

Άσκηση 1

Καταληκτική ημερομηνία και ώρα ηλεκτρονικής υποβολής: 17/5/2015, 23:59:59

Χαρούμενοι αριθμοί (0.25+0.25 = 0.5 βαθμοί)

Εν μέσω της πρόσφατης κρίσης, στα στελέχη του υπουργείου οικονομικών είναι πια προφανές ότι είναι αδύνατο να λάβουν μέτρα ώστε να κάνουν χαρούμενους τους Έλληνες πολίτες. Επικεντρώνονται λοιπόν στο να κάνουν χαρούμενους τους αριθμούς στον κρατικό προϋπολογισμό και στις ετήσιες οικονομικές αναφορές, με απώτερο στόχο να γίνουν χαρούμενοι αυτοί που θα τους διαβάσουν και θα τους ελέγξουν. Για να δούμε πώς ακριβώς το εννοούν.

Έστω ένας θετικός φυσικός αριθμός N . Ξεκινάμε από αυτόν και εφαρμόζουμε την ακόλουθη διαδικασία. Αθροίζουμε τα τετράγωνα όλων των ψηφίων του αριθμού, αντικαθιστούμε τον αριθμό με το αποτέλεσμα, και επαναλαμβάνουμε τη διαδικασία. Αν μετά από κάποια βήματα το αποτέλεσμα γίνει ίσο με 1 (και παραμένει εκεί), τότε λέμε ότι ο αριθμός N είναι χαρούμενος. Αντίθετα, αν η διαδικασία επαναλαμβάνεται επ' άπειρον χωρίς ποτέ να προκύπτει ο αριθμός 1, τότε λέμε ότι ο αριθμός N είναι λυπημένος.

Για παράδειγμα, ο αριθμός 7 είναι χαρούμενος γιατί η διαδικασία που περιγράφηκε παραπάνω οδηγεί στα ακόλουθα βήματα: 7, 49, 97, 130, 10, 1, 1, 1... Αντίθετα, ο αριθμός 42 είναι λυπημένος γιατί η διαδικασία οδηγεί σε μία άπειρη ακολουθία 42, 20, 4, 16, 37, 58, 89, 145, 42, 20, 4, 16, 37...

Δίνονται δύο θετικοί φυσικοί αριθμοί A και B . Το υπουργείο οικονομικών ενδιαφέρεται να μάθει πόσοι χαρούμενοι αριθμοί υπάρχουν στο διάστημα μεταξύ A και B (συμπεριλαμβανομένων). Αυτό που ζητάει η άσκηση είναι να γραφούν δύο προγράμματα (ένα σε C/C++ και ένα σε ML) τα οποία να απαντούν σε αυτό το ερώτημα.

Τα στοιχεία εισόδου διαβάζονται από ένα αρχείο αποτελούμενο από μία μόνο γραμμή, όπως φαίνεται στα παραδείγματα που ακολουθούν. Η μοναδική γραμμή του αρχείου περιέχει δύο ακέραιους αριθμούς: A και B ($A \leq B \leq 1\,000\,000\,000$), χωρισμένους με ένα κενό διάστημα.

Περιορισμοί: όριο χρόνου εκτέλεσης: 10 seconds, όριο μνήμης: 256 MB.

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

```
Σε C/C++, MLton, ή σε OCaml
> ./happy report1.txt
20
> ./happy report2.txt
138935
```

```
Σε SML/NJ
- happy "report1.txt";
val it = 20 : int
- happy "report2.txt";
val it = 138935 : int
```

όπου τα δύο αρχεία με τα όρια των αριθμών είναι τα εξής:

```
> cat report1.txt
1 100

> cat report2.txt
1000000 2000000
```

Εκρηκτικοί συνδυασμοί νέων μέτρων ($0.25+0.25 = 0.5$ βαθμοί)

Στην κατάσταση που βρίσκεται η χώρα μας, μερικά επιπρόσθετα δημοσιονομικά μέτρα μπορεί να αποδειχθούν κοινωνικά πολύ επικίνδυνα. Κάποια άλλα δημοσιονομικά μέτρα μπορεί να μην είναι κοινωνικά επικίνδυνα από μόνα τους αλλά, αν παρθούν σε συνδυασμό με κάποια άλλα μέτρα επίσης μπορεί να δημιουργήσουν κοινωνικά επικίνδυνες καταστάσεις. Τόσο η κυβέρνηση όσο και οι «θεσμοί» είναι πολύ προβληματισμένοι για το τι να κάνουν και χρειάζονται τη βοήθειά σας.

