ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Εαρινό 2018

ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ Ι

Άσκηση 1

Καταληκτική ημερομηνία και ώρα ηλεκτρονικής υποβολής: 15/4/2018, 23:59:59

Ραντεβού στην αγορά (0.25+0.25 = 0.5 βαθμοί)

Στην αγορά της πόλης φτάνουν οι κάτοικοι των κοντινών χωριών για να κάνουν προμήθειες. Υπάρχουν N χωριά γύρω από την πόλη. Οι κάτοικοι του πρώτου χωριού πηγαίνουν στην πόλη κάθε x_1 μέρες για ανεφοδιασμό, αυτοί του δεύτερου χωριού κάθε x_2 μέρες, κ.ο.κ. Σήμερα έτυχε να συναντηθούν στην αγορά της πόλης οι κάτοικοι όλων των χωριών — αυτό συμβαίνει αρκετά σπάνια. Βρείτε μετά από πόσες μέρες θα συναντηθούν πάλι στην αγορά της πόλης οι κάτοικοι τουλάχιστον $N-1$ χωριών.

Η άσκηση σας ζητάει να γράψετε δύο προγράμματα (ένα σε C/C++ και ένα σε ML) τα οποία να διαβάζουν το πλήθος των χωριών N και τη συχνότητα επίσκεψης x_i των κατοίκων τους στην αγορά της πόλης και να βρίσκουν τον ελάχιστο αριθμό ημερών μετά από τις οποίες θα συναντηθούν οι κάτοικοι τουλάχιστον $N-1$ χωριών. Αν εκείνη τη μέρα λείπουν οι κάτοικοι ενός χωριού, το πρόγραμμά σας πρέπει επίσης να βρίσκει ποιο θα είναι αυτό το χωριό.

Τα στοιχεία εισόδου θα διαβάζονται από ένα αρχείο όπως φαίνεται στα παραδείγματα που ακολουθούν. Η πρώτη γραμμή του αρχείου περιέχει τον ακέραιο N ($1 \le N \le 1.000.000$). Η δεύτερη γραμμή περιέχει N θετικούς ακέραιους αριθμούς x_i, χωρισμένους μεταξύ τους με ένα κενό διάστημα. Κάθε αριθμός x_i ($1 \le x_i \le 1.000.000$) εκφράζει κάθε πόσες μέρες έρχονται στην αγορά της πόλης οι κάτοικοι του i-οστού χωριού. Θεωρήστε δεδομένο ότι οι κάτοικοι όλων των χωριών θα συναντηθούν και πάλι στην αγορά της πόλης το πολύ σε 10^{18} μέρες (τιμή πολύ μεγάλη για να αναπαρασταθεί με αριθμό των 32 bit).

Το πρόγραμμά σας πρέπει να εκτυπώνει μία γραμμή αποτελούμενη από ακριβώς δύο ακέραιους αριθμούς, χωρισμένους μεταξύ τους με ένα κενό διάστημα. Ο πρώτος είναι ο ζητούμενος αριθμός ημερών, μετά από τις οποίες οι κάτοικοι τουλάχιστον $N-1$ χωριών θα ξανασυναντηθούν στην αγορά της πόλης. Ο δεύτερος αριθμός είναι μηδέν (0) αν εκείνη τη μέρα θα συναντηθούν πάλι οι κάτοικοι όλων των χωριών. Διαφορετικά, θα είναι ο αύξων αριθμός του χωριού που οι κάτοικοί του θα λείπουν. (Η αρίθμηση των χωριών είναι από 1 έως και N.)

Παρακάτω δίνονται κάποια παραδείγματα σε C/C++ και σε ML.

