



## Game

Andrej je dečak koji voli igre. Kada Andreju postavljaju pitanja, on radije igra igru zavlacenja umesto da odgovora direktno. Jednog dana, Andrej je sreo svog druga Ivana i ispričao mu je o Tajvanskoj avionskoj mreži (i o Markovoj opsednutosti kišobranima). U Tajvanu se nalazi  $n$  gradova (numerisani brojevima  $0, \dots, n - 1$ ) i neki od njih su povezani avionskim linijama. Svaki let povezuje neka dva različita grada i njime se može leteti u oba smera.

Ivan je pitao Andreja da li je moguće putovati avionom između svaka dva grada (bilo direktno ili indirektno). Andrej nije želeo da mu otkrije odgovor već je predložio da igraju igru. Ivan može da mu postavlja isključivo pitanja oblika "Da li su gradovi  $x$  i  $y$  direktno povezani avionskom linijom?", a Andrej na ova pitanja odgovara odmah. Ivan će pitati pitanja za svaki par gradova tačno jednom, tj. postaviće ukupno  $r = n(n - 1)/2$  pitanja. Ivan pobeđuje ako, posle dobijanja odgovora na prvih  $i$  pitanja za neko  $i < r$ , on može zaključiti da li je avionska mreža povezana, tj. da li je moguće putovati avionom između svaka dva grada (bilo direktno ili indirektno). U suprotnom, tj. ako mu je potrebno svih  $r$  pitanja, pobeđuje Andrej.

Da bi igra bila zabavnija za Andreja, drugari su se složili da Andrej može zaboraviti pravu Tajvansku avionsku mrežu i da mu je dopušteno da izmišlja ovu mrežu dok igra traje, birajući svoje odgovore na osnovu Ivanovih prethodnih pitanja. Vaš zadatak je da pomognete Andreju da pobeđi tako što ćete odlučivati kako će on odgovarati na pitanja.

## Primeri

Objasnićemo detaljnije pravila igre uz pomoć tri primera. Svaki primer ima  $n = 4$  gradova i  $r = 6$  rundi pitanja i odgovora.

U prvom primeru (sledeća tabela), Andrej *gubi* jer posle četvrte runde pitanja i odgovora, Ivan sa sigurnošću može zaključiti da je moguće putovati avionom između svaka dva grada, bez obzira na to kako Andrej odgovori na pitanja 5 i 6.

redni broj	pitanje	odgovor
1	0, 1	da
2	3, 0	da
3	1, 2	ne
4	0, 2	da
-----	-----	-----
5	3, 1	ne
6	2, 3	ne

U sledećem primeru, posle treće runde pitanja i odgovora Ivan može dokazati da bez obzira na to kako Andrej odgovori na pitanja 4, 5 i 6, *nije moguće* putovati avionom (ni direktno ni indirektno) između

gradova 0 i 1, pa Andrej ponovo gubi.

redni broj	pitanje	odgovor
1	0, 3	ne
2	2, 0	ne
3	0, 1	ne
-----	-----	-----
4	1, 2	da
5	1, 3	da
6	2, 3	da

U poslednjem primeru Ivan ne može odrediti da li je moguće putovati avionom između svaka dva grada sve dok ne dobije odgovor na svih šest pitanja, pa Andrej *pobeđuje*. Inače, kako je Andrej odgovorio *da* na poslednje pitanje (u sledećoj tabeli), sledi da je moguće putovati avionom između svaka dva grada. Međutim, da je Andrej odgovorio *ne* na poslednje pitanje, tada bi bilo nemoguće putovati avionom između svaka dva grada.

redni broj	pitanje	odgovor
1	0, 3	ne
2	1, 0	da
3	0, 2	ne
4	3, 1	da
5	1, 2	ne
6	2, 3	da

## Zadatak

Napišite program koji pomaže Andreju da pobedi. Obratite pažnju da ni Andrej ni Ivan ne znaju strategiju onog drugog. Ivan može postavljati pitanja za parove gradova u proizvoljnom redosledu, dok Andrej mora na njih odgovarati odmah, ne znajući buduća pitanja. Potrebno je da implementirate sledeće dve funkcije.

- `initialize(n)` -- Prvo ćemo pozvati vaš `initialize`. Parametar  $n$  je broj gradova.
- `hasEdge(u, v)` -- Zatim ćemo pozvati `hasEdge`  $r = n(n - 1)/2$  puta. Ovi pozivi predstavljaju Ivanova pitanja, u redosledu kojim ih on postavlja. Vi morate odgovoriti da li postoji direktna avionska linija između različitih gradova  $u$  i  $v$ . Preciznije, vrednost koju vraćate treba biti 1 ako postoji direktna avionska linija, odnosno 0 u suprotnom.

## Podzadaci

Svaki podzadatak se sastoji od nekoliko igara. Za odgovarajući podzadatak dobijate poene samo ako vaš program pobedi u svim igrama za Andreja.

podzadatak	poeni	$n$
1	15	$n = 4$
2	27	$4 \leq n \leq 80$
3	58	$4 \leq n \leq 1500$

## Detalji implementacije

Morate poslati tačno jedan fajl sa nazivom `game.c`, `game.cpp` ili `game.pas`. Taj fajl mora implementirati gore opisane funkcije/procedure koristeći sledeće potpise.

### C/C++ programi

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

### Pascal programi

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

### Opis grejdera

Grejder koji je dostupan na vašem računaru čita ulaz u sledećem formatu:

- linija 1:  $n$
- narednih  $r$  linija: svaka linija sadrži dva cela broja  $u$  i  $v$  i opisuje pitanje za gradove  $u$  and  $v$ .