

## Άσκηση 1

Καταληκτική ημερομηνία και ώρα ηλεκτρονικής υποβολής: 1/5/2013, 23:59:59

### Το μεγάλο ψάρι τρώει το μικρό... (0.25+0.25 = 0.5 βαθμοί)

Σε ένα εντελώς υποθετικό παράλληλο σύμπαν υπάρχει ένας πλανήτης που μοιάζει αρκετά με τη γη. Σε μία ήπειρο αυτού του πλανήτη υπάρχουν  $N$  χώρες που, πριν λίγα χρόνια, αποφάσισαν να συνασπιστούν και να δημιουργήσουν μία οικονομική ένωση. Κάθε χώρα χαρακτηρίζεται από το (οικονομικό) μέγεθός της το οποίο, χάριν απλότητας, το παριστάνουμε με ένα θετικό φυσικό αριθμό.

Στην αρχή όλα ήταν ρόδινα στη χαρούμενη ένωση των χωρών. Όμως, όπως όλοι ξέρουμε, το μεγάλο ψάρι τρώει το μικρό! Στον παράλληλο κόσμο που μελετάμε (ουδεμία σχέση φυσικά με πραγματικές καταστάσεις και γεγονότα), αυτό σημαίνει ότι μία χώρα με μέγεθος  $A$  αργά ή γρήγορα θα καταστρέψει (οικονομικά ή αλλιώς) κάθε άλλη χώρα που έχει μέγεθος (αυστηρά) μικρότερο από  $A/2$ . Οι μικρές χώρες δηλαδή, αργά ή γρήγορα, θα αναγκαστούν σε άτακτη χρεοκοπία και αποχώρηση από την ένωση. Αν λοιπόν αφήσουμε την κατάσταση να εξελιχθεί χωρίς καμία παρέμβαση, μετά από λίγα χρόνια η ένωση χωρών θα έχει πολύ λιγότερα μέλη!

Η κεντρική κυβέρνηση της ένωσης των χωρών (ναι, υπάρχει και τέτοια!) έχει φυσικά προβλέψει τι μέλλει γενέσθαι και έχει αποφασίσει να κάνει ό,τι χρειαστεί προκειμένου στο τέλος να παραμείνουν στην ένωση όσο γίνεται περισσότερες χώρες. Η κυβέρνηση μπορεί να αποβάλει από την ένωση όποιες χώρες επιθυμεί, προκειμένου να αποτρέψει την καταστροφή και αποχώρηση κάποιων άλλων χωρών. Αυτή η αποβολή όμως κοστίζει ακριβώς στην κυβέρνηση (οικονομικά και πολιτικά) και για αυτό πρέπει να αποβληθούν όσο λιγότερες χώρες γίνεται.

Αυτό που ζητάει η άσκηση είναι να γραφούν δύο προγράμματα (ένα σε C και ένα σε ML) τα οποία να παίρνουν ως είσοδο το πλήθος και τα μεγέθη των χωρών και να επιστρέφουν ως έξοδο δύο αριθμούς: (1) το πλήθος χωρών στη βέλτιστη τελική κατάσταση, αν η κεντρική κυβέρνηση προχωρήσει σε αποβολές, και (2) το ελάχιστο πλήθος των χωρών που πρέπει να αποβληθούν έτσι ώστε η ένωση να οδηγηθεί στη βέλτιστη τελική κατάσταση.

Τα στοιχεία εισόδου θα διαβάζονται από ένα αρχείο, όπως φαίνεται στα παραδείγματα που ακολουθούν. Η πρώτη γραμμή του αρχείου θα περιέχει έναν φυσικό αριθμό  $N$ , το πλήθος των χωρών. Η δεύτερη γραμμή θα περιέχει  $N$  θετικούς φυσικούς αριθμούς  $x_1, x_2, \dots, x_N$  οι οποίοι αντιστοιχούν στα μεγέθη των χωρών, χωρισμένους ανά δύο με ένα κενό διάστημα.

Περιορισμοί:  $1 \leq N \leq 10^6$ ,  $1 \leq x_i \leq 10^9$ , όριο χρόνου εκτέλεσης: 10 seconds, όριο μνήμης: 256 MB.

Παρακάτω δείχνουμε κάποιες πιθανές κλήσεις των προγραμμάτων σε C και σε ML.

