

Άσκηση 2

Καταληκτική ημερομηνία και ώρα ηλεκτρονικής υποβολής: 2/6/2019, 23:59:59

Πολύχρωμη κορδέλα, ξανά (0.25 βαθμοί)

Το πρόβλημα με την πολύχρωμη κορδέλα είναι γνωστό από την πρώτη σειρά ασκήσεων της φετινής χρονιάς. Το ζητούμενο αυτής της άσκησης είναι να γραφεί η λύση του σε Prolog. Επειδή τα συστήματα Prolog δεν τρέχουν native code, ο χρονικός περιορισμός για την άσκηση θα είναι σημαντικά αυξημένος. Το πρόγραμμά σας θα πρέπει να περιέχει ένα κατηγορημα `colors/2` το οποίο θα έχει ως πρώτο όρισμα το όνομα του αρχείου εισόδου και θα επιστρέφει στο δεύτερο όρισμά του τη λύση. Για τα παραδείγματα της εκφώνησης της πρώτης σειράς, το κατηγορημά σας θα πρέπει να συμπεριφέρεται όπως φαίνεται παρακάτω.¹

```
?- colors('c1.txt', Answer).  
Answer = 4 ;  
false.
```

```
?- colors('c2.txt', Answer).  
Answer = 10 ;  
false.
```

```
?- colors('c3.txt', Answer).  
Answer = 0 ;  
false.
```

Για το διάβασμα της εισόδου, δείτε το υπόδειγμα που δίνεται στη δεύτερη άσκηση.

Σώσε τη γάτα, ξανά (0.25 βαθμοί)

Το πρόβλημα με τη σωτηρία της γάτας είναι κι αυτό γνωστό από την πρώτη σειρά ασκήσεων της φετινής χρονιάς. Το ζητούμενο αυτής της άσκησης είναι να γραφεί η λύση του σε Python 3. Επειδή οι υλοποιήσεις της Python δεν τρέχουν native code, ο χρονικός περιορισμός για αυτή την άσκηση θα είναι αυξημένος. Το πρόγραμμά σας θα πρέπει να έχει την ίδια συμπεριφορά με τα προγράμματα σε C/C++ που παραδώσατε για την πρώτη σειρά ασκήσεων. Για τα παραδείγματα της εκφώνησης της πρώτης σειράς, η έξοδος του προγράμματός σας πρέπει να είναι η εξής:

```
$ python3 savethecat.py a1.txt  
infinity  
stay  
  
$ python3 savethecat.py a2.txt  
5  
RDDLL  
  
$ python3 savethecat.py a3.txt  
15  
DDDRRRURR
```

¹ Σε όλα τα παραδείγματα αυτής της σειράς ασκήσεων, ανάλογα με το σύστημα Prolog που θα χρησιμοποιήσετε, στις περιπτώσεις που υπάρχει κάποια λύση, η γραμμή με το `false` μπορεί να λέει `fail` ή `no` ή μπορεί να μην τυπώνεται (που μάλλον είναι το καλύτερο διότι δείχνει ότι η εκτέλεση του κατηγορημάτός σας είναι ντετερμινιστική).

```
$ python3 savethecat.py a4.txt
infinity
LLUU
```

Τι πληρώνει το λαχείο (0.25+0.25 = 0.5 βαθμοί)

Σε μία λαχειοφόρο αγορά, έστω ότι έχουν διατεθεί N λαχνοί, καθένας από τους οποίους είναι ένας φυσικός αριθμός X_i αποτελούμενος από ακριβώς K δεκαδικά ψηφία (οι λαχνοί επιτρέπεται να ξεκινούν και με το ψηφίο 0). Κληρώνεται ένας τυχερός αριθμός Y . Βάσει αυτού του αριθμού, κάθε λαχνός κερδίζει ένα χρηματικό ποσό ως εξής: αν τα M τελευταία ψηφία του λαχνού X_i είναι τα ίδια με τα M τελευταία (λήγοντα) ψηφία του τυχερού αριθμού Y , τότε ο κάτοχος του λαχνού κερδίζει $2^M - 1$ ευρώ.

Για παράδειγμα, αν $K = 6$, $X_i = 391742$ και $Y = 421742$, τότε ο λαχνός X_i έχει $M = 4$ κοινά λήγοντα ψηφία με τον τυχερό αριθμό Y (δηλαδή και οι δύο λήγουν σε 1742) και ο κάτοχός του κερδίζει $2^4 - 1 = 15$ ευρώ.

