Toy Train

Arezou și fratele ei Borzou sunt gemeni. Cei doi au primit ca și cadou de ziua lor un set cu trenuri de jucărie. Cei doi au construit cu ajutorul cadoului un sistem de cale ferată cu n stații și m linii *orientate*. Stațiile sunt numerotate de la 0 la $n-1$. Fiecare linie pleacă dintr-o stație si ajunge in aceeasi sau alta stație. Exista cel putin o linie care pleaca din fiecare stație.

Unele statii sunt *stații de încărcare*. Cand un tren ajunge într-o stație de incărcare, se încarcă complet. Un tren incărcat complet are destulă energie sa parcurgă n linii consecutive. Astfel, cand trenul ajunge pe a $n+1$-a linie de la ultima incarcare, va ramâne fără energie și se va opri.

Fiecare stație are un macaz care se poate indrepta către una din liniile care pornesc din stația curentă. Cand un tren se află într-o stație, pleaca din acea stație prin linia spre care e îndreptat macazul.

Gemenii vor să se joace cu trenul lor. Acestia și-au împarțit stațiile între ei: fiecare stație este deținuta de Arezou sau de Borzou. Există un singur tren. La începutul jocului trenul se află în stația s și e încărcat complet. Pentru a începe jocul, proprietarul stației s îndreapta macazul din stație către una din stațiile care pornesc din stația s. Apoi dau drumul la tren și trenul iși începe traseul dealungul liniilor.

De fiecare data cănd trenul intra într-o stație pentru prima data, proprietarul acelei stații fixează macazul din acea stație.

Odată ce macazul a fost fixat, acesta rămâne îndreptat spre aceeasi linie pentru restul jocului. Astfel, daca un tren reintra într-o stație pe care a vizitat-o înainte, o va părăsi pe aceeași linie ca data precedentă.

Din moment ce există un număr finit de stații, trenul eventual va intra intr-un *ciclu*. Un ciclu este o secvența de stații distincte $c[0], c[1], \cdots, c[k-1]$ astfel încât trenul pleacă din stația $c[i]$ (pentru $0 \leq i < k-1$) pe o linie către stația $c[i+1]$, și pleacă din stația $c[k-1]$ pe o linie către stația $c[0]$. Un ciclu poate fi format și dintr-o singură stație (ex. $k=1$) dacă trenul pleacă din stația $c[0]$ pe o linie care se indreaptă înapoi spre $c[0]$.

Arezou câștigă jocul dacă trenul merge incontinuu, și Borzou cîstigă dacă trenul rămâne fără energie. Altfel spus, dacăa există cel puțin o stație de încărcare printre $c[0], c[1], \cdots, c[k-1]$, trenul se poate reincărca, ceea ce înseamnă că va merge incontinuu și Arezou va câștiga. În caz contrar, trenul va rămâne fără energie (probabil dupa ce va parcurge ciclul de câteva ori), ceea ce inseamnă că va castiga Borzou.

Se dă sistemul de cale de ferata. Arezou și Borzou vor juca n jocuri. În al s-lea joc, pentru $0 \leq s \leq n-1$, trenul va fi inițial in statia s. Se cere pentru fiecare joc sa se afle daca exista o strategie pentru Arezou care sa îi garanteze victoria indiferent cum joaca Borzou.

Detalii de implementare

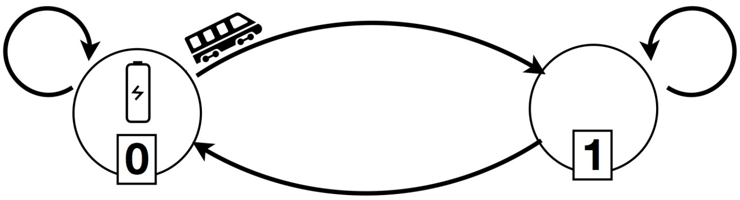
Se cere să se implementeze următoarea procedură:

```
int[] who_wins(int[] a, int[] r, int[] u, int[] v)
```

