

Άσκηση 2

Καταληκτική ημερομηνία και ώρα ηλεκτρονικής υποβολής: 19/7/2010, 12:30

ΔNT (0.25 + 0.25 = 0.5 βαθμοί)

Μια γνωστή ιδιότητα της αριθμητικής είναι ότι όταν η δεκαδική αναπαράσταση ενός ακεραίου αποτελείται από $v = 3$ διαδοχικά ψηφία (σε οποιαδήποτε σειρά), τότε ο αριθμός διαιρείται ακριβώς με τον ακέριο τρία (3). Όπως μπορείτε να διαπιστώσετε και μόνοι σας η ιδιότητα ισχύει για όλους τους τριψήφιους ακεραίους των οποίων η δεκαδική αναπαράσταση αποτελείται από διαδοχικά ψηφία (π.χ. ισχύει τόσο για τον 123 όσο και για τον 768).

Ας γενικεύσουμε την παραπάνω ιδιότητα και ας την ονομάσουμε ΔNT. Λέμε ότι η ιδιότητα ΔNT ισχύει όταν όλοι οι ακεραίοι σε αναπαράσταση βάσης Δ που αποτελούνται από N ψηφία τα οποία είναι διαδοχικά διαιρούνται ακριβώς με κάποιον τρίτο ακέριο T .

Η άσκηση ζητάει να γράψετε δύο προγράμματα (ένα σε ML και ένα σε Java) που να δέχονται ως είσοδο τους τρεις αυτούς ακεραίους ($2 \leq \Delta \leq 1024$), ($1 \leq N \leq 256$) και ($1 \leq T \leq 256$) και να επιστρέφουν true αν η ιδιότητα ΔNT ισχύει για αυτούς και false αν δεν ισχύει. Παρακάτω, δείχνουμε παραδείγματα από τη χρήση του προγράμματός σας σε ML (δώστε στην αρχική σας συνάρτηση στο ML πρόγραμμά σας το παρακάτω όνομα) και σε Java.

- dnt 10 3 3;	> java Dnt 10 3 3
val it = true : bool	true
- dnt 15 4 4;	> java Dnt 15 4 4
val it = false : bool	false
- dnt 451 9 9;	> java Dnt 451 9 9
val it = true : bool	true

Master Mind (0.25 + 0.25 = 0.5 βαθμοί)

Όταν ήμουν στο Γυμνάσιο ένα από τα αγαπημένα μου επιτραπέζια παιχνίδια ήταν το MasterMind.¹ Πιθανώς κάποιοι από σας να το ξέρετε. Το αυθεντικό παιχνίδι παίζεται με δύο παίκτες και με «πιόνια» που έχουν συνολικά οκτώ (για την ακρίβεια 6+2) διαφορετικά χρώματα αλλά, για να μη ζωγραφίζω, στη συνέχεια θα περιγράψω το παιχνίδι με συμβολοσειρές που αποτελούνται από ψηφία του δεκαδικού συστήματος (0-9).

Ο ένας από τους παίκτες επιλέγει στην αρχή του παιχνιδιού έναν κώδικα (π.χ. "42") κάποιου συγκεκριμένου πλήθους ψηφίων (2 στη συγκεκριμένη περίπτωση, το πλήθος αυτό είναι γνωστό και στους δύο παίκτες) και ο δεύτερος παίκτης προσπαθεί να βρει αυτόν τον κώδικα με διαδοχικές προσπάθειες από τις οποίες παίρνει κάποια πληροφορία για το πόσα ψηφία του κώδικα έχει βρει

¹ [http://en.wikipedia.org/wiki/Mastermind_\(board_game\)](http://en.wikipedia.org/wiki/Mastermind_(board_game))

να είναι στη σωστή τους θέση και πόσα να μην υπάρχουν στον κώδικα αλλά σε λανθασμένη στον αριθμό που μάντεψε. Για παράδειγμα αν ο δεύτερος παίκτης μαντέψει τη συμβολοσειρά **"14"** τότε θα λάβει τους αριθμούς 0 και 1 ως πληροφορία από τον πρώτο παίκτη που σημαίνει ότι κανένα ψηφίο δεν είναι στη σωστή του θέση και ένα ψηφίο είναι σε λανθασμένη. Αν μετά μαντέψει τη συμβολοσειρά **"21"** θα λάβει την ίδια απάντηση (0 1) και αν στη συνέχεια μαντέψει **"22"** θα λάβει την απάντηση 1 0.

