# Game

Jian-Jia adalah seorang pemuda yang menyukai permainan. Dia lebih menyukai bermain daripada menjawab langsung pertanyaan. Jian-Jia bertemu temannya Mei-Yu dan mengatakan padanya tentang jaringan penerbangan di Taiwan. Ada $n$ kota di Taiwan (dinomori 0, ..., $n-1$), beberapa di antara kota-kota tersebut dihubungkan oleh penerbangan. Setiap penerbangan menghubungkan dua kota dan penerbangan dapat dilaksanakan dua arah.

Mei-Yu bertanya ke Jian-Jia apakah mungkin untuk bepergian antara 2 kota manapun dengan pesawat terbang (baik secara langsung atau tidak langsung). Jian-Jia tidak ingin menjawabnya, malahan mengusulkan untuk bermain sebuah permainan. Mei-Yu dapat mengajukan pertanyaan dalam bentuk "Are cities $x$ and $y$ *directly* connected with a flight?", dan Jian-Jia akan langsung menjawabnya. Mei-Yu akan bertanya hanya sekali saja untuk setiap pasangan kota, sehingga seluruh pertanyaannya adalah $r = n(n-1)/2$ pertanyaan. Mei-Yu akan memenangkan permainan jika setelah mendapat jawaban dari $i$ pertanyaan pertama, dengan $i < r$, dia dapat menyimpulkan (*infer*) apakah jaringan kota tersebut saling terhubung (*connected*), yaitu apakah mungkin untuk bepergian dari suatu kota ke kota manapun dengan menggunakan penerbangan (langsung atau tidak langsung). Namun sebaliknya, jika ia membutuhkan seluruh $r$ pertanyaan, maka pemenangnya adalah Jian-Jia.

Agar permainan lebih menarik untuk Jian-Jia, teman-temannya setuju bahwa ia boleh mengabaikan penerbangan yang sebenarnya ada di Taiwan, boleh mengarang jaringan penerbangan seiring dengan berlangsungnya permainan, boleh menentukan jawabannya berdasarkan pertanyaan yang pernah diajukan Mei-Yu sebelumnya. Tugas Anda adalah membantu Jian-Jia memenangkan permainan, dengan memutuskan bagaimana ia sebaiknya menjawab pertanyaan.

# Examples

Aturan permainan dijelaskan melalui tiga contoh. Setiap contoh mempunyai $n = 4$ kota dan $r = 6$ putaran (*round*) menentukan tanya-jawab.

Pada contoh pertama (tabel sebagai berikut), Jian-Jia *kalah* sebab setelah putaran keempat, Mei-Yu yakin bahwa seseorang dapat bepergian dengan penerbangan antara dua kota manapun, apapun jawaban Jian-Jia terhadap pertanyaan 5 atau 6.

| round (*putaran*) | question | answer |
|---|---|---|
| 1 | 0, 1 | yes |
| 2 | 3, 0 | yes |
| 3 | 1, 2 | no |
| 4 | 0, 2 | yes |
| ----- | -------- | ------ |
| 5 | 3, 1 | no |
| 6 | 2, 3 | no |

Pada contoh berikutnya Mei-Yu dapat membuktikan bahwa setelah putaran ketiga, apapun jawaban Jian-Jia terhadap pertanyaan 4, 5, atau 6, seseorang *tak akan dapat* bepergian dengan penerbangan antara kota 0 dan kota 1, sehingga Jian-Jia kalah lagi.

| round (*putaran*) | question | answer |
|---|---|---|
| 1 | 0, 3 | no |
| 2 | 2, 0 | no |
| 3 | 0, 1 | no |
| ----- | -------- | ------ |
| 4 | 1, 2 | yes |
| 5 | 1, 3 | yes |
| 6 | 2, 3 | yes |

Pada contoh terakhir, Mei-Yu tak dapat menentukan apakah seseorang dapat bepergian dengan penerbangan antara dua kota manapun, sampai semua enam pertanyaan dijawab, sehingga Jian-Jia menang. Jika Jian-Jia menjawab *yes* ke pertanyaan terakhir (pada tabel sebagai berikut), maka dimungkinkan bepergian antara pasangan dua kota manapun. Kalau saja Jian-Jia menjawab *no* ke pertanyaan terakhir, akibatnya menjadi tidak mungkin.

| round (*putaran*) | question | answer |
|---|---|---|
| 1 | 0, 3 | no |
| 2 | 1, 0 | yes |
| 3 | 0, 2 | no |
| 4 | 3, 1 | yes |
| 5 | 1, 2 | no |
| 6 | 2, 3 | yes |

# Task

Tuliskan sebuah program yang membantu Jian-Jia untuk memenangkan permainan. Perhatikan bahwa baik Mei-Yu maupun Jian-Jia tidak tahu strategi masing-masing. Mei-Yu dapat menanyakan pasangan kota dengan urutan sembarang, dan Jian-Jia harus menjawab langsung tanpa tahu pertanyaan berikutnya. Anda perlu mengimplementasi dua fungsi sebagai berikut.

- `initialize(n)` -- We will call your `initialize` first. The parameter $n$ is the number of cities.
- `hasEdge(u, v)` -- Then we will call `hasEdge` for $r = n(n-1)/2$ times. These calls represent Mei-Yu's questions, in the order that she asks them. You must answer whether there is a direct flight between cities $u$ and $v$. Specifically, the return value should be 1 if there is a direct flight, or 0 otherwise.

# Subtasks

Setiap subtask terdiri dari beberapa permainan. Anda hanya akan mendapat point untuk sebuah subtask jika program Anda membuat Jian-Jia menang untuk seluruh permainan pada subtask tersebut.

| subtask | points | $n$ |
|---|---|---|
| 1 | 15 | $n = 4$ |
| 2 | 27 | $4 \leq n \leq 80$ |
| 3 | 58 | $4 \leq n \leq 1500$ |

# Implementation details

Anda harus men-*submit* satu file, dengan nama `game.c`, `game.cpp` atau `game.pas`. File ini mengimplementasikan subprogram yang dijelaskan di atas dengan menggunakan *signature* sebagai berikut.

## C/C++ programs

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

## Pascal programs

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

## Sample grader

The sample grader reads the input in the following format:

- line 1: `n`
- the following $r$ lines: each line contains two integers `u` and `v` that describe a question regarding cities $u$ and $v$.