

Άσκηση 1

Καταληκτική ημερομηνία και ώρα ηλεκτρονικής υποβολής: 23/4/2021, 23:59:59

ΜΕΘ υπό διαρκή πίεση (0.25+0.25 = 0.5 βαθμοί)

Λόγω πανδημίας, τα συστήματα υγείας ανά τον κόσμο βρίσκονται υπό πίεση. Κάθε μέρα κάποιοι συνάνθρωποί μας εισάγονται στις ΜΕΘ (και καταλαμβάνουν μία κλίνη) και κάποιοι άλλοι είτε αναρρώνουν ή, δυστυχώς, καταλήγουν (και ελευθερώνουν μία κλίνη). Μελετάμε ένα διάστημα M συνεχόμενων ημερών (όπου $1 \leq M \leq 500.000$), σε κάθε μία από τις οποίες αντιστοιχούμε έναν ακέραιο αριθμό: τη διαφορά των κατειλημμένων κλινών ΜΕΘ από την προηγούμενη ημέρα.

Έστω ότι στη χώρα που μας ενδιαφέρει υπάρχουν N νοσοκομεία (όπου $1 \leq N \leq 1.000$). Θα λέμε ότι μια χρονική περίοδος K συνεχόμενων ημερών είναι «καλή» για τη χώρα, αν σε αυτή τη χρονική περίοδο συνολικά ελευθερώνεται (κατά μέσο όρο) τουλάχιστον μία κλίνη ανά νοσοκομείο ανά ημέρα. Για παράδειγμα, αν τα νοσοκομεία της χώρας είναι $N=3$ και σε μία περίοδο $K=4$ ημερών το άθροισμα των ημερήσιων διαφορών κατειλημμένων κλινών είναι -14 , τότε αυτή η περίοδος είναι καλή για τη χώρα (γιατί ο μέσος όρος κλινών που ελευθερώνονται ανά νοσοκομείο και ανά ημέρα είναι $14/12 \geq 1$). Αντίθετα, αν σε μία περίοδο $K=6$ ημερών το αντίστοιχο άθροισμα είναι -17 , τότε η περίοδος αυτή δε θεωρείται καλή (γιατί ο μέσος όρος κλινών που ελευθερώνονται ανά νοσοκομείο και ανά ημέρα είναι $17/18 < 1$).

Βοηθήστε την (ισόβια και ίσως υπεραιωνόβια, για τέτοιες τιμές του M που συζητάμε) πρόεδρο της χώρας να υπολογίσει το πλήθος των ημερών της μακρύτερης σε διάρκεια «καλής» περιόδου.

Η άσκηση σας ζητάει να γράψετε δύο προγράμματα (ένα σε C/C++ και ένα σε ML) τα οποία να διαβάζουν την είσοδο όπως φαίνεται παρακάτω και να τυπώνουν τη διάρκεια K σε ημέρες της μακρύτερης σε διάρκεια «καλής» περιόδου της χώρας.

Τα στοιχεία εισόδου θα διαβάζονται από ένα αρχείο όπως φαίνεται στα παραδείγματα που ακολουθούν. Η πρώτη γραμμή του αρχείου περιέχει δύο φυσικούς ακέραιους: M (μέρες) και N (νοσοκομεία). Η δεύτερη γραμμή του αρχείου περιέχει M ακεραίους (τη διαφορά των εισαγωγών μείον τις εξαγωγές από τις ΜΕΘ την αντίστοιχη ημέρα), χωρισμένους μεταξύ τους με ένα κενό διάστημα. Μπορείτε να υποθέσετε ότι τόσο οι αριθμοί εισόδου όσο και τα αλγεβρικά αθροίσματα για κάθε δυνατή χρονική δε θα υπερβαίνουν το $1.000.000.000$ κατ' απόλυτο τιμή.

Παρακάτω δίνεται κάποιο παράδειγμα σε C/C++ και σε ML. Με βάση τα δεδομένα στο αρχείο εισόδου, η μεγαλύτερη σε διάρκεια «καλή» περίοδος της χώρας ($-6 \ -12 \ 16 \ -15 \ -11$) κράτησε 5 ημέρες.

Σε C/C++, MLton, ή σε OCaml

```
$ ./longest f.txt
5
```

Σε SML/NJ

```
- longest "f.txt";
5
val it = () : unit
```

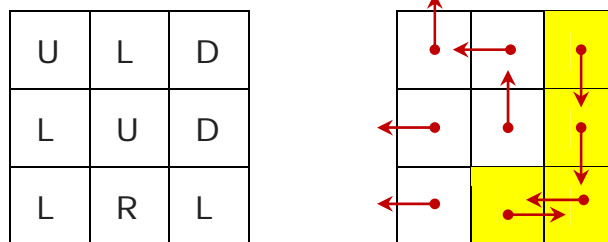
όπου το αρχείο με τα δεδομένα εισόδου είναι το εξής (η εντολή `cat` είναι εντολή του Unix):

```
$ cat f.txt
11 3
42 -10 8 1 11 -6 -12 16 -15 -11 13
```

Βρόχοι σε τηλεπαιχνίδια (0.25+0.25 = 0.5 βαθμοί)