Ας δούμε και ένα παράδειγμα με οκτώ ψηφία. Αν ο κώδικας είναι ο **"12342424"** και ο δεύτερος παίκτης μαντέψει **"12233344"** τότε θα λάβει ως απάντηση τους αριθμούς 3 3 γιατί τα πρώτα 1, 2 και το τελευταίο 4 είναι στη σωστή τους θέση, το άλλο 2 το ένα 3 και το άλλο 4 υπάρχουν στα υπόλοιπα ψηφία του κώδικα αλλά είναι σε λάθος θέση.

Η άσκηση σας ζητάει να γράψετε δύο προγράμματα (ένα σε ML και ένα σε Java) τα οποία να δέχονται ως είσοδο κάποια υποθετική κατάσταση του παιχνιδιού, δηλαδή το μέγεθος των ψηφίων του κώδικα και κάποιες συμβολοσειρές που ο δεύτερος παίκτης έχει μαντέψει και με τις απαντήσεις που έχει πάρει από τον πρώτο παίκτη και να δίνει ως έξοδο τον μικρότερο δυνατό κώδικα (πάλι σε μορφή συμβολοσειράς) ο οποίος είναι ακόμα συνεπής με την κατάσταση του παιχνιδιού. Στην περίπτωση που δεν υπάρχει κάποιος τέτοιος αριθμός, δηλαδή η κατάσταση της εισόδου δεν είναι δυνατό να αντιστοιχεί σε κάποιον κώδικα με το συγκεκριμένο μέγεθος, το πρόγραμμά σας θα πρέπει να επιστρέφει -1 (στην ML τη συμβολοσειρά **"-1"**).

Το πρόγραμμά σας σε ML πρέπει να δουλεύει ως εξής:

```
- mastermind 4 [("3157",1,2),("1350",2,1),("6120",0,2),("2381",3,0)];  
val it = "2351" : string  
- mastermind 7 [];  
val it = "0000000" : string
```

Στην πρώτη περίπτωση, το αποτέλεσμα του προγράμματος ισούται με τον κώδικα.

Το πρόγραμμά σας σε Java θα πρέπει να διαβάζει την είσοδο από το stdin ως εξής: Η πρώτη γραμμή περιέχει τον αριθμό των ψηφίων του κώδικα και τον αριθμό των γραμμών που ακολουθούν. Οι επόμενες γραμμές περιέχουν τις τριάδες. Όλα αυτά φαίνονται στα παρακάτω παραδείγματα που αντιστοιχούν στα δύο παραπάνω σε ML:

```
4 4  
3157 1 2  
1350 2 1  
6120 0 2  
2381 3 0
```

και

```
7 0
```

Οι αντίστοιχες έξοδοι του προγράμματος είναι **2351** και **0000000**

Περαιτέρω οδηγίες για την άσκηση

- Μπορείτε να δουλέψετε σε ομάδες το πολύ 2 ατόμων (αλλά μπορείτε αν θέλετε να σχηματίσετε διαφορετική ομάδα σε σχέση με την προηγούμενη άσκηση – οι ομάδες θα είναι έτσι και αλλιώς καινούργιες)
- Δεν επιτρέπεται να μοιράζεστε ασκήσεις με άλλους συμφοιτητές σας ή να βάλετε τις ασκήσεις σας σε μέρος που άλλοι μπορούν να τις βρουν εύκολα (π.χ. σε κάποια σελίδα στο διαδίκτυο, σε ιστοτόπους συζητήσεων, ...)

- Τα προγράμματα σε ML πρέπει να είναι σε ένα αρχείο και να δουλεύουν σε SML/NJ v110.67 ή πιο πρόσφατη.
- Τα προγράμματα σε Java μπορεί να είναι σε περισσότερα του ενός αρχείου αλλά θα πρέπει να είναι τέτοια ώστε να μπορούν να μεταγλωττιστούν χωρίς πρόβλημα με τον Java compiler από το command line με εντολές της μορφής **javac Dnt.java** ή **javac Mastermind.java**
- Η αποστολή των προγραμμάτων θα γίνει ηλεκτρονικά μέσω του moodle, όπως και στην προηγούμενη άσκηση. Θα υπάρξει σχετική ανακοίνωση μόλις το submission site καταστεί ενεργό. Τα προγράμματά σας πρέπει να διαβάζουν την είσοδο όπως αναφέρεται και δεν πρέπει να έχουν κάποιου άλλου είδους έξοδο εκτός από τη ζητούμενη διότι δε θα γίνουν δεκτά από το σύστημα.
- Οι ασκήσεις αυτής της σειράς θα έχουν περιορισμό χρόνου εκτέλεσης ο οποίος θα καθοριστεί με επόμενη ανακοίνωση.