

Άσκηση 1

Καταληκτική ημερομηνία και ώρα ηλεκτρονικής υποβολής: 14/4/2019, 23:59:59

Πολύχρωμη κορδέλα (0.25+0.25 = 0.5 βαθμοί)

Τη γάτα μου τη λένε Princess Arjumand. Της αρέσει να παίζει με μια πολύχρωμη κορδέλα. Η κορδέλα έχει μήκος N εκατοστά και κάθε εκατοστό της είναι χρωματισμένο με ένα χρώμα από K δυνατά χρώματα, αριθμημένα από το 1 έως και το K .



spoiler!

Στο παραπάνω παράδειγμα, η κορδέλα έχει μήκος $N=10$ εκατοστά και υπάρχουν $K=3$ χρώματα: το κίτρινο (1), το πορτοκαλί (2) και το πράσινο (3).

Όπως βλέπετε, κάθε εκατοστό της κορδέλας είναι χρωματισμένο με ένα από αυτά.

Η Arjumand έχει καλλιτεχνικές τάσεις και αρνείται να παίζει με την κορδέλα, αν δεν έχει όλα τα χρώματα από το 1 έως και το K . Αν όμως η κορδέλα είναι πολύ μακριά, τότε εύκολα μπερδεύεται και η Arjumand εκνευρίζεται.

Βοηθήστε με να βρω και να της κόψω το μικρότερο δυνατό κομμάτι της κορδέλας που να περιέχει όλα τα χρώματα!

Η άσκηση σας ζητάει να γράψετε δύο προγράμματα (ένα σε C/C++ και ένα σε ML) τα οποία να διαβάζουν το μήκος της κορδέλας, το πλήθος των χρωμάτων και τους κωδικούς των χρωμάτων με τα οποία είναι χρωματισμένο κάθε εκατοστό της κορδέλας, και να βρίσκουν το μήκος του ελάχιστου δυνατού κομματιού της κορδέλας που περιέχει όλα τα χρώματα.

Τα στοιχεία εισόδου θα διαβάζονται από ένα αρχείο όπως φαίνεται στα παραδείγματα που ακολουθούν. Η πρώτη γραμμή του αρχείου περιέχει δύο ακέραιους αριθμούς χωρισμένους μεταξύ τους με ένα κενό διάστημα: το μήκος της κορδέλας N ($1 \leq N \leq 1.000.000$) και το πλήθος των χρωμάτων K ($1 \leq K \leq 100.000$). Η δεύτερη γραμμή περιέχει ακριβώς N ακέραιους αριθμούς c_i ($1 \leq c_i \leq K$) χωρισμένους μεταξύ τους με ένα κενό διάστημα. Οι αριθμοί αυτοί είναι οι κωδικοί των χρωμάτων με τα οποία είναι χρωματισμένο κάθε εκατοστό της κορδέλας, από τη μία άκρη μέχρι την άλλη.

Το πρόγραμμά σας πρέπει να εκτυπώνει μία γραμμή αποτελούμενη από ακριβώς έναν ακέραιο αριθμό: το μήκος του μικρότερου δυνατού κομματιού της κορδέλας που περιέχει όλα τα χρώματα μεταξύ 1 και K . Αν δεν υπάρχει τέτοιο κομμάτι (γιατί στην αρχική κορδέλα δεν υπήρχαν όλα τα χρώματα), τότε στην έξοδο πρέπει να γράφεται ο αριθμός 0 (μηδέν).