- a: un sir de lungime n. Daca Arezou deține a i-a stație, $a[i] = 1$. Altfel, Borzou deține a i-a stație si $a[i] = 0$.
- r: un șir de lungime n. Dacă a i-a statie este o stație de încărcare, atunci $r[i] = 1$. Altfel, $r[i] = 0$.
- u și v: doua șiruri de lungime m. Pentru toate $0 \leq i \leq m-1$, există o linie orientată care pleacă din stația $u[i]$ și ajunge în stația $v[i]$.
- Această procedură ar trebui să returneze un șir w de lungime n. Pentru toate $0 \leq i \leq n-1$, valoarea lui $w[i]$ ar trebui sa fie 1 daca Arezou poate câștiga jocul care incepe în stația i, neluand în considerare cum joacă Borzou. Altfel, valoare lui $w[i]$ ar trebui să fie 0.

Exemplu

```
who_wins([0, 1], [1, 0], [0, 0, 1, 1], [0, 1, 0, 1])
```



- Sunt 2 statii. Borzou este proprietarul stației 0, care este o stație de încărcare. Arezou este proprietarul stației 1, care nu este o stație de încărcare.
- Sunt 4 linii $(0,0), (0,1), (1,0)$, si $(1,1)$, unde (i,j) semnifica o linie orientata de la stația i la stația j.
- Considerăm jocul în care trenul este inițial în stația 0. Daca Borzou fixează macazul stației 0 către linia $(0,0)$, trenul va merge incontinuu prin această linie (de luat în considerare că stația 0 este o stație de încărcare). În acest caz, Arezou castigă. În caz contrar, daca Borzou fixează macazul stației 0 către linia $(0,1)$, Arezou poate îndrepta macazul stației 1 către linia $(1,0)$. În acest caz va merge incontinuu prin ambele stații. Din nou Arezou câștigă, din moment ce stația 0 este o statie de încărcate și trenul nu se va opri. Asadar, Arezou câștigă jocul, indiferent ce face Borzou.
- Similar, în jocul care incepe în stația 1 Arezou de asemenea poate câștiga, indiferent cum joaca Borzou. Asadar, procedura ar trebui să returneze $[1, 1]$.

Restricții si precizări

- $1 \leq n \leq 5000$.
- $n \leq m \leq 20\,000$.
- Există cel puțin o stație de încărcare.
- Există cel puțin o linie care pornește din fiecare stație.
- Pot exista linii care pornesc și se termină în aceeași stație (ex. $u[i] = v[i]$).
- Fiecare linie este distinctă. In alte cuvinte, nu există doua poziții i si j ($0 \leq i < j \leq m-1$) astfel incât $u[i] = u[j]$ si $v[i] = v[j]$.
- $0 \leq u[i], v[i] \leq n-1$ (pentru toate $0 \leq i \leq m-1$).

Subtask-uri

1. (5 puncte) Pentru fiecare $0 \leq i \leq m-1$, avem $v[i] = u[i]$ ori $v[i] = u[i] + 1$.
2. (10 puncte) $n \leq 15$.
3. (11 puncte) Arezou deține toate stațiile.
4. (11 puncte) Borzou deține toate stațiile.
5. (12 puncte) Exista o singură stație de încărcare.
6. (51 puncte) Nu exista restricții adiționale.

Evaluator local

Evaluatorul local citește inputul în următorul format:

- linia 1: $\ n\ \ m$
- linia 2: $\ a[0]\ \ a[1]\ \ldots\ a[n-1]$
- linia 3: $\ r[0]\ \ r[1]\ \ldots\ r[n-1]$
- linia $4+i$ (pentru $0 \leq i \leq m-1$): $\ u[i]\ \ v[i]$

Evaluatorul local printează valoarea returnată de `who_wins` în următorul format:

- linia 1: $\ w[0]\ \ w[1]\ \ldots\ w[n-1]$