

Alice si Bob se plictisesc. Asa ca au decis sa joace un joc numit Smart. Jocul Smart se joaca pe un graf conex neorientat mai special, in sensul in care un nod din graf este marcat ca radacina. Jucatorii muta pe rand. O mutare consta in eliminarea unei muchii din graf. Daca prin eliminarea muchiei din graf unele noduri nu mai sunt legate cu nodul radacina, nodurile respective si muchiile adiacente nodurilor se elimina si ele. Jucatorul care nu mai poate efectua nici o mutare (nu mai sunt muchii in graf) pierde.

## Cerinta

Scrieti un program care determina daca primul jucator are strategii sigure de castig.

**Date de intrare**

Datele de intrare se gasesc in fisierul `smart.in`. Fisierul contine 3 seturi de date, separate intre ele printr-o linie libera. Pe prima linie a unui set de date sunt scrise trei numere  $N$ ,  $M$  si  $K$  (in aceasta ordine) separate prin spatii.  $N$  reprezinta numarul de noduri din graf si  $M$  numarul de muchii. Nodurile grafului sunt numerotate de la 0 la  $N - 1$ .  $K$  este nodul radacina. Pe urmatoarele  $M$  linii se gasesc cate doua numere  $A$  si  $B$  separate prin spatii, semnificand faptul ca intre nodurile  $A$  si  $B$  exista o muchie.

**Date de iesire**

Fisierul `smart.out` va contine 3 linii, fiecare linie continand cifra 0 daca pentru setul respectiv de date jucatorul care incepe nu are strategie sigura de castig si cifra 1 daca are.

## Restriictii

- $N \leq 100$
- $M \leq 1\,000$
- $0 \leq K \leq N - 1$
- exista cel mult o muchie intre doua noduri.
- se poate intampla sa existe o muchie de la un nod la el insusi.
- in cel putin 40% din testele folosite la evaluare, graful initial va fi aciclic (doua noduri vor fi conectate prin cel mult un lant si nu vor exista bucle (muchii de la un nod la el insusi)).

## Exemplu

smart.in	smart.out
4 3 0	1
0 1	0
1 2	1
0 3	
3 2 0	
0 1	
0 2	
6 5 0	
0 1	
1 2	
2 3	
0 4	
4 5	

**Timp maxim de executie/test: 0.1 secunde**