Δίνονται οι λαχνοί που έχουν διατεθεί και μία λίστα από Q πιθανούς τυχερούς αριθμούς Y_i . Για κάθε έναν τέτοιο τυχερό αριθμό Y_i ζητούνται πόσοι λαχνοί κερδίζουν κάποιο μη μηδενικό ποσό (δηλαδή πόσοι έχουν ένα ή περισσότερα κοινά λήγοντα ψηφία με τον τυχερό αριθμό Y_i) και ποιο είναι το συνολικό ποσό που κερδίζουν όλοι οι λαχνοί. Επειδή το συνολικό ποσό μπορεί να είναι πολύ μεγάλος αριθμός, ζητείται να υπολογίσετε το υπόλοιπο της ακέριας διαίρεσης (modulo) αυτού με τον αριθμό $10^9 + 7$.

Αυτό που ζητάει η άσκηση είναι να γραφούν δύο προγράμματα (ένα σε ML και ένα σε Prolog) τα οποία να παίρνουν ως είσοδο τα παραπάνω και να υπολογίζουν το ζητούμενο για κάθε εναλλακτικό τυχερό αριθμό.

Τα στοιχεία εισόδου θα διαβάζονται από ένα αρχείο με μορφή σαν και αυτή που φαίνεται στα παραδείγματα παρακάτω. Η πρώτη γραμμή του αρχείου θα περιέχει τρεις ακέριους αριθμούς K , N , και Q , χωρισμένους ανά δύο με ένα κενό διάστημα: K είναι το πλήθος των δεκαδικών ψηφίων των λαχνών, N είναι το πλήθος των λαχνών που έχουν διατεθεί και Q το πλήθος των εναλλακτικών τυχερών αριθμών. Κάθε μία από τις επόμενες N γραμμές θα περιέχει έναν ακέριο αριθμό X_i με ακριβώς K ψηφία, που παριστάνει έναν λαχνό που έχει διατεθεί. Κάθε μία από τις επόμενες Q γραμμές θα περιέχει έναν ακέριο αριθμό Y_i με ακριβώς K ψηφία, που παριστάνει έναν εναλλακτικό τυχερό αριθμό. Θεωρήστε ότι οι λαχνοί X_i είναι διαφορετικοί.

Περιορισμοί:

- Για το 20% της βαθμολογίας θα είναι: $1 \leq K \leq 9$, $1 \leq N \leq 1000$, $1 \leq Q \leq 1000$ και το συνολικό ποσό που θα κερδίζουν οι λαχνοί θα είναι μικρότερο του 10^9 .
- Για το 40% της βαθμολογίας θα είναι $1 \leq K \leq 30$, $1 \leq N \leq 1000$, $1 \leq Q \leq 1000$ και το συνολικό ποσό που θα κερδίζουν οι λαχνοί θα είναι μικρότερο του 10^9 .
- Για το 100% της βαθμολογίας θα είναι $1 \leq K \leq 100$, $1 \leq N \leq 1.000.000$, $1 \leq Q \leq 1.000.000$ και $(N+Q) \cdot K \leq 1.000.000$.

Παρακάτω δίνονται κάποια παραδείγματα σε ML και σε Prolog. Για κάθε υποψήφιο τυχερό αριθμό, η απάντησή σας πρέπει να είναι δύο ακέριοι: το πλήθος των λαχνών που κερδίζουν και το υπόλοιπο της ακέριας διαίρεσης (modulo) του συνολικού ποσού που κερδίζουν όλοι οι λαχνοί με τον αριθμό $10^9 + 7$.

Σε MLton ή σε OCaml

```
$ ./lottery t1.txt
1 15
```

```
$ ./lottery t2.txt
2 16
0 0
3 7
```

Σε SML/NJ

```
- lottery "t1.txt";
1 15
val it = () : unit
```

```
- lottery "t2.txt";
2 16
0 0
3 7
val it = () : unit
```

Σε Prolog

```
?- lottery('t1.txt', L).
L = [[1, 15]].
```

```
?- lottery('t2.txt', L).
L = [[2, 16], [0, 0], [3, 7]].
```



spoiler!