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

Σε C/C++, MLton, ή σε OCaml

```
$ ./colors c1.txt  
4
```

```
$ ./colors c2.txt  
10
```

Σε SML/NJ

```
- colors "c1.txt";  
4  
val it = () : unit
```

```
- colors "c2.txt";  
10  
val it = () : unit
```

```

$ ./colors c3.txt          - colors "c3.txt";
0                          0
                           val it = () : unit

```

όπου τα αρχεία με τα δεδομένα εισόδου είναι τα εξής (η εντολή `cat` είναι εντολή του Unix και δεν έχει καμία συγγένεια με την `Arjumand`):

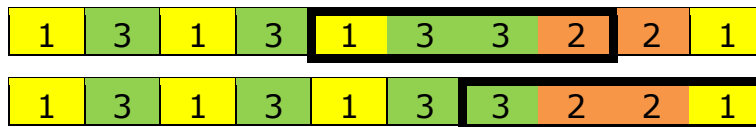
```

$ cat c1.txt                $ cat c2.txt
10 3                        10 4
1 3 1 3 1 3 3 2 2 1        2 1 4 4 4 1 1 1 4 3

$ cat c3.txt
10 5
1 2 2 5 1 3 1 2 1 3

```

Το πρώτο αρχείο εισόδου αντιστοιχεί στην κορδέλα του σχήματος της προηγούμενης σελίδας. Υπάρχουν δύο κομμάτια που περιέχουν όλα τα χρώματα και έχουν μήκος 4:



Δεν υπάρχει μικρότερο κομμάτι της κορδέλας που να περιέχει όλα τα χρώματα.

Σώσε τη γάτα! (0.25+0.25 = 0.5 βαθμοί)

Αφού βαρέθηκε να παίζει με την κορδέλα της, η `Arjumand` βγήκε βόλτα. Για κακή της τύχη, ο δρόμος την έφερε σε ένα σκοτεινό υπόγειο και, για ακόμα χειρότερη τύχη της, το υπόγειο είναι έτοιμο να πλημμυρίσει. Η `Arjumand` μισεί το νερό!

Δίνεται ένας διδιάστατος χάρτης του υπογείου, αποτελούμενος από $N \times M$ τετραγωνάκια ($1 \leq N, M \leq 1000$). Κάθε τετραγωνάκι του χάρτη περιέχει ένα από τα εξής σύμβολα:

- “**A**” Η αρχική θέση της `Arjumand`.
- “**W**” Το τετράγωνο περιέχει ένα σπασμένο σωλήνα νερού.
- “.” (τελεία): Το τετράγωνο είναι αρχικά κενό.
- “**X**” (εμπόδιο): Σε αυτό το τετράγωνο δεν μπορεί να πάει ούτε η `Arjumand` ούτε το νερό.

Κάθε χρονική στιγμή, το νερό που βγαίνει από τους σπασμένους σωλήνες εξαπλώνεται στα γειτονικά τετράγωνα (αριστερά, δεξιά, πάνω και κάτω), αν αυτά δεν είναι εμπόδια, με αποτέλεσμα ο χάρτης σιγά-σιγά να πλημμυρίζει. Το νερό κινείται με ρυθμό ένα τετράγωνο ανά μονάδα χρόνου. Ευτυχώς για εκείνη, και η `Arjumand` μπορεί να κινείται στα γειτονικά της τετράγωνα, με ρυθμό επίσης ένα τετράγωνο ανά μονάδα χρόνου. Αυτό που πρέπει να βρείτε είναι:

- ποια είναι η αργότερη χρονική στιγμή που μπορούμε να σώσουμε την `Arjumand`, και
- σε ποιο τετράγωνο του χάρτη θα τη βάλουμε να πάει για να τη σώσουμε.

Η απάντηση στη δεύτερη ερώτηση είναι μία συμβολοσειρά η οποία περιγράφει την ακολουθία κινήσεων που θα κάνει η `Arjumand` ώστε να βρεθεί στο τετράγωνο από το οποίο θα τη σώσουμε. Οι δυνατές κινήσεις παριστάνονται με τα σύμβολα:

- “**R**” Μετακίνηση ένα τετράγωνο δεξιά στο χάρτη.
- “**L**” Μετακίνηση ένα τετράγωνο αριστερά στο χάρτη.
- “**U**” Μετακίνηση ένα τετράγωνο προς τα πάνω στο χάρτη.
- “**D**” Μετακίνηση ένα τετράγωνο προς τα κάτω στο χάρτη.



spoiler!