Σε C/C++, MLton, ή σε OCaml

```
$ ./agora a1.txt
360 7

$ ./agora a2.txt
840 0

$ ./agora a3.txt
25845383485350 4
```

Σε SML/NJ

```
- agora "a1.txt";
360 7
val it = () : unit

- agora "a2.txt";
840 0
val it = () : unit

- agora "a3.txt";
25845383485350 4
val it = () : unit
```

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

```
$ cat a1.txt
10
1 2 3 4 5 6 7 8 9 10
```

```
$ cat a2.txt
9
10 14 15 30 21 5 40 4 8
```

```
$ cat a3.txt
5
25065 3575 12305 88590 1758
```

Για το πρώτο αρχείο εισόδου, μετά από 360 μέρες θα συναντηθούν οι κάτοικοι όλων των χωριών εκτός από το 7ο. Για το δεύτερο, μετά από 840 μέρες θα συναντηθούν ξανά οι κάτοικοι όλων των χωριών. Για το τρίτο, στο οποίο φαίνεται ότι οι κάτοικοι των χωριών είναι υπεραιωνόβιοι και επίσης δεν τρώνε πολύ, θα συναντηθούν οι κάτοικοι τεσσάρων από αυτά (εκτός του 4ου) μετά από 25,845,383,485,350 μέρες (περισσότερες από εικοσιπέντε τρισεκατομμύρια!).

Η συντέλεια του κόσμου (0.25+0.25 = 0.5 βαθμοί)

Δίνεται ένας διδιάστατος χάρτης του σύμπαντος, αποτελούμενος από *N* x *M* τετραγωνάκια (1 ≤ *N*, *M* ≤ 1000). Κάθε τετραγωνάκι του χάρτη περιέχει ένα από τα εξής σύμβολα:

- "**+**" Το τετράγωνο περιέχει ύλη.
- "**–**" Το τετράγωνο περιέχει αντιύλη.
- "**.**" (τελεία): Το τετράγωνο είναι κενό και μπορεί να καταληφθεί από ύλη ή αντιύλη.
- "**X**" (εμπόδιο): Το τετράγωνο δεν μπορεί να καταληφθεί ούτε από ύλη ούτε από αντιύλη.

Στο σύμπαν αυτό, οι δύο μορφές της ύλης διαστέλλονται. Κάθε χρονική στιγμή, η ύλη που βρίσκεται σε ένα τετράγωνο εξαπλώνεται στα γειτονικά αυτού τετράγωνα (αριστερά, δεξιά, πάνω και κάτω), αν αυτά δεν είναι εμπόδια. Το ίδιο κάνει και η αντιύλη. Και (προφανώς, τι επιστημονική φαντασία διαβάζετε;) αν κάποια χρονική στιγμή σε κάποιο τετράγωνο συνυπάρξουν ύλη και αντιύλη, αυτές θα εξουδετερωθούν και θα προκαλέσουν τη συντέλεια του κόσμου. Αυτό που πρέπει να βρείτε είναι:

- πότε θα συμβεί αυτό (αν δεν τη γλυτώσουμε τελικά), και
- πώς θα μοιάζει το σύμπαν τότε.

Η άσκηση σας ζητάει να γράψετε δύο προγράμματα (ένα σε C/C++ και ένα σε ML) τα οποία θα απαντούν στα παραπάνω ερωτήματα.

Η είσοδος του προγράμματός σας διαβάζεται από ένα αρχείο αποτελούμενο από *N* γραμμές, κάθε μία από τις οποίες περιέχει *M* σύμβολα. Το αρχείο αυτό αναπαριστά το χάρτη.

Παρακάτω δίνονται κάποια παραδείγματα σε C/C++ και σε ML. Έστω ότι τα αρχεία με τα δεδομένα εισόδου είναι τα εξής (όπως είπαμε, η εντολή **cat** είναι εντολή του Unix και από μόνη της δεν συνηθίζει να χρωματίζει χάρτες του σύμπαντος):

```
$ cat d1.txt
+.......–
```

```
$ cat d2.txt
+.......–
```

```
$ cat d3.txt
+..X..X...
....XX...–
```

```
$ cat d4.txt
+X..XX....–
.X..X..X–..
.X.........
...XX......
XXX.+X.....
..X.....XXX
...XXX..X–.
.–.....X...
```

Η έξοδος του προγράμματός σας πρέπει να είναι η ακόλουθη. Στην πρώτη γραμμή πρέπει να