όπου τα αρχεία με τα δεδομένα εισόδου είναι τα εξής (η εντολή `cat` είναι εντολή του Unix):

<code>\$ cat t1.txt</code>	<code>\$ cat t2.txt</code>
<code>6 1 1</code>	<code>4 6 3</code>
<code>391742</code>	<code>3581</code>
<code>421742</code>	<code>2619</code>
	<code>4904</code>
	<code>8859</code>
	<code>3919</code>
	<code>4964</code>
	<code>4904</code>
	<code>8933</code>
	<code>3419</code>

Στο πρώτο παράδειγμα, υπάρχει ένας λαχνός (391742) και ένας εναλλακτικός τυχερός αριθμός (421742). Το παράδειγμα αυτό εξηγείται παραπάνω στην εκφώνηση.

Στο δεύτερο παράδειγμα, υπάρχουν έξι λαχνοί (3581, 2619, 4904, 8859, 3919, 4964) και τρεις εναλλακτικοί τυχεροί αριθμοί (4904, 8933, 3419). Αν κληρωθεί ο πρώτος (4904) τότε θα κερδίσει 15 ευρώ ο λαχνός 4904 (που συμπίπτει με τον τυχερό αριθμό) και 1 ευρώ ο 4964. Αν κληρωθεί ο δεύτερος (8933) τότε δεν κερδίζει κανένας λαχνός. Αν κληρωθεί ο τρίτος (3419) τότε οι λαχνοί 2619 και 3919 κερδίζουν από 3 ευρώ καθένας και ο 8859 κερδίζει 1 ευρώ.

Περαιτέρω οδηγίες για τις ασκήσεις

- Μπορείτε να δουλέψετε σε ομάδες το πολύ δύο ατόμων. Μπορείτε αν θέλετε να σχηματίσετε διαφορετική ομάδα σε σχέση με την προηγούμενη σειρά ασκήσεων – οι ομάδες στο σύστημα υποβολής είναι έτσι και αλλιώς καινούργιες για κάθε σειρά ασκήσεων.
- Δεν επιτρέπεται να μοιράζεστε τα προγράμματά σας με συμφοιτητές εκτός της ομάδας σας ή να τα βάλετε σε μέρος που άλλοι μπορούν να τα βρουν (π.χ. σε κάποια σελίδα στο διαδίκτυο, σε ιστοσελίδες συζητήσεων, ...). Σε περίπτωση που παρατηρηθούν «περίεργες» ομοιότητες σε προγράμματα, ο βαθμός των εμπλεκόμενων φοιτητών σε *όλες τις σειρές ασκήσεων* γίνεται αυτόματα μηδέν ανεξάρτητα από το ποια ομάδα... «εμπνεύστηκε» από την άλλη.
- Μπορείτε να χρησιμοποιήσετε «βοηθητικό» κώδικα (π.χ. κάποιο κώδικα που διαχειρίζεται κάποια δομή δεδομένων) που βρήκατε στο διαδίκτυο στα προγράμματά σας, με την προϋπόθεση ότι το πρόγραμμά σας περιέχει σε σχόλια την παραδοχή για την προέλευση αυτού του κώδικα και ένα σύνδεσμο σε αυτόν.
- Τα προγράμματα σε ML πρέπει να είναι σε ένα αρχείο και να δουλεύουν σε SML/NJ v110.76 ή σε MLton 20100608 ή σε Objective Caml version 4.01.0. Το σύστημα ηλεκτρονικής υποβολής επιτρέπει να επιλέξετε μεταξύ αυτών των υλοποιήσεων της ML.
- Τα προγράμματα σε Python πρέπει να είναι σε ένα αρχείο και να δουλεύουν σε Python 3.4.2. (Προσέξτε ότι η Python 2 είναι διαφορετική διάλεκτος της Python!)
- Τα προγράμματα σε Prolog πρέπει να είναι σε ένα αρχείο και να δουλεύουν σε κάποιο από τα παρακάτω συστήματα SWI Prolog (6.6.6), GNU Prolog (1.3.0) ή YAP (6.2.2).
- Η υποβολή των προγραμμάτων θα γίνει ηλεκτρονικά μέσω του moodle, όπως και στην προηγούμενη άσκηση, και για να μπορέσετε να τις υποβάλλετε, τα μέλη της ομάδας σας (και οι δύο) θα πρέπει να έχουν ήδη λογαριασμό στο moodle. Θα υπάρξει σχετική ανακοίνωση μόλις το σύστημα υποβολής καταστεί ενεργό. Τα προγράμματά σας πρέπει να διαβάζουν την είσοδο όπως αναφέρεται και δεν πρέπει να έχουν κάποιου άλλους είδους έξοδο εκτός από τη ζητούμενη διότι δε θα γίνουν δεκτά από το σύστημα υποβολής.