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 υπογείων):

```
$ cat a1.txt
```

```
A...
....
```

```
$ cat a2.txt
```

```
...W...
.A.XX.
XX.X.
.....X
```

```
$ cat a3.txt
```

```
.....
.X.X.
.A.X.X.
.X.W.XX
.XX.X.
.....X
```

```
$ cat a4.txt
```

```
WX.XX...W
.X.X.XW.
.X.
.XX
XXX.WX.
.X.XXXX
.XX.XX
.A.X.X.
```

Η έξοδος του προγράμματός σας πρέπει να είναι η ακόλουθη. Στην πρώτη γραμμή πρέπει να τυπώνεται η αργότερη χρονική στιγμή κατά την οποία θα πρέπει να σώσουμε την *Arjmand*. Αν η *Arjmand* είναι ασφαλής και μπορούμε να τη σώσουμε οποιαδήποτε στιγμή, στην πρώτη γραμμή πρέπει να τυπώνεται η λέξη "infinity". Διαφορετικά, η πρώτη γραμμή πρέπει να περιέχει ακριβώς έναν φυσικό αριθμό. Στη δεύτερη γραμμή πρέπει να τυπώνεται η συμβολοσειρά που αντιστοιχεί στην ακολουθία κινήσεων της *Arjmand*. Αν αυτή η ακολουθία είναι κενή, δηλαδή αν πρόκειται να σώσουμε την *Arjmand* από το τετράγωνο στο οποίο βρίσκεται, τότε στη δεύτερη γραμμή πρέπει να τυπώνεται η λέξη "stay". Αν υπάρχουν πολλές διαφορετικές λύσεις, τότε επιλέγουμε μία ακολουθία κινήσεων που την οδηγεί στο τετράγωνο με τις μικρότερες λεξικογραφικά συντεταγμένες (πρώτα γραμμή και μετά στήλη) και αν υπάρχουν πολλές ακολουθίες κινήσεων που οδηγούν σε αυτό το τετράγωνο, τότε επιλέγουμε τη μικρότερη σε πλήθος κινήσεων και (μεταξύ ισομηκών) τη λεξικογραφικά μικρότερη.

Σε C/C++, MLton, ή σε OCaml

```
$ ./savethecat a1.txt
infinity
stay
```

```
$ ./savethecat a2.txt
5
RDDLL
```

```
$ ./savethecat a3.txt
15
DDRRRURR
```

```
$ ./savethecat a4.txt
infinity
LLUU
```

Σε SML/NJ

```
- savethecat "a1.txt";
infinity
stay
val it = () : unit
```

```
- savethecat "a2.txt";
5
RDDLL
val it = () : unit
```

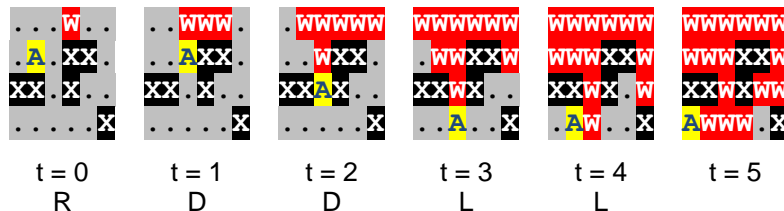
```
- savethecat "a3.txt";
15
DDRRRURR
val it = () : unit
```

```
- savethecat "a4.txt";
infinity
LLUU
val it = () : unit
```

