



遊戲 (Game)

健佳是一個喜歡做遊戲的小男生。當有人問問題時，他更喜歡通過玩遊戲的方式作答，而不是直接回答。健佳碰到了他的朋友梅玉，跟她講了臺灣的航空網。在臺灣有 n 個城市（編號為 $0, \dots, n-1$ ），其中有些城市之間有航線。每個航線連接兩個城市，並且是雙向的。

梅玉問健佳，是否任意兩個城市之間都可以坐飛機互達（直接或間接），健佳不想直接回答，而是要通過做遊戲的方式來告訴她。梅玉可以問“城市 x 和 y 之間有直接航線嗎？”，健佳會立刻直接回答該問題。梅玉會詢問每對城市的情況恰好一次，因此總計會有 $r = n(n-1)/2$ 個問題。如果由前 i ($i < r$) 個問題的答案可以推斷出整個航空網是否連通，也就是說，是否任意一對城市之間都航線互通（直接或間接），梅玉就獲勝。否則意味著她需要獲知全部 r 個答案，此時健佳獲勝。

爲了讓遊戲更好玩，他們倆同意，健佳可以不要管臺灣的真實航空網，而是可以隨著遊戲的進展而編造航空網，也就是根據梅玉此前的提問來決定其後如何作答。你的任務是，通過決定健佳如何回答，來幫助他贏得遊戲。

例子

我們用三個例子來解釋遊戲規則。每個例子有 $n = 4$ 個城市和 $r = 6$ 個回合的問題和回答。

在第一個例子中（見下表），健佳在第 4 回合之後就會輸掉了遊戲，因爲不管健佳如何回答問題 5 和 6，梅玉都已經知任意兩個城市間是航線互通的。

回合	提問	回答
1	0, 1	yes
2	3, 0	yes
3	1, 2	no
4	0, 2	yes
-----	-----	-----
5	3, 1	no
6	2, 3	no

在下個例子中，梅玉在第 3 回合之後可以證明，無論健佳如何回答第 4、5、6 個問題，城市 0 和 1 都不可能航線互通，因此健佳又輸掉了。

回合	提問	回答
1	0, 3	no
2	2, 0	no
3	0, 1	no
----	-----	-----
4	1, 2	yes
5	1, 3	yes
6	2, 3	yes

在最後一個例子中，除非獲知所有 6 個問題的回答，否則梅玉無法確定是否任意城市之間航線互通，因此健佳贏了。具體而言，如果健佳對最後一個問題的回答是 *yes*（見下表），那麼任意兩個城市之間是航線互通的。否則，如果健佳的回答是 *no*，那麼就不是任意城市之間都航線互通。

回合	提問	回答
1	0, 3	no
2	1, 0	yes
3	0, 2	no
4	3, 1	yes
5	1, 2	no
6	2, 3	yes

任務

請寫出一個可以幫助健佳獲勝的程式。注意，無論是梅玉還是健佳，都不知道對方的策略。梅玉可以以任意的順序來詢問城市對，而健佳必須在不清楚後面問題的前提下立刻給出回答。你需要實現下面的兩個函數/過程：

- `initialize(n)` -- 我們會先調用你的 `initialize` 函數。參數 n 是城市數目。
- `hasEdge(u, v)` -- 接著我們會調用 `hasEdge` 函數 $r = n(n - 1)/2$ 次。這些調用代表了梅玉的問題，次序與她提問的次序相同。你必須回答在城市 u 和 v 之間是否有直接航線。具體而言，返回值 1 表示有，0 表示沒有。

子任務

每個子任務包含若干個遊戲。只有在你的程式幫助健佳贏得了某個子任務中的所有遊戲之後，才能得到該子任務的分數。

子任務	分值	n
1	15	$n = 4$
2	27	$4 \leq n \leq 80$

子任務	分值	n
3	58	$4 \leq n \leq 1500$

實現細節

你只能提交一個檔，名字為 `game.c`，`game.cpp` 或者 `game.pas`。該檔需實現上述的函數/過程，其命名與介面需遵循下面的要求。

C/C++程式

```
void initialize(int n);  
int hasEdge(int u, int v);
```

Pascal程式

```
procedure initialize(n: longint);  
function hasEdge(u, v: longint): longint;
```

評測相關

評測系統將讀入如下格式的輸入資料：

- 第1行： n
- 餘下 r 行：每行包含兩個整數 u 和 v ，表示對城市 u 和 v 的提問。