

## Problema 1 – Perle

50 puncte

Granița nu se trece ușor. Asta pentru că Balaurul Arhired (mare pasionat de informatică) nu lasă pe nimeni să treacă decât după ce răspunde la niște întrebări...

În acea țară există 3 tipuri de perle normale (le vom nota cu 1, 2 și 3) și 3 tipuri de perle magice (le vom nota cu A, B și C). Perlele magice sunt deosebite prin faptul că se pot transforma în alte perle (una sau mai multe, normale sau magice).

Perla magică de tipul A se poate transforma în orice perlă normală (una singură).

Perla magică de tipul B se poate transforma într-o perlă normală de tipul 2 și una magică de tipul B, sau într-o perlă normală de tipul 1, una magică de tipul A, una normală de tipul 3, una magică de tipul A și una magică de tipul C.

Perla magică de tipul C se poate transforma într-o perlă normală de tipul 2 sau într-o perlă normală de tipul 3, una magică de tipul B și una magică de tipul C sau într-o perlă normală de tipul 1, una normală de tipul 2 și una magică de tipul A.

Ca să rezumăm cele de mai sus putem scrie:

A -> 1 | 2 | 3

B -> 2B | 1A3AC

C -> 2 | 3BC | 12A

Balaurul Arhired ne lasă la început să ne alegem o perlă magică (una singură), iar apoi folosind numai transformările de mai sus trebuie să obținem un anumit șir de perle normale. Când o perlă magică se transformă, perlele din stânga și din dreapta ei rămân la fel (și în aceeași ordine). De asemenea ordinea perlelor rezultate din transformare este chiar cea prezentată mai sus.

De exemplu, dacă balaurul ne cere să facem șirul de perle 21132123, putem alege o perlă magică de tipul B și următorul șir de transformări: B -> 2B -> 21A3AC -> 21A3A12A -> 21132123.

Întrucât Balaurul nu are prea multă răbdare, el nu ne cere decât să spunem dacă se poate sau nu obține șirul respectiv de perle.

### Cerință

Să se determine pentru fiecare șir de intrare dacă se poate obține prin transformările de mai sus sau nu (alegând orice primă perlă magică, la fiecare șir).

### Date de intrare

Fișierul de intrare **perle.in** are următoarea structură:

- pe prima linie numărul **N**, reprezentând numărul de șiruri din fișierul de intrare
- urmează **N** linii; a **i**-a linie dintre cele **N** descrie șirul **i**, printr-o succesiune de numere naturale despărțite de câte un spațiu. Primul număr reprezintă lungimea șirului **L<sub>i</sub>**, iar următoarele **L<sub>i</sub>** numere sunt tipurile de perle normale, în ordine, de la stânga la dreapta.

### Date de ieșire

Fișierul **perle.out** va conține **N** linii. Pe linia **i** se va scrie un singur număr **1** sau **0** (**1** dacă se poate obține șirul respectiv (al **i**-lea) și **0** dacă nu se poate).

### Restricții

- $0 < N < 11$
- $0 < L_i < 10001$ , pentru oricare **i**

### Exemplu

perle.in	perle.out
3	1
8 2 1 1 3 2 1 2 3	0
2 2 2	1
1 3	

Timp maxim de execuție/fișier test: 1 secundă.

## Problema 2 – Romeo și Julieta

50 puncte

În ultima ecranizare a celebrei piese shakespeariene Romeo și Julieta trăiesc într-un oraș modern, comunică prin e-mail și chiar învață să programeze. Într-o secvență tulburătoare sunt prezentate frământările interioare ale celor doi eroi încercând fără succes să scrie un program care să determine un punct optim de întâlnire.

Ei au analizat harta orașului și au reprezentat-o sub forma unei matrice cu  $n$  linii și  $m$  coloane, în matrice fiind marcate cu spațiu zonele prin care se poate trece (străzi lipsite de pericole) și cu  $X$  zonele prin care nu se poate trece. De asemenea, în matrice au marcat cu  $R$  locul în care se află locuința lui Romeo, iar cu  $J$  locul în care se află locuința Julietei.

Ei se pot deplasa numai prin zonele care sunt marcate cu spațiu, din poziția curentă în oricare dintre cele 8 poziții învecinate (pe orizontală, verticală sau diagonale).

Cum lui Romeo nu îi place să aștepte și nici să se lase așteptat n-ar fi tocmai bine, ei au hotărât că trebuie să aleagă un punct de întâlnire în care atât Romeo, cât și Julieta să poată ajunge în același timp, plecând de acasă. Fiindcă la întâlniri amândoi vin într-un suflet, ei estimează timpul necesar pentru a ajunge la întâlnire prin numărul de elemente din matrice care constituie drumul cel mai scurt de acasă până la punctul de întâlnire. Și cum probabil există mai multe puncte de întâlnire posibile, ei vor să îl aleagă pe cel în care timpul necesar pentru a ajunge la punctul de întâlnire este minim.

### Cerință

Scrieți un program care să determine o poziție pe hartă la care Romeo și Julieta pot să ajungă în același timp. Dacă există mai multe soluții, programul trebuie să determine o soluție pentru care timpul este minim.

### Date de intrare

Fișierul de intrare `rj.in` conține:

- pe prima linie numerele naturale  $N$   $M$ , care reprezintă numărul de linii și respectiv de coloane ale matricei, separate prin spațiu;
- pe fiecare dintre următoarele  $N$  linii se află  $M$  caractere (care pot fi doar  $R$ ,  $J$ ,  $X$  sau spațiu) reprezentând matricea.

### Date de ieșire

Fișierul de ieșire `rj.out` va conține o singură linie pe care sunt scrise trei numere naturale separate prin câte un spațiu  $t_{min}$   $x$   $y$ , având semnificația:

- $x$   $y$  reprezintă punctul de întâlnire ( $x$  – numărul liniei,  $y$  – numărul coloanei);
- $t_{min}$  este timpul minim în care Romeo (respectiv Julieta) ajunge la punctul de întâlnire.

### Restricții și precizări

$1 < N, M < 101$

Liniile și coloanele matricei sunt numerotate începând cu 1.

Pentru datele de test există întotdeauna soluție.

### Exemple

<code>rj.in</code>	<code>rj.out</code>	<i>Explicație</i>
5 8	4 4 4	Traseul lui Romeo poate fi: (1, 3), (2, 4), (3, 4), (4, 4)
XXX XXX		Deci timpul necesar lui Romeo pentru a ajunge de acasă la punctul de întâlnire este 4.
X X X		Traseul Julietei poate fi:
J X X X		(3, 1), (4, 2), (4, 3), (4, 5).
XX		Timpul necesar Julietei pentru a ajunge de acasă la punctul de întâlnire este de asemenea 4.
XXX XXXX		În plus 4, este punctul cel mai apropiat de ei cu această proprietate.

**Timp maxim de execuție/test: 1 secundă.**