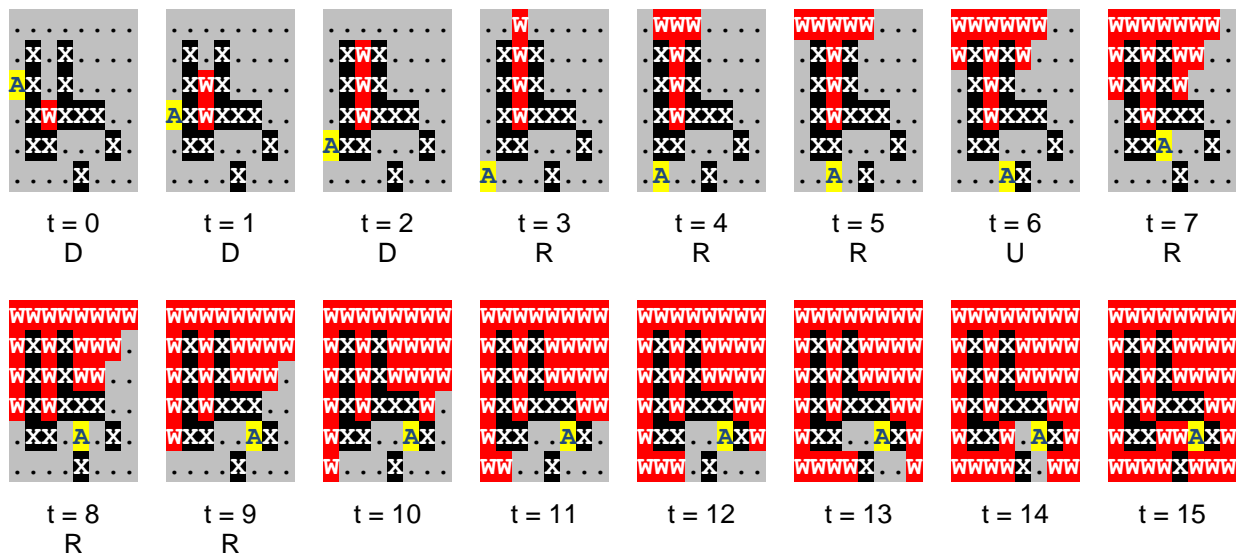
Στο πρώτο παράδειγμα η *Arjmand* είναι ασφαλής γιατί δεν υπάρχει σπασμένος σωλήνας. Βρίσκεται ήδη στο τετράγωνο με τις ελάχιστες λεξικογραφικά συντεταγμένες (στην πάνω αριστερή

γωνία του χάρτη), άρα δε χρειάζεται να κινηθεί. Ομοίως, στο τέταρτο παράδειγμα είναι επίσης ασφαλής γιατί βρίσκεται σε ένα χώρο του χάρτη που δε θα πλημμυρίσει. Όμως, πρέπει να μετακινηθεί στην πιο πάνω-αριστερά γωνία αυτού του χώρου και για το σκοπό αυτό θα κάνει τις κινήσεις LLUU (θα μπορούσε να βρεθεί εκεί και με τις κινήσεις ULUL αλλά η πρώτη συμβολοσειρά είναι λεξικογραφικά μικρότερη).

Για το δεύτερο παράδειγμα, δίνεται παρακάτω ο χάρτης του υπογείου για όλες τις χρονικές στιγμές από την αρχική ($t = 0$) μέχρι τη σωτηρία της Arjmand ($t = 5$). Προσέξτε ότι η Arjmand θα μπορούσε να πάει και στο άλλο τετράγωνο που είναι στεγνό τη χρονική στιγμή $t = 5$, αλλά οι συντεταγμένες εκείνου είναι λεξικογραφικά μεγαλύτερες.



Για το τρίτο (και πιο ενδιαφέρον) παράδειγμα, δίνεται πάλι παρακάτω ο χάρτης του υπογείου για όλες τις χρονικές στιγμές μέχρι τη σωτηρία της Arjmand.



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 μοιράζετε 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 $\geq 4.9.2$) με εντολές της μορφής, π.χ.

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

- Τα προγράμματα σε ML πρέπει επίσης να είναι σε ένα αρχείο και να δουλεύουν σε SML/NJ $\geq v110.76$ ή σε MLton ≥ 20100608 ή σε Objective Caml version $\geq 4.01.0$. 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 άλλου είδους έξοδο διότι δεν θα γίνουν δεκτά από το σύστημα στο οποίο θα υποβληθούν.