τυπώνεται η χρονική στιγμή στην οποία θα επέλθει η συντέλεια του κόσμου και στις επόμενες γραμμές πρέπει να τυπώνεται ο χάρτης του σύμπαντος αυτή τη χρονική στιγμή. Αν η συντέλεια του κόσμου δεν πρόκειται να συμβεί (γιατί ύλη και αντιύλη δεν θα συναντηθούν), τότε στην πρώτη γραμμή πρέπει να τυπώνεται το μήνυμα “the world is saved” και οι επόμενες γραμμές πρέπει να περιέχουν την τελική μορφή του χάρτη. Στην τελική μορφή του χάρτη, τα τετράγωνα όπου συναντιούνται ύλη και αντιύλη πρέπει να σημειώνονται με “*****” (αστεράκι).

Σε C/C++, MLton, ή σε OCaml

```
$ ./doomsday d1.txt
4
++++*----

$ ./doomsday d2.txt
5
++++**----

$ ./doomsday d3.txt
the world is saved
+++X..X---
++++XX----

$ ./doomsday d4.txt
5
+X..XX-----
+X..X--X---
+X..-------
+++XX-*----
XXX++X**---
--X++++*XXX
---XXX++X--
------*X---
```

Σε SML/NJ

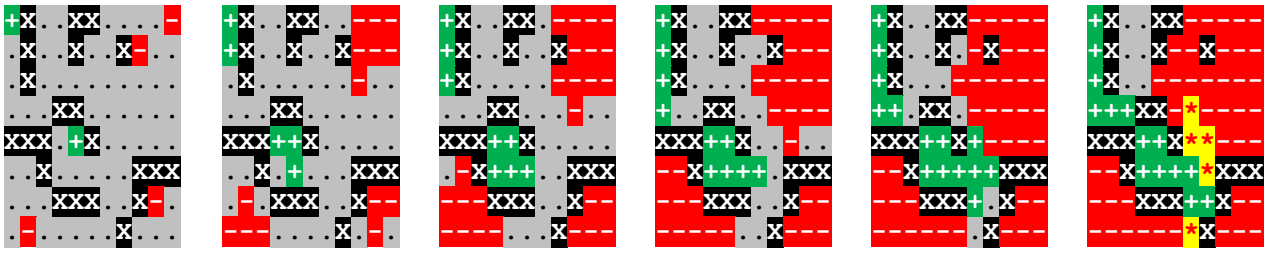
```
- doomsday "d1.txt";
4
++++*----
val it = () : unit

- doomsday "d2.txt";
5
++++**----
val it = () : unit

- doomsday "d3.txt";
the world is saved
+++X..X---
++++XX----
val it = () : unit

- doomsday "d4.txt";
5
+X..XX-----
+X..X--X---
+X..-------
+++XX-*----
XXX++X**---
--X++++*XXX
---XXX++X--
------*X---
val it = () : unit
```

Για το τελευταίο (και πιο ενδιαφέρον) παράδειγμα, δίνεται παρακάτω ο χάρτης του σύμπαντος για όλες τις χρονικές στιγμές από την αρχική (t = 0) μέχρι και τη συντέλεια του κόσμου (t = 5).



t = 0 t = 1 t = 2 t = 3 t = 4 t = 5

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. κώδικα ταξινόμησης, 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/C++ πρέπει να είναι σε ένα αρχείο και να μπορούν να μεταγλωττιστούν χωρίς warnings με gcc/g++ (version ≥ 4.9.2) με εντολές της μορφής, π.χ.

```
gcc –std=c99 -Wall –Werror -O3 –o agora agora.c
g++ –std=c++11 -Wall –Werror -O3 -o agora agora.cpp
```

- Τα προγράμματα σε ML πρέπει επίσης 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 σας επιτρέπει να επιλέξετε μεταξύ αυτών των διαλέκτων της ML.

- 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 σχετική ανακοίνωση για την ακριβή διαδικασία υποβολής όταν ανοίξει το σύστημα. Τα προγράμματά σας πρέπει να διαβάζουν την είσοδο όπως αναφέρεται και δεν πρέπει να έχουν κάποιου άλλους είδους έξοδο διότι δεν θα γίνουν δεκτά από το σύστημα στο οποίο θα υποβληθούν.