Σε C, MLton, ή σε OCaml	Σε SML/NJ
> ./countries countries1.txt	- countries "countries1.txt";
5 1	val it = (5, 1) : int * int
> ./countries countries2.txt	- countries "countries2.txt";
1 0	val it = (1, 0) : int * int
> ./countries countries3.txt	- countries "countries3.txt";
6 0	val it = (6, 0) : int * int

όπου τα τρία αρχεία με τις χώρες είναι τα εξής (η εντολή `cat` είναι εντολή του Unix):

```

> cat countries1.txt
7
5 9 8 4 2 4 6

> cat countries2.txt
3
1 5 11

> cat countries3.txt
6
5 8 7 7 4 6

```

Στο πρώτο παράδειγμα, αρκεί να αποβληθεί η χώρα με μέγεθος 9. Τότε, αφού καταλαγιάσει ο καπνός, η ένωση θα περιέχει πέντε χώρες με μεγέθη 5, 8, 4, 4, και 6 — η χώρα με μέγεθος 2 θα έχει καταστραφεί και αποχωρήσει. Δεν είναι δυνατή λύση με περισσότερες από πέντε χώρες στην τελική κατάσταση.

Στο δεύτερο παράδειγμα, η βέλτιστη τελική κατάσταση θα περιέχει αναγκαστικά μία μόνο χώρα. Δεν έχει νόημα επομένως να αποβληθεί κάποια χώρα, η κεντρική κυβέρνηση απλώς θα αφήσει τη μεγαλύτερη να καταστρέψει τις μικρότερες.

Στο τρίτο παράδειγμα, όλες οι χώρες μπορούν να συνυπάρξουν ειρηνικά και πάλι δε χρειάζεται να αποβληθεί κάποια χώρα.

## Εξωγήινο μπάσκετ... ( $0.25+0.25 = 0.5$ βαθμοί)

Στο ίδιο υποθετικό παράλληλο σύμπαν, σε έναν άλλο πλανήτη που δε μοιάζει καθόλου με τη γη υπάρχουν  $K$  είδη πλασμάτων. Στον πλανήτη αυτόν παίζουν κάποιο άθλημα που μοιάζει με το μπάσκετ αλλά με λίγο διαφορετικούς κανόνες από ό,τι ξέρουμε. Κάθε ομάδα αποτελείται από  $K$  παίκτες και αυτοί πρέπει υποχρεωτικά να είναι ένα πλάσμα από το κάθε διαφορετικό είδος του πλανήτη. Επιπλέον, για λόγους που δεν έχει νόημα να προσπαθήσουμε να καταλάβουμε, δεν έχει τόση σημασία να υπάρχουν ψηλοί παίκτες σε μια ομάδα. Έχει όμως μεγάλη σημασία όλοι οι παίκτες της ομάδας να έχουν περίπου το ίδιο ύψος.

Ο δήμαρχος ενός χωριού αποφάσισε να επενδύσει αρκετά χρήματα για να φτιάξει μία καλή ομάδα. Προσπαθεί να βρει ανάμεσα στα πλάσματα του χωριού του  $K$  έτσι ώστε αυτά να προέρχονται ένα από κάθε διαφορετικό είδος και η διαφορά ύψους του ψηλότερου από το κοντύτερο να είναι η ελάχιστη δυνατή. Τα πλάσματα του χωριού είναι συνολικά  $N$  και για κάθε ένα είναι γνωστό το είδος στο οποίο ανήκει και το ύψος του.

Αυτό που ζητάει η άσκηση είναι να γραφούν δύο προγράμματα (ένα σε C και ένα σε ML) τα οποία να παίρνουν ως είσοδο τον πληθυσμό του χωριού και να επιστρέφουν ως έξοδο έναν αριθμό: την ελάχιστη διαφορά ύψους μεταξύ του ψηλότερου και του κοντύτερου πλάσματος στην καλύτερη δυνατή ομάδα που μπορεί να σχηματίσει ο δήμαρχος.

