



## 游戏(Game)

健佳是一个喜欢做游戏的小男生。当有人问问题时，他更喜欢通过玩游戏的方式作答，而不是直接回答。健佳碰到了他的朋友梅玉，跟她讲了台湾的航空网。在台湾有 $n$ 个城市（编号为 $0, \dots, n-1$ ），其中有些城市之间有航线。每个航线连接两个城市，并且是双向的。

梅玉问健佳，是否任意两个城市之间都可以坐飞机互达（直接或间接），健佳不想直接回答，而是要通过玩游戏的方式来告诉她。梅玉可以问“城市 $x$ 和 $y$ 之间有直接航线吗？”，健佳会立刻直接回答该问题。梅玉会询问每对城市恰好一次，因此总计会有 $r = n(n-1)/2$ 个问题。如果由前 $i$  ( $i < r$ ) 个问题的答案可以推断出整个航空网是否连通，也就是说，是否任意一对城市之间都可以坐飞机互达（直接或间接），梅玉就获胜。否则意味着她需要知道全部 $r$ 个回答，此时健佳获胜。

为了让游戏更好玩，他们俩同意，健佳可以不要管台湾的真实航空网，而是可以随着游戏的进展而编造航空网，也就是根据梅玉此前的提问来决定此后如何作答。你的任务是，通过决定健佳如何回答，来帮助他赢得游戏。

### 例子

我们用三个例子来解释游戏规则。每个例子有 $n = 4$ 个城市和 $r = 6$ 个回合的提问和回答。

在第一个例子中（见下表），健佳在第4回合之后输掉了游戏，因为不管健佳如何回答提问5和6，梅玉都确切知道任意两个城市间是可以坐飞机互达的。

回合	提问	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

在下个例子中，梅玉在第3回合之后可以证明，无论健佳如何回答第4、5、6个问题，城市0和1都不可能坐飞机互达，因此健佳又输掉了。

回合	提问	回答
1	0, 3	no
2	2, 0	no

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

在最后一个例子中，除非知道所有6个提问的回答，否则梅玉无法确定是否任意城市之间都可以坐飞机互达，因此健佳赢了。具体而言，如果健佳对最后一个提问的回答是yes（见下表），那么任意两个城市之间都是可以坐飞机互达的。否则，如果健佳的回答是no，那么就不是任意城市之间都可以坐飞机互达。

回合	提问	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

## 任务

请写出一个可以帮助健佳获胜的程序。注意，无论是梅玉还是健佳，都不知道对方的策略。梅玉可以以任意的顺序来询问城市对，而健佳必须在不清楚后面提问的前提下立刻给出回答。你需要实现下面的两个函数：

- `initialize(n)` -- 我们会先调用你的`initialize`函数。参数  $n$  是城市数目。
- `hasEdge(u, v)` -- 接着我们会调用`hasEdge`函数  $r = n(n-1)/2$  次。这些调用代表了梅玉的提问，顺序与她提问的次序相同。你必须回答在城市  $u$  和  $v$  之间是否有直接航线。具体而言，返回值1表示有，0表示没有。

## 子任务

每个子任务包含若干个游戏。只有在你的程序帮助健佳赢得了某个子任务中的所有游戏之后，才能得到该子任务的分数。

子任务	分值	$n$
1	15	$n = 4$
2	27	$4 \leq n \leq 80$
3	58	$4 \leq n \leq 1500$

## 实现细节

你只能提交一个文件，名字为 `game.c`，`game.cpp` 或者 `game.pas`。该文件需实现上述的函数，其命名与接口需遵循下面的要求。

## C/C++程序

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

## Pascal程序

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

## 评测相关

评测系统将读入如下格式的输入数据：

- 第1行： $n$
- 余下 $r$ 行：每行包含两个整数 $u$ 和 $v$ ，表示对城市 $u$ 和 $v$ 的提问。