Color — 100 puncte

Fişiere sursă: `color.pas` sau `color.c` sau `color.cpp`

Ion şi Vasile joacă un joc. Ei au la dispoziție un arbore binar strict (adică fiecare nod are **0** sau **2** fii) cu **N** noduri, numerotate de la **1** la **N** (nodul numerotat cu **1** este rădăcina arborelui). Inițial, toate nodurile sunt colorate în alb. Jucătorii vor efectua mutări alternativ, iar jucătorul aflat la mutare va colora în negru un nod colorat în alb. Ion efectuează prima mutare şi poate colora în negru orice nod al arborelui. Considerând că ultimul nod colorat de unul dintre jucători este **P**, jucătorul care urmează la mutare poate colora în negru unul din următoarele noduri (dacă nu au fost deja colorate în negru) :

- unul din cei **2** fii ai lui **P** (dacă **P** nu este frunză în arbore)
- tatăl lui **P** (dacă **P** nu este rădăcina arborelui)

Jocul continuă până când unul dintre jucători nu mai poate efectua nici o mutare. Atunci, jucătorul care a efectuat ultima mutare este considerat câştigător.

Cerință

Considerând că ambii jucători joacă optim, determinați toate nodurile din arbore pe care le poate colora Ion la prima mutare, astfel încât să fie sigur de victorie.

Date de intrare

Prima linie a fişierului **`color.in`** conține numărul întreg **N**, reprezentând numărul de noduri din arbore. Următoarele **N-1** linii conțin câte două numere întregi separate printr-un spațiu, **a** şi **b**, având semnificația că **a** este tatăl lui **b**.

Date de ieşire

Pe prima linie a fişierului **`color.out`** veți afişa numărul întreg **M**, reprezentând numărul de noduri pe care le poate colora Ion la prima mutare, astfel încât să fie sigur de victorie. Pe următoarea linie veți afişa numerele acestor noduri, în ordine crescătoare.

Restricții şi precizări

- **`1 <= N <= 16 000`**, **N** impar
- **40%** din teste vor avea **`N <= 1 000`**

Exemplu

color.in	color.out
9 1 2 1 3 2 4 2 5 4 6 4 7 3 8 3 9	6 1 5 6 7 8 9

Timp maxim de executare/test: 0.2 secunde pentru Linux şi 0.3 secunde pentru Windows

Magic — 100 puncte

Fişiere sursă: `magic.pas` sau `magic.c` sau `magic.cpp`

Misopan şi Trofonaced sunt doi eroi care vor să-şi unească forțele în lupta împotriva răului. Regatul este reprezentat printr-o matrice dreptunghiulară de **`N`** linii şi **`M`** coloane. Fiecare element al matricei corespunde unei bucăți de teren uscat sau mlăştinos. Cei doi eroi nu se vor aventura în părțile mlăştinoase ale regatului – se vor deplasa numai pe uscat. Ei se pot muta dintr-o poziție a matricei în una din cele **`4`** poziții vecine pe orizontală sau pe verticală, dacă această poziție corespunde unei zone de uscat. Unele poziții de uscat pot fi transformate prin vrajă în mlaştină.

Cerință

Ajutați un vrăjitor malefic să aleagă un număr minim de poziții ”transformabile“, prin schimbarea cărora cei doi eroi să nu se poată întâlni (să nu existe un drum pe uscat între cei doi).

Date de intrare

Prima linie a fişierului **`magic.in`** conține două numere întregi **`N`** şi **`M`** reprezentând numărul de linii, respectiv de coloane ale matricei. Următoarele **`N`** linii conțin câte **`M`** caractere cu următoarea semnificație:

- **`.`** pentru o poziție uscată
- **`x`**(mic) pentru una mlăştinoasă
- **`*`** pentru una uscată ”transformabilă“ în una mlăştinoasă de către vrăjitor
- **`M`** pentru poziția eroului Misopan
- **`T`** pentru poziția eroului Trofonaced

Date de ieşire

Pe prima linie a fişierului **`magic.out`** se scrie numărul întreg **`R`**, reprezentând numărul minim de poziții care trebuie transformate. Pe următoarele **`R`** linii vor apărea câte 2 numere, reprezentând pozițiile alese. Primul număr va fi linia (între **`1`** şi **`N`**), iar al doilea va fi coloana (între **`1`** şi **`M`**).

