Problema FOO
Campionat de fotbal
Day 2, fişiere sursă foo.*
Memorie disponibilă: 32 MB. Timp maxim de execuție: 3s.

Într-un campionat de fotbal sunt **n** echipe (vom presupune că **n** este număr par). Pe parcursul unui sezon fiecare echipă joacă cu fiecare dintre celelalte echipe, exact o dată. Sezonul constă în **n**-1 etape. Fiecare două echipe se întâlnesc într-un sezon o singură dată.
Este de dorit ca fiecare echipă să joace meciurile consecutive pe stadioane diferite, adică un meci pe stadionul propriu, iar meciul următor în deplasare ş.a.m.d. Din nefericire nu este posibilă o planificare, astfel încât să nu existe echipe care joacă două meciuri consecutive acasă sau două meciuri consecutive în deplasare. Când se face o planificare a meciurilor dintr-un sezon, numărul acestor situații trebuie minimizat. (De exemplu, dacă o echipă joacă în deplasare, apoi de 4 ori acasă, apoi în deplasare, se numără ca **trei**, situațiile în care a jucat două meciuri consecutive numai acasă sau numai în deplasare)

Programul vostru trebuie să minimizeze numărul situațiilor în care o echipă joacă două meciuri consecutive acasă sau două meciuri consecutive în deplasare şi să construiască o planificare a meciurilor sezonului corespunzătoare acestei minimizări.

Sezonul constă în **n**-1 etape. O etapă constă în **n**/2 meciuri – fiecare echipă jucând exact un meci într-o etapă. Sunt în total **n**(**n**-1)/2 meciuri în întreg sezonul. Orice meci se joacă pe unul din stadioanele celor două echipe, care joacă respectivul meci (nu se joacă pe stadioane neutre). Numărul total al situațiilor în care o echipă joacă două meciuri consecutive acasă sau două meciuri consecutive în deplasare trebuie să fie minim.

Cerință
Scrieți un program, care:

- citeşte de la intrarea standard numărul de echipe
- calculează minimul numărului de situații în care o echipă joacă două meciuri consecutive acasă sau două meciuri consecutive în deplasare şi construieşte o planificare corespunzătoare acestui minim a jocurilor.
- scrie rezultatul la ieşirea standard

Date de intrare
Prima linie şi singura linie a intrării standard conține un număr întreg par **n** ($2 \le \mathbf{n} \le 1000$) – numărul de echipe.

Date de ieşire
Prima linie a ieşirii standard va conține un singur număr întreg –numărul minim de situații în care o echipă joacă două meciuri consecutive acasă sau două meciuri consecutive în deplasare. Următoarele **n**-1 linii vor conține o planificare a jocurilor: linia **k**+1 va conține o descriere a etapei cu numărul de ordine **k**.

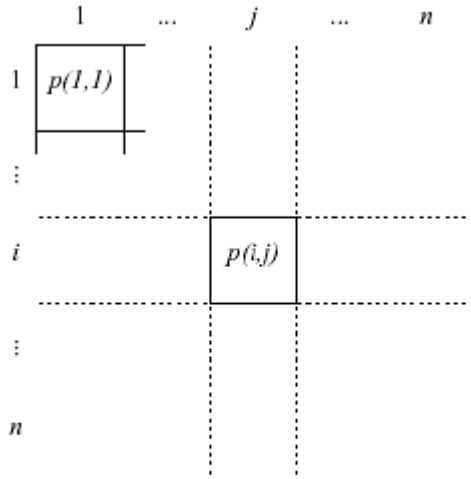
Descrierea unei etape constă din **n** numere diferite $\mathbf{d_1}$, $\mathbf{d_2}$, ..., $\mathbf{d_n}$ din mulțimea {1, 2, ..., **n**}, separate printr-un singur spațiu. Perechile $\mathbf{d_{2i-1}}$, $\mathbf{d_{2i}}$ (pentru **i**=1, 2, ..., **n**/2) reprezintă un meci între echipele $\mathbf{d_{2i-1}}$ şi $\mathbf{d_{2i}}$, în care echipa $\mathbf{d_{2i-1}}$ joacă acasă, iar echipa $\mathbf{d_{2i}}$ joacă în deplasare.

Exemplu:

Pentru datele de intrare:	Răspunsul corect este:
4	2 1 2 3 4 4 1 2 3 1 3 4 2

Problema PUZ
Puzzel
Day 2, fişiere sursă puz.*
Memorie disponibilă: 32 MB. Timp maxim de execuţie: 0.3 s.

Regele din Byteland a primit cadou un puzzel. Puzzelul constă într-o tablă de dimensiune **n**×**n**. Celula de pe linia **i** şi coloana **j** are coordonatele (**i**,**j**) şi conţine o piesă cu un număr $1 \le p(i,j) \le n^2$. Fiecare din numerele 1, 2, …, $\mathbf{n}^2$ apare in exact o celulă.



Pentru a rezolva puzzelul trebuie să aşezaţi numerele în ordine astfel încât fiecare celulă (**i**,**j**) să conţină numărul **i**+(**j**-1)***n**.
Mutările permise sunt:

- O permutare circulară a celulelor unei linii la dreapta cu un anumit număr de celule
- O permutare circulară a celulelor unei coloane în jos cu un anumit număr de celule

Regele din Byteland a reuşit să rezolve puzzelul, dar el nu este sigur dacă l-ar putea rezolva pornind dintr-o configuraţie iniţială diferită. Ajutaţi regele să rezolve această problemă.

