόρθωτο. Δίνονται δύο λίστες  $L$  και  $M$  που περιέχουν θετικούς φυσικούς αριθμούς. Η  $L$  περιέχει τα μήκη των αριστερών καλτσών και η  $M$  τα μήκη των δεξιών (το μήκος της  $L$  δεν είναι υποχρεωτικά ίδιο με αυτό της  $M$ ). Ο Μήτσος θέλει να ταιριάζει όσο περισσότερες κάλτσες γίνεται, βάζοντας μια αριστερή με μια δεξιά. Σκοπός του είναι να ελαχιστοποιήσει την «ασχήμια» του αποτελέσματος (γιατί επίκειται έλεγχος από τη μητέρα του). Η «ασχήμια» ενός ταιριάσματος ορίζεται ως η μέγιστη απόλυτη διαφορά ανάμεσα στα μεγέθη όλων των ζευγαριών που συμμετέχουν στο ταιριασμα. Ορίστε σε Prolog τη σχέση `ugliness(L,M,U)` η οποία δεδομένων των  $L$  και  $M$  υπολογίζει την «ασχήμια»  $U$ . Για παράδειγμα, `ugliness([2,3],[2,3],U)` επιστρέφει  $U=0$ , `ugliness([2,39,41,45],[39,42,46],U)` επιστρέφει  $U=1$  και `ugliness([7,6,1,2,10],[9,11,6,3,12],U)` επιστρέφει  $U=4$ .
4. (25%) Ο μικρός Μήτσος μεγάλωσε και έγινε διευθυντής σε πολυεθνική η οποία έχει  $N$  υπάλληλους, αριθμημένους από το 1 μέχρι το  $N$ . Ο Μήτσος έχει τον αριθμό 1. Κάθε υπάλληλος (εκτός από το Μήτσο) έχει ένα προϊστάμενο. Κάθε προϊστάμενος μπορεί να έχει πολλούς υφιστάμενους για τους οποίους είναι άμεσα υπεύθυνος.

Κάθε φορά που η εταιρεία αναλαμβάνει ένα project, ο Μήτσος το αναθέτει στον υφιστάμενό του που έχει το μικρότερο αριθμό. Ο υπάλληλος αυτός με τη σειρά του αναθέτει το project στον υφιστάμενο του που έχει το μικρότερο αριθμό. Η διαδικασία συνεχίζεται μέχρι να φτάσει το project σε ένα υπάλληλο που δεν έχει υφιστάμενο, ο οποίος αναγκαστικά πρέπει να κάνει τη δουλειά.

Στο σημείο αυτό αρχίζουν τα προβλήματα. Ο υπάλληλος που κάνει τη δουλειά, παίρνει 1 νόμισμα, ο προϊστάμενος του παίρνει 2, κλπ. Ο Μήτσος παίρνει τόσα νομίσματα όσοι είναι οι υπάλληλοι στην ακολουθία που ξεκινάει από τον υπάλληλο που έκανε τη δουλειά και φτάνει μέχρι τον ίδιο το Μήτσο. Προφανώς, ο υπάλληλος που έκανε τη δουλειά, καταλαβαίνει πόσο άδικο είναι το σύστημα, και παραιτείται αναζητώντας καλύτερη τύχη σε άλλη πολυεθνική.

Όταν έρθει ένα νέο project η διαδικασία επαναλαμβάνεται αλλά χωρίς τον υπάλληλο που παραιτήθηκε. Αυτή η κατάσταση συνεχίζεται μέχρι να μείνει μόνος του ο Μήτσος στην εταιρεία, οπότε αναγκάζεται να κάνει την πρώτη (αλλά και τελευταία) του δουλειά.

Ορίστε σε Prolog τη σχέση `corporation(L,M)` όπου  $L$  είναι μια λίστα με μήκος  $N - 1$  αποτελούμενη από τους θετικούς φυσικούς αριθμούς  $a_2, a_3, \dots, a_N$  ( $1 \leq a_i < i$ ), όπου  $a_i$  είναι ο προϊστάμενος του

υπαλλήλου  $i$ , και  $M$  είναι μια λίστα με  $N$  αριθμούς, όπου ο  $i$ -οστός αριθμός είναι ίσος με τον αριθμό των νομισμάτων που πήρε ο υπάλληλος  $i$ . Για παράδειγμα, `corporation([1,2,2,4],M)` θα επιστρέψει  $M=[13,8,1,3,1]$ .

5. (25%) Έστω μια λίστα η οποία περιέχει όλους τους αριθμούς από το 1 μέχρι το  $n$ , χωρίς επαναλήψεις και όχι υποχρεωτικά στη σειρά. Υποθέτουμε ότι τη λίστα αυτή αρχικά δεν τη γνωρίζουμε. Γνωρίζουμε όμως πέντε λίστες που έχουν προκύψει από την άγνωστη λίστα με τον ακόλουθο τρόπο. Η πρώτη λίστα έχει προκύψει από την άγνωστη λίστα με μετακίνηση ενός από τους αριθμούς σε μια άλλη θέση. Η δεύτερη λίστα έχει προκύψει από τη μετακίνηση ενός διαφορετικού αριθμού από την αρχική λίστα σε μια άλλη θέση, κλπ. Ορίστε σε Prolog τη σχέση `findlist(L,M)` η οποία δεδομένης μιας λίστας  $L$  που έχει για στοιχεία πέντε λίστες, επιστρέφει ως αποτέλεσμα την αρχική λίστα. Για παράδειγμα, το ερώτημα `findlist([[1,2,5,3,4],[1,5,3,4,2],[4,2,1,5,3],[2,3,1,5,4],[2,1,3,4,5]],M)` θα πρέπει να επιστρέφει  $M=[2,1,5,3,4]$ .

**Παράδοση Ασκήσεων:** Η παράδοση πρέπει να γίνει μέχρι τις 23:59μμ, την 5/12/2018. Θα δημιουργήσετε ένα αρχείο το οποίο θα περιέχει τις λύσεις όλων των ασκήσεων και θα το στείλετε με *email* και στις δύο παρακάτω διευθύνσεις: `cs1180004@di.uoa.gr` και `prondo@di.uoa.gr`. **Δεν θα υπάρξει παράταση στην παράδοση των ασκήσεων.** Τα ονόματα των κατηγορημάτων που θα χρησιμοποιήσετε στα προγράμματά σας πρέπει να είναι **ακριβώς** τα ίδια με αυτά που καθορίζονται από την παραπάνω εκφώνηση. Καθυστερημένες ασκήσεις δεν θα βαθμολογούνται.

**Σημείωση:** Για να μπορέσει κάποιος να λάβει μέρος στην τελική εξέταση του μαθήματος, θα πρέπει να έχει παραδώσει τις δύο πρώτες εργασίες (Prolog και Haskell) με προβιβάσιμο βαθμό.