Δίνεται ένα σύνολο αποτελούμενο από N δημοσιονομικά μέτρα, αριθμημένα από το 1 έως το N . Δίνονται επίσης όλοι οι επικίνδυνοι συνδυασμοί δημοσιονομικών μέτρων. (Προσέξτε ότι ένα δημοσιονομικό μέτρο που είναι από μόνο του επικίνδυνο μπορεί να θεωρηθεί ως ένας συνδυασμός που αποτελείται μόνο από το ίδιο). Οι «θεσμοί» θέλουν να αναμείξουν όσο γίνεται περισσότερα από αυτά τα δημοσιονομικά μέτρα, χωρίς όμως να προκύψει στο μίγμα μας κανένας κοινωνικά επικίνδυνος συνδυασμός.

Η άσκηση σας ζητάει να γράψετε δύο προγράμματα (ένα σε C/C++ και ένα σε ML) τα οποία να παίρνουν ως είσοδο τις παραπάνω παραμέτρους και να επιστρέφουν ως έξοδο ένα υποσύνολο αποτελούμενο από το μέγιστο δυνατό πλήθος δημοσιονομικών μέτρων που μπορούν να επιλεγούν, χωρίς να συμπεριλαμβάνεται σε αυτά κανένας επικίνδυνος συνδυασμός.

Τα στοιχεία εισόδου θα διαβάζονται από ένα αρχείο όπως φαίνεται στο παράδειγμα που ακολουθεί. Η πρώτη γραμμή του αρχείου περιέχει δύο θετικούς ακέραιους N και M ($1 \leq N \leq 42$, $0 \leq M \leq 1000$). Η i -οστή από τις επόμενες M γραμμές περιέχει τον i -οστό επικίνδυνο συνδυασμό: έναν αριθμό K_i ($1 \leq K_i \leq N$), που παριστάνει το πλήθος των μέτρων του συνδυασμού, ακολουθούμενο από K_i αριθμούς μεταξύ του 1 και του N , των αριθμών αυτών των μέτρων.

Περιορισμοί: όριο χρόνου εκτέλεσης: 10 seconds, όριο μνήμης: 256 MB.

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

```
Σε C, C++, MLton, ή σε OCaml
> ./danger forbidden1.txt
1 3 5 6 8
```

```
Σε SML/NJ
- danger "forbidden1.txt";
val it = [1,3,5,6,8] : int list
```

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

```
> cat forbidden1.txt
8 5
2 1 2
2 3 2
1 7
2 3 4
3 5 6 4
```

Επιλέγοντας το υποσύνολο 1, 3, 5, 6, και 8 δε δημιουργείται κανένας επικίνδυνος συνδυασμός δημοσιονομικών μέτρων και δεν υπάρχει μεγαλύτερο ασφαλές υποσύνολο.

Αν υπάρχουν πολλές έγκυρες απαντήσεις με το ίδιο (μέγιστο) πλήθος μέτρων, τα προγράμματά σας μπορούν να επιστρέφουν οποιαδήποτε από αυτές. Επίσης, τα στοιχεία των υποσυνόλων μπορούν να εμφανίζονται με οποιαδήποτε σειρά. Παρατηρήστε ότι μεταξύ των διαφορετικών αριθμών τόσο της εισόδου όσο και της εξόδου πρέπει να υπάρχει ένα κενό (space). Το πρόγραμμα σε SML/NJ πρέπει να ορίζει μια συνάρτηση με όνομα `danger` η οποία να επιστρέφει μια λίστα από ακεραίους.

Περαιτέρω οδηγίες για τις ασκήσεις

- Μπορείτε να δουλέψετε σε ομάδες το πολύ δύο ατόμων, τόσο σε αυτή όσο και στις επόμενες σειρές ασκήσεων. Όμως, έχετε υπ' όψη σας ότι, αν δεν περάσετε το μάθημα φέτος, οι βαθμοί των προγραμματιστικών ασκήσεων κρατούνται μόνο για όσους δεν τις έκαναν σε ομάδα αλλά τις έκαναν μόνοι τους.

- 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 C πρέπει να είναι σε ένα αρχείο και να μπορούν να μεταγλωττιστούν χωρίς warnings με gcc (version 4.7.2) με μια εντολή της μορφής (π.χ. για την πρώτη άσκηση):

```
gcc -std=c99 -Wall -Werror -O3 -o happy yourfile.c
```

(Μπορείτε να υποθέσετε κάποια αντίστοιχη εντολή για 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 3.12.1. 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 άλλους είδους έξοδο διότι δε θα γίνουν δεκτά από το σύστημα στο οποίο θα υποβληθούν.