電車のおもちゃ

双子の Arezou と Borzou は，誕生日にもらった電車のおもちゃで，n 個の駅と m 本の一方通行の線路からなる鉄道網を作った． 各々の線路はある駅を始点とし，ある駅を終点とする．始点と終点が同じこともあるかもしれない．各々の駅は，少なくとも 1 本の線路の始点となっている．

いくつかの駅は，**充電設備のある駅**である．電車は充電設備のある駅を訪れると，必ず完全に充電される．完全に充電された電車は，ちょうど線路 n 本分の距離を走ることができる． すなわち，最後に完全に充電されてから $n+1$ 本目の線路に差し掛かると，電車は充電不足で停止する．

各々の駅には，その駅を始点とする線路どれでも 1 つを指定できるスイッチがある．駅に入ってきた電車は，スイッチの指定する線路に沿って進む．

双子は，電車のおもちゃでゲームをしようとしている．双子は既に駅すべてを 2 人で分け終わっていて，それぞれの駅は Arezou または Borzou のちょうど片方がその所有者である．電車は 1 つだけあり，ゲームの開始時に完全に充電され，駅 s にある．ゲームの開始時に，駅 s の所有者は，駅 s のスイッチの指し示す線路を，s を始点とする線路の中から任意に設定し，電車はその線路に沿って進む．

ゲームの最中，ある駅に初めて電車が訪れたときにはいつでも，その駅の所有者はその駅のスイッチの指定する線路を任意に設定することができる．所有者によって一度指定する線路を設定されたスイッチは，それ以降指定する線路を変更することはできない． すなわち，もし電車が以前に訪れた駅に再び訪れたなら，電車は以前進んだのと同じ線路に沿って進む．

駅の個数は有限なので，最終的には電車は**サイクル**に陥る．サイクルとは**相異なる**駅からなる列 $c[0], c[1], \ldots, c[k-1]$ で， 電車が駅 $c[i] (0 \leq i < k-1)$ を訪れた直後に駅 $c[i+1]$ を訪れ，駅 $c[k-1]$ を訪れた直後に駅 $c[0]$ を訪れるものとする．始点と終点が同じ線路もあるかもしれないので，サイクルはただ 1 つの駅からなるかもしれない．

Arezou の勝利条件は電車を永遠に走らせ続けることであり，Borzou の勝利条件は電車を充電不足で停止させることである． すなわち，サイクルを成す駅 $c[0], c[1], \ldots, c[k-1]$ の中に 1 つでも充電設備のある駅があった場合，電車は永遠に走り続け，Arezou が勝利する．そうでない場合，電車はサイクルを何周かして充電不足で停止し，Borzou が勝利する．

鉄道網の情報が与えられる．双子はゲームを n 回行う．$s (0 \leq s \leq n-1)$ 回目のゲームでは，電車は最初駅 s にある． あなたの仕事は，電車が最初にある駅それぞれに対し，Arezou には Borzou の操作によらず必ず勝利する方法があるかどうかを求めることである．

実装の詳細

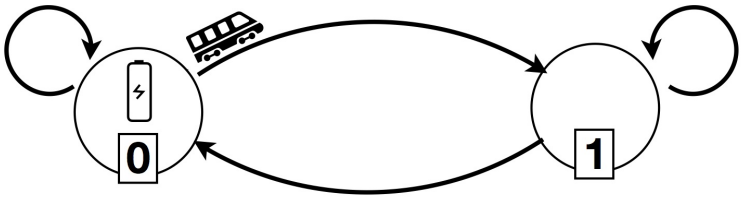
あなたは，以下のプロシージャを実装する必要がある．

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

- a: 長さ n の配列である. Arezou が駅 i の所有者である場合, $a[i] = 1$ である. そうでない場合, すなわち Borzou が駅 i の所有者である場合, $a[i] = 0$ である.
- r: 長さ n の配列である. もし駅 i が充電設備のある駅ならば, $r[i] = 1$ である. そうでない場合, $r[i] = 0$ である.
- u と v: 長さ m の配列である. すべての $0 \leqq i \leqq m - 1$ に対し, $u[i]$ を始点として $v[i]$ を終点とする一方通行の線路が存在する.
- このプロシージャは, 長さ n の配列 w を返さなければならない. 最初駅 i に電車がある場合に Arezou が勝利するなら, $w[i]$ は 1 でなければならない. そうでない場合, $w[i]$ は 0 でなければならない.

入出力例

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



- この入力例には, 駅が 2 つある. Borzou が充電設備のある駅 0 の所有者であり, Arezou は充電設備のない駅 1 の所有者である.
- (i, j) で 駅 i から駅 j へ向かう線路を表すことにすれば, $(0, 0), (0, 1), (1, 0), (1, 1)$ の 4 本の線路がある.
- 電車が最初に駅 0 に置かれている場合を考える. Borzou が駅 0 のスイッチを線路 $(0, 0)$ を指定するように設定したとき, 駅 0 は 充電設備がある駅なので, 電車は永遠に走り続け, Arezou が勝利する. Borzou が駅 0 のスイッチを線路 $(0, 1)$ を指定するように設定したとき, Arezou は駅 1 のスイッチを線路 $(1, 0)$ を指定するように設定すれば, 駅 0 は充電設備のある駅なので, 電車は 2 つの駅の間を永遠に走り続け, やはり Arezou が勝利する. よって, Borzou の操作によらず, Arezou が勝利する.
- 電車が最初に駅 1 に置かれている場合も, 同様の理由で, Borzou の操作によらず, Arezou が勝利する. よって, あなたのプロシージャは $[1, 1]$ を返さなければならない.

制約

- $1 \leqq n \leqq 5\,000$.
- $n \leqq m \leqq 20\,000$.
- 少なくとも 1 つの駅が, 充電設備のある駅である.

- 各々の駅は, 少なくとも 1 本の線路の始点となっている.
- 始点と終点が同じ線路もあるかもしれない. すなわち, $u[i] = v[i]$ かもしれない.
- すべての線路は相異なる. すなわち, 任意の $0 \leqq i < j \leqq m - 1$ に対し, $u[i] \neq u[j]$ または $v[i] \neq v[j]$ である.
- $0 \leqq u[i], v[i] \leqq n - 1$ $(0 \leq i \leq m - 1)$.

小課題

1. (5 点) すべての$0 \leqq i \leqq m - 1$ に対し, $v[i] = u[i]$ または $v[i] = u[i] + 1$ が成り立つ.
2. (10 点) $n \leqq 15$.
3. (11 点) Arezou がすべての駅の所有者である.
4. (11 点) Borzou がすべての駅の所有者である.
5. (12 点) 充電設備のある駅はちょうど 1 つである.
6. (51 点) 追加の制約はない.

採点プログラムのサンプル

採点プログラムのサンプルは以下の書式で入力を読み込む.

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

採点プログラムのサンプルは以下の書式で `who_wins` 関数の返り値を出力する.

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