Cerinţă
Scrieţi un program, care:

- citeşte de la intrarea standard descrierea configuraţiei iniţiale a puzzelului
- determină dacă celulele de pe tablă pot fi puse în ordine folosind doar mutările de mai sus. Dacă există soluţie, programul va determina mutările care conduc la soluţie
- scrie rezultatul la ieşirea standard

Date de intrare
Pe prima linie se află un număr întreg **n** – dimensiunea tablei ($2 \le \mathbf{n} \le 200$). Următoarele linii conţin descrierea configuraţiei iniţiale. Linia i+1 conţine n numere întregi **p**(**i**,1), **p**(**i**,2), …, **p**(**i**,**n**) separate prin câte un singur spaţiu.

Date de ieşire
Dacă nu există soluţie programul va scrie la ieşirea standard o singură linie conţinând numai cuvântul **NO** .
Dacă există soluţie, prima linie va conţine un număr întreg **m** – care este numărul de mutări care conduc la rezolvarea puzzelului. Numărul de mutări în soluţia voastră nu trebuie să

depăşească 400000. Următoarele **m** linii vor conține descrierea mutărilor, o mutare pe câte o linie.
Fiecare astfel de linie conține o literă **R** (pentru permutarea circulară a unei linii la dreapta) sau litera **C** (pentru permutarea circulară a unei coloane în jos) un spațiu şi două numere întregi **k** (1≤**k**≤**n**)şi **l** (1≤**l**≤**n**-1), separate printr-un singur spațiu. O linie care conține **R** ***k l*** descrie o permutare circulară a liniei **k** cu **l** celule la dreapta. După această mutare tabla va avea următoarea configurație:

$$p'(i,j)=\begin{cases} p(i,j+n-l) & dacă \quad i=k \quad şi \quad j\le l \\ p(i,j-l) & dacă \quad i=k \quad şi \quad j>l \\ p(i,j) & dacă \quad i\ne k \end{cases}$$

Similar, o linie care conține **C** ***k l*** descrie o permutare circulară a coloanei **k** cu **l** celule în jos.

Exemplu

Pentru datele de intrare

```
4
4 6 2 3
5 10 7 8
9 14 11 12
13 1 15 16
```

Rezultatul corect este

```
2
C 2 1
R 1 3
```

Secvența de mişcări de mai sus determină următoarea secvență de configurații

4	6	2	3
5	10	7	8
9	14	11	12
13	1	15	16

4	1	2	3
5	6	7	8
9	10	11	12
13	14	15	16

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Problema TWO
Două maşini de prelucrat lemne
Day 2, fişiere sursă two.*
Memorie disponibilă: 32 MB. Timp maxim de execuție 0.1 s.

De-a lungul unei şosele, care leagă vârful cu baza unui deal sunt **n** copaci. Administrația locală a decis să-i taie. Pentru a nu irosi lemn, copacii vor fi prelucrați cu maşini speciale. Copacii pot fi transportați doar de sus în jos. În partea de jos a şoselei este o maşină de prelucrat lemne. Două alte maşini de prelucrat lemne pot fi construite de-a lungul şoselei. Trebuie să decideți unde trebuie să fie construite aceste maşini de prelucrat lemne astfel încât costul de transport să fie minim. Costul transportului este de un cent pentru un kilogram de lemn transportat un metru.

Cerință
Scrieți un program, care:
- citeşte de la intrarea standard număerul de copaci, precum şi greutățile şi pozițiile lor
- calculează costul minim de transport
- scrie rezultatul la ieşirea standard

Date de intrare
Prima linie a fişierului de intrare conține un număr întreg **n** – numărul de copaci ($2 \le \mathbf{n} \le 20000$). Copacii sunt numerotați 1, 2, …, **n**, pornind din vârful dealului şi mergând spre vale. Fiecare dintre următoarele **n** linii conține două numere întregi pozitive separate prin câte un spațiu. Linia **i+1** conține: $\mathbf{w_i}$ – greutatea (în kilograme) a copacului **i** şi $\mathbf{d_i}$ – distanța (în metri) între copacii numerotați cu **i** şi **i+1**, ultimul dintre aceste numere $\mathbf{d_n}$, reprezintă distanța dintre copacul **n** şi vale (de unde începe şoseaua). Este garantat că costul total de transport al copacilor până la maşina de prelucrat lemne din vale este mai mică decât 2000000000.

Date de ieşire
Prima şi singura linie va conține un număr întreg care reprezintă costul minim de transport.

Exemplu

Pentru datele de intrare:	rezultatul corect este:
9 1 2 2 1 3 3 1 1 3 2 1 6 2 1 1 2 1 1	26

Figura urmatoare prezintă pozițiile optime a maşinilor de prelucrare a lemnelor pentru exemplu. Copacii sunt reprezentați prin cercuri având greutățile date sub cercuri. Maşinile de prelucrat lemne sunt marcate cu negru şi sunt poziționate în această figură în poziții în care se află copaci. Rezultatul este egal cu:

$$1\cdot(2+1) + 2\cdot1 + 1\cdot(1+2) + 3\cdot2 + 2\cdot(1+2+1) + 1\cdot(2+1) + 1\cdot1$$