Από τα λίγα πράγματα που γνωρίζουν ανάπτυξη μέσα στην πανδημία είναι όλων των ειδών τα χαζά τηλεπαιχνίδια. Σε ένα από αυτά, οι παίκτες τοποθετούνται σε ένα λαβύρινθο από όπου πρέπει να βγουν. Ο λαβύρινθος αποτελείται από $N \times M$ δωμάτια σε διάταξη ορθογωνίου. Σε κάθε δωμάτιο υπάρχει ένας γρίφος και, αν ο παίκτης τον λύσει, τότε ανοίγει μία πόρτα που οδηγεί σε ένα συγκεκριμένο γειτονικό δωμάτιο (πάνω, κάτω, αριστερά ή δεξιά). Αν ένα δωμάτιο βρίσκεται στο περίγραμμα του λαβύρινθου και ανοίξει μία πόρτα που οδηγεί προς τα έξω, τότε ο παίκτης βγαίνει και κερδίζει. Μιλάμε για υπερπαραγωγή!

Οι διαστάσεις του λαβύρινθου και η θέση της πόρτας εξόδου για κάθε δωμάτιο είναι γνωστές εξ αρχής. Σε κάθε δωμάτιο, η πόρτα είναι σημειωμένη με ένα από τα γράμματα "U" (up), "D" (down), "L" (left) ή "R" (right). Άρα, αν ένας παίκτης ξεκινήσει από κάποιο αρχικό δωμάτιο και λύσει κατά σειρά όλους τους γρίφους στα δωμάτια που επισκέπτεται, η διαδρομή που θα ακολουθήσει είναι συγκεκριμένη. Δυστυχώς όμως για τους παίκτες, είναι πιθανό οι πόρτες εξόδου να είναι έτσι τοποθετημένες ώστε από κάποια αρχικά δωμάτια να μην είναι δυνατή η έξοδος από το λαβύρινθο! Δείτε το παρακάτω παράδειγμα. Τα βέλη στο δεξιό μέρος του σχήματος δείχνουν την πορεία των παικτών από κάθε δωμάτιο.



Αν ένας παίκτης ξεκινήσει από κάποιο αρχικό δωμάτιο που είναι σημειωμένο με κίτρινο χρώμα, τότε θα κολλήσει σε μια κυκλική διαδρομή και δε θα καταφέρει να βγει ποτέ από το λαβύρινθο, όσους γρίφους και αν λύσει. Αντίθετα, από τα υπόλοιπα δωμάτια θα καταφέρει να βγει, αν φυσικά λύσει τους γρίφους που απαιτούνται.

Οι παραγωγοί του παιχνιδιού θέλουν να είναι δίκαιο το τηλεπαιχνίδι τους, δηλαδή τα αρχικά δωμάτια από τα οποία οι παίκτες δεν μπορούν να κερδίσουν όσους γρίφους και αν λύσουν να είναι λίγα. Βοηθήστε τους να τα μετρήσουν!

Τα στοιχεία εισόδου διαβάζονται από ένα αρχείο όπως φαίνεται στο παράδειγμα που ακολουθεί. Η πρώτη γραμμή του αρχείου έχει δύο ακέραιους N και M ($1 \leq N, M \leq 1.000$), τις διαστάσεις του λαβυρινθου (γραμμές και στήλες, αντίστοιχα). Κάθε μία από τις επόμενες N γραμμές παριστάνει μία γραμμή του λαβύρινθου και περιέχει ακριβώς M χαρακτήρες, κάθε ένας από τους οποίους είναι ένα από τα γράμματα "U", "D", "L" ή "R". Το γράμμα που αντιστοιχεί σε κάθε δωμάτιο συμβολίζει τη θέση της πόρτας εξόδου του δωματίου.

Η άσκηση σας ζητάει να γράψετε δύο προγράμματα (ένα σε C/C++ και ένα σε ML). Το κάθε πρόγραμμά σας θα πρέπει να διαβάζει το λαβύρινθο, να κάνει τους απαραίτητους υπολογισμούς και να τυπώνει έναν ακέραιο αριθμό: το πλήθος των δωματίων από τα οποία οι παίκτες δεν μπορούν να κερδίσουν, όσους γρίφους και αν λύσουν.

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

Σε C/C++, MLton, ή σε OCaml

```
$ ./loop_rooms maze1.txt
4

$ ./loop_rooms maze2.txt
2
```

Σε SML/NJ

```
- loop_rooms "maze1.txt";
4
val it = () : unit
- loop_rooms "maze2.txt";
2
val it = () : unit
```

όπου τα δύο αρχεία με τα δεδομένα εισόδου είναι τα εξής:

```
$ cat maze1.txt          $ cat maze2.txt
3 3                      2 2
ULD                      UL
LUD                      RL
LRL
```

Ο λαβύρινθος του αρχείου `maze1.txt` αντιστοιχεί στο παραπάνω σχήμα.

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 $\geq 8.3.0$) με εντολές της μορφής, π.χ.

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

- Τα προγράμματα σε ML πρέπει επίσης να είναι σε ένα αρχείο και να δουλεύουν σε SML/NJ $\geq v110.79$ ή σε MLton ≥ 20130715 ή σε Objective Caml version $\geq 4.05.0$. 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 άλλου είδους έξοδο διότι δεν θα γίνουν δεκτά από το σύστημα στο οποίο θα υποβληθούν.