



玩具火車

孿生雙胞胎Arezou和Borzou收到一份不尋常的玩具火車組當做他們的生日禮物，他們用這玩具組建構擁有 n 個車站及 m 段單向鐵道的鐵路系統。每一段鐵道從一個車站出發可以回到同一車站或到達一相異的車站。每一車站至少有一段鐵道由該站出發。

有些車站為具有充電功能的充電車站，每當玩具火車抵達充電車站時，它將被充飽電力，一輛充飽電力的玩具火車將有充足能量可以通過 n 段連續的鐵道。也就是，該火車在最近一次充飽電力後，在進入第 $n + 1$ 段鐵道時將耗盡電力。

每一車站都有一轉轍器可以用來指向任一由該車站出發的鐵道，停在一車站的火車將循著由該車站的轉轍器所指的鐵道駛離。

這對雙胞胎將用他們的玩具火車玩一遊戲，每一車站不是被Arezou擁有就是被Borzou擁有，並且只有一輛玩具火車，遊戲開始時該火車停在車站 s 並且充滿電力。為使遊戲開始，車站 s 的擁有者將設定其轉轍器指向由車站 s 出發的其中一條鐵道，之後他們開動該玩具火車使其開始沿著鐵道行駛。

當玩具火車第一次進入一車站時，該站的擁有者將設定其轉轍器指向由該站出發的一條鐵道，轉轍器一旦被設定後在本回合遊戲就不會再被改變，因此火車重新造訪之前到過的車站時，它將沿著和之前相同的鐵道離開。

因為車站的數量是有限的，該火車終將進入一迴圈，所謂迴圈是由一串相異的車站 $c[0], c[1], \dots, c[k - 1]$ 所構成，使得火車離開車站 $c[i]$ ($0 \leq i < k - 1$) 後將使用連向車站 $c[i + 1]$ 的鐵道，且火車離開車站 $c[k - 1]$ 時將使用連向車站 $c[0]$ 的鐵道。注意一迴圈可能只包含一個車站（也就是 $k = 1$ 時）如果火車駛離車站 $c[0]$ 並駛向連回車站 $c[0]$ 的鐵道。

如果該火車能不停地一直行駛下去，則Arezou將獲勝。如果該火車耗盡所有電力，則Borzou將獲勝。也就是，如果沿著車站 $c[0], c[1], \dots, c[k - 1]$ 存在至少一個充電車站，使得該火車可以充電並且可以沿著該迴圈無限行駛下去，則Arezou將獲勝。否則，該火車將耗盡電力（有可能沿著一迴圈繞行若干次後耗盡），則Borzou將獲勝。

你將有一份鐵路系統的描述，Arezou和Borzou將一起玩這遊戲 n 次，在第 s 次時 ($0 \leq s \leq n - 1$) 火車會由車站 s 開始。你的任務是對每次遊戲判斷，不管Borzou的策略為何，Arezou是否有一策略保證一定會贏。

Implementation details

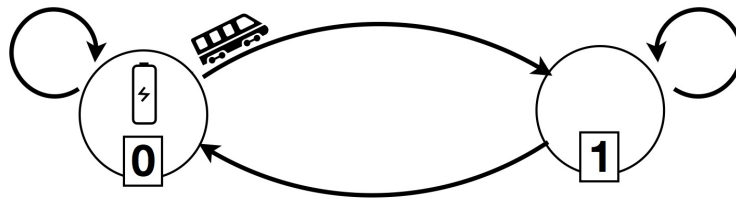
你需要實作以下函式：

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

- a : 長度為 n 的陣列, 若 Arezou 擁有車站 i , 則 $a[i] = 1$, 否則, 若 Borzou 擁有車站 i 則 $a[i] = 0$ 。
- r : 長度為 n 的陣列, 若車站 i 為充電車站時則 $r[i] = 1$, 否則 $r[i] = 0$ 。
- u 和 v : 分別為長度 m 的陣列, 對所有 $0 \leq i \leq m - 1$, 存在一段單向鐵道由車站 $u[i]$ 開始連向車站 $v[i]$ 。
- 這函式必須回傳一長度為 n 的陣列 w , 對每一 $0 \leq i \leq n - 1$, 如果不論 Borzou 的策略為何, Arezou 可以贏得由車站 i 開始的遊戲, 則 $w[i]$ 等於 1。否則 $w[i]$ 將等於 0。

Example

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



- 範例中2個車站 Borzou 擁有車站 0, 且該站為充電站。Arezou 擁有車站 1, 且該站不是充電站。
- 共有 4 段鐵道 $(0, 0)$, $(0, 1)$, $(1, 0)$ 和 $(1, 1)$, 其中 (i, j) 表示由車站 i 連向車站 j 的鐵道。
- 考慮遊戲一開始火車停在車站 0, 如果 Borzou 將車站 0 的轉轍器設定指向鐵道 $(0, 0)$ 則火車可以無限地沿著迴圈的鐵道行使下去(注意車站 0 為充電站), 在這情況下 Arezou 獲勝。否則, 如果 Borzou 將車站 0 的轉轍器設定指向鐵道 $(0, 1)$, 則 Arezou 可以將在車站 1 的轉轍器設定指向鐵道 $(1, 0)$ 。如果發生這情況, 則該火車將無限地在這兩個車站之間一直行駛下去, 同樣的, Arezou 將獲勝, 因為車站 0 是一充電站, 該火車將不會停下來。因此 不管 Borzou 的策略為何, Arezou 可以贏得勝利。
- 透過類似的理解方式, 如果火車是由車站 1 開始, 不管 Borzou 的策略為何, Arezou 也可以贏得勝利。所以函式應回傳 $[1, 1]$ 。

Constraints

- $1 \leq n \leq 5000$.
- $n \leq m \leq 20\,000$.
- 至少有一充電車站。
- 每一車站至少有一鐵道由該站出發。
- 可能存在鐵道其起點站和終點站是同一站 (也就是, $u[i] = v[i]$)。
- 每一鐵道都是相異的, 也就是, 不會有兩個註標(index) i 和 j ($0 \leq i < j \leq m - 1$) 使得 $u[i] = u[j]$ 及 $v[i] = v[j]$ 。
- $0 \leq u[i], v[i] \leq n - 1$ (對所有 $0 \leq i \leq m - 1$)。

Subtasks

1. (5 points) 對所有 $0 \leq i \leq m - 1$, 不是 $v[i] = u[i]$ 就是 $v[i] = u[i] + 1$ 。
2. (10 points) $n \leq 15$ 。

3. (11 points) Arezou 擁有所有車站。
4. (11 points) Borzou 擁有所有車站。
5. (12 points) 恰好只有一充電車站。
6. (51 points) 無額外限制。

Sample grader

樣板評分程式使用下列格式讀取輸入資料：

- line 1: n m
- line 2: $a[0]$ $a[1]$... $a[n - 1]$
- line 3: $r[0]$ $r[1]$... $r[n - 1]$
- line 4 + i (for $0 \leq i \leq m - 1$): $u[i]$ $v[i]$

樣板評分程式使用下列格式列印who_wins回傳的答案：

- line 1: $w[0]$ $w[1]$... $w[n - 1]$