Restricții şi precizări

- **`1 <= N, M <= 50`**
- **`R`** (rezultatul afişat) poate fi **`0`**
- Pe testele date va exista întotdeauna soluție
- Se garantează că în toată matricea caracterele **`M`**, respectiv **`T`** vor apărea fiecare exact o dată
- Pozițiile eroilor sunt implicit zone de uscat care nu pot fi transformate de vrăjitor

Exemplu

magic.in	magic.out
4 4 MxxT .x*. .**. **x.	1 3 3

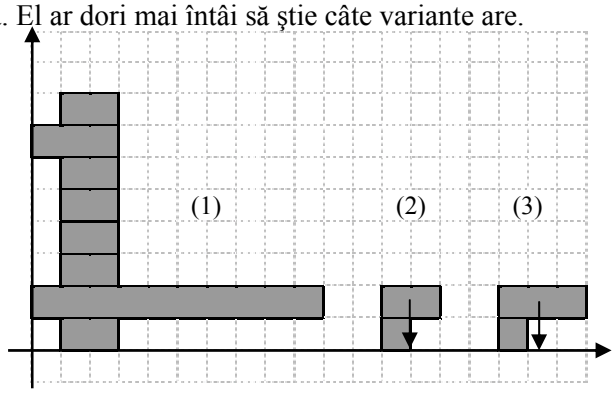
Timp maxim de executare/test: 0.6 secunde pentru Linux şi 1.8 secunde pentru Windows

Turnuri — 100 puncte

Fişiere sursă: `turnuri.pas` sau `turnuri.c` sau `turnuri.cpp`

Renumitul arhitect Prăbuşilă doreşte să construiască unul din cele mai interesante turnuri de pe planetă. Acest turn, în mod cu totul deosebit, va avea etaje de diverse lăţimi, între 1 şi 100, numere întregi.

Prăbuşilă s-a hotărât deja ce dimensiune va avea fiecare din etajele turnului, dar nu şi cum să le aşeze pe orizontală. El ar dori mai întâi să ştie câte variante are.



Etajele pot fi aşezate la coordonate întregi şi va trebui ca un astfel de turn să nu se dărâme.

- Condiția pentru ca un turn să fie stabil este ca la fiecare etaj perpendiculara coborâtă din centrul de greutate al grupului etajelor superioare să cadă strict în interiorul acelui etaj (nu are voie să fie pe margini sau în afară - de ex. turnurile 2 şi 3 sunt instabile).
- Centrul de greutate al unui etaj se află la mijlocul etajului respectiv.
- Centrul de greutate al unui grup de etaje are drept coordonată x (orizontală) media coordonatelor x ale centrelor de greutate ale etajelor componente. (Etajele au mase egale, indiferent de cât de late sunt).

În exemplul 1, etajul din vârf are coordonata x a centrului de greutate 2, iar grupul celor 2 etaje din vârf are centrul de greutate la coordonata x=1.75 (media aritmetică intre 2 şi 1.5)
Se observă în figura 1 că, deşi perpendiculara din centrul de greutate al etajului 2 cade în afara etajului 1, totuşi turnul este stabil, deoarece perpendiculara din centrul de greutate al grupului format din etajele 2–8 cade strict în interiorul etajului 1.

Cerință

Să se afle câte turnuri stabile există.

Date de intrare

Fişierul de intrare **`turnuri.in`** conține pe o singură linie lista de numere naturale separate prin câte un spațiu, numere reprezentând lățimile etajelor turnului, **începând cu cel mai de sus**. Lista se termină cu un 0.

Date de ieşire

Fişierul de ieşire **`turnuri.out`** conține numărul de turnuri.

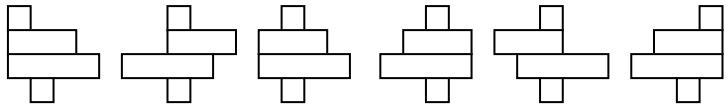
Restricții şi precizări:

- numărul maxim de turnuri nu va depăşi 2 miliarde
- numărul maxim de etaje ale unui turn este 200
- lățimea maximă a unui etaj este 100

Exemplu:

`turnuri.in`	**`turnuri.out`**
`1 3 4 1 0`	`6`

Cele 6 variante sunt:



Timp maxim de executare/test: 0.2 secunde pentru Linux şi 0.2 secunde pentru Windows