

lacuri

100 puncte

Fișiere sursă: **lacuri.cpp**, **lacuri.c**, **lacuri.pas**

Pe un teren de formă pătrată sunt zone de uscat și lacuri. Harta terenului este memorată într-un tablou bidimensional $n \times n$ cu valori **1** (apă) sau **0** (uscat). Liniile sunt numerotate de la **1** la **n**, de sus în jos și coloanele de la **1** la **n**, de la stânga la dreapta. Fiecare lac este înconjurat de o suprafață de teren uscat. Excepție fac doar lacurile situate la marginea terenului care sunt înconjurate de uscat doar prin interiorul terenului nu și prin afara acestuia.

Cerință

Se dorește să se traverseze terenul **pe uscat**, de la poziția **(1,1)** la poziția **(n,n)**, pe un drum de lungime minimă, mergând pas cu pas. La un pas, se ajunge de la un pătrățel la altul învecinat cu el spre nord, est, sud sau vest. Să se scrie un program care:

- Determină numărul lacurilor care sunt de formă pătrată și în același timp sunt „pline de 1”.
- În cazul în care toate lacurile sunt de formă pătrată și în același timp „pline de 1”, determinați un drum cu proprietatea de mai sus.

Date de intrare

Fișierul de intrare **lacuri.in** are pe prima linie numărul **n**, iar pe următoarele **n** linii este harta terenului (fiecare linie are **n** valori **0** sau **1**, separate de câte un spațiu).

Date de ieșire

Fișierul de ieșire **lacuri.out** va conține:

- pe prima linie: numărul de lacuri ce sunt de formă pătrată și în același timp „pline de 1”;
- în cazul în care toate lacurile sunt de formă pătrată și în același timp „pline de 1”, urmează liniile ce descriu drumul de lungime minimă ales. Fiecare linie din fișier conține câte o pereche de coordonate ale pozițiilor succesive prin care trece drumul (linia și coloana, separate cu un spațiu), începând cu **(1,1)** și terminând cu **(n,n)**.

Restricții și precizări

- $2 \leq n \leq 100$
- Pozițiile **(1,1)** și **(n,n)** sunt sigur libere (cu valoarea **0**).
- Dacă există mai multe soluții se va afișa oricare dintre ele.
- Pot fi cel mult 100 de lacuri și cel puțin unul; dacă un lac este de formă pătrată, atunci $1 \leq \text{latura} \leq n-1$.
- Indiferent de forma lor**, lacurile sunt sigur „izolate”, adică nu se „ating” deloc de alt lac. De exemplu, dacă un lac conține poziția **(3,3)**, atunci un alt lac nu poate conține vreuna din pozițiile învecinate cu **(3,3)**, adică: **(2,3)**, **(2,4)**, **(3,4)**, **(4,4)**, **(4,3)**, **(4,2)**, **(3,2)** și **(2,2)**.
- Pentru cerința a) se acordă 30% din punctaj, iar pentru cerința b) se acordă 70% din punctaj.

Exemplu

lacuri.in	lacuri.out	Explicație
6 0 0 0 0 0 0 0 1 0 1 1 1 0 0 0 1 1 1 0 0 0 1 1 1 1 1 0 0 0 0 1 1 0 0 1 0	4 1 1 1 2 1 3 2 3 3 3 4 3 5 3 5 4 5 5 5 6 6 6	<p>a) Prima linie conține 4 (sunt 4 lacuri de formă pătrată și „pline de 1”)</p> <p>b) Deoarece toate cele 4 lacuri sunt de formă pătrată și „pline de 1”, se scrie și drumul ales: de la (1,1), (1,2), (1,3), (2,3), (3,3), ..., (6,6).</p> <p>Observații</p> <p>1) Dacă în poziția (3,5) ar fi fost un 0, atunci lacul cu latura 3 nu ar mai fi fost „plin de 1” și atunci prima linie ar fi conținut doar valoarea 3 (ar fi fost doar 3 lacuri pătrate și „pline de 1”).</p> <p>2) În exemplul inițial, dacă în poziția (6,1) ar fi fost valoarea 0, atunci nu ar fi fost toate lacurile pătrate (cel din stânga-jos nu ar fi fost pătrat) și s-ar fi afișat doar un 3 în fișierul de ieșire.</p> <p>3) În exemplul inițial, dacă în poziția (5,2) ar fi fost valoarea 0, atunci s-ar fi afișat doar un 3, pentru că lacul din stânga-jos nu ar fi un lac pătrat și „plin de 1”.</p>

Timp maxim de execuție/test: 0.25 secunde