Τα στοιχεία εισόδου θα διαβάζονται από ένα αρχείο, όπως φαίνεται στα παραδείγματα που ακολουθούν. Η πρώτη γραμμή του αρχείου θα περιέχει δύο φυσικούς αριθμούς  $N$  και  $K$ , το πλήθος των κατοίκων και το πλήθος των ειδών. Θεωρούμε ότι τα είδη είναι αριθμημένα από το 1 έως το  $K$ . Κάθε μία από τις επόμενες  $N$  γραμμές αντιστοιχεί σε ένα πλάσμα κάτοικο του χωριού και περιέχει δύο φυσικούς αριθμούς, χωρισμένους με ένα κενό διάστημα: το είδος  $E_i$  στο οποίο ανήκει το πλάσμα και το ύψος του  $Y_i$ . Θεωρήστε δεδομένο ότι πάντα θα υπάρχει ένας τουλάχιστον κάτοικος που να ανήκει σε κάθε ένα διαφορετικό είδος.

Περιορισμοί:  $1 \leq N \leq 10^6$ ,  $1 \leq K \leq 10^3$ ,  $1 \leq Y_i \leq 10^9$ , όριο χρόνου εκτέλεσης: 10 seconds, όριο μνήμης: 256 MB.

Παρακάτω δείχνουμε κάποιες πιθανές κλήσεις των προγραμμάτων σε C και σε ML.

```
Σε C, MLton, ή σε OCaml
> ./basket basket1.txt
35
> ./basket basket2.txt
0
> ./basket basket3.txt
3
```

```
Σε SML/NJ
- basket "basket1.txt";
val it = 35 : int
- basket "basket2.txt";
val it = 0 : int
- basket "basket3.txt";
val it = 3 : int
```

όπου τα τρία αρχεία με τους πληθυσμούς του χωριού είναι τα εξής:

```
> cat basket1.txt
5 5
2 188
1 174
5 209
3 199
4 193

> cat basket2.txt
10 5
5 89
4 96
3 37
1 89
5 91
5 94
2 89
4 89
3 62
3 89

> cat basket3.txt
8 2
1 44
1 53
2 40
2 41
2 66
1 77
1 37
1 71
```

Στο πρώτο παράδειγμα, ο δήμαρχος δεν έχει επιλογή: μόνο μία ομάδα μπάσκετ μπορεί να δημιουργηθεί. Η διαφορά ύψους του ψηλότερου από τον κοντύτερο παίκτη είναι  $209-174=35$ .

Στο δεύτερο παράδειγμα, ο δήμαρχος μπορεί να επιλέξει πέντε παίκτες, έναν από κάθε είδος, που να έχουν όλοι ύψος 89. Επομένως, η διαφορά ύψους είναι 0.

Στο τρίτο παράδειγμα, η βέλτιστη ομάδα αποτελείται από τον παίκτη με ύψος 40 και αυτόν με ύψος 37, με διαφορά ύψους 3.

## Περαιτέρω οδηγίες για την άσκηση

- Μπορείτε να δουλέψετε σε ομάδες το πολύ δύο ατόμων.
- 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 στις ασκήσεις γίνεται αυτόματα μηδέν ανεξάρτητα από το ποια ομάδα... «εμπνεύστηκε» από την άλλη.
- Τα προγράμματα σε C πρέπει να είναι σε ένα αρχείο και να μπορούν να μεταγλωττιστούν χωρίς warnings με gcc (version 4.4.5) με μια εντολή της μορφής (π.χ. για την πρώτη άσκηση):

```
gcc -Wall -O3 -o countries yourfile.c
```

- Τα προγράμματα σε ML πρέπει επίσης να είναι σε ένα αρχείο και να δουλεύουν σε SML/NJ v110.74 ή σε MLton 20100608 ή σε Objective Caml version 3.11.2. Το σύστημα ηλεκτρονικής υποβολής σας επιτρέπει να επιλέξετε μεταξύ αυτών των διαλέκτων της ML. (Μπορεί να προσπαθήσουμε να αναβαθμίσουμε τις εκδόσεις των παραπάνω υλοποιήσεων.)
- Η αποστολή των προγραμμάτων θα γίνει ηλεκτρονικά μέσω του moodle 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 σύντομα σχετική ανακοίνωση για την ακριβή διαδικασία υποβολής. Τα προγράμματά σας πρέπει να διαβάζουν την είσοδο όπως αναφέρεται και δεν πρέπει να έχουν κάποιου άλλου είδους έξοδο διότι δε θα γίνουν δεκτά από το σύστημα στο οποίο θα υποβληθούν.