Game

Jian-Jia é um jovem que gosta de jogos. Quando lhe fazem uma pergunta, ele prefere brincar ao invés de responder diretamente. Jian-Jia encontrou sua amiga Mei-Yu e falou-lhe sobre a rede aérea em Taiwan. Há n cidades em Taiwan (numeradas 0, ..., $n-1$), algumas das quais são conectadas por voos. Cada voo conecta duas cidades e pode ser usado em ambas as direções.

Mei-Yu perguntou a Jian-Jia se é possível viajar de avião entre quaisquer duas cidades (direta ou indiretamente). Jian-Jia não quis dar a resposta, ao invés disso sugeriu um jogo. Mei-Yu pode fazer perguntas da forma "As cidades x e y são conectadas *diretamente* por algum voo?", e Jian-Jia responderá essas perguntas de imediato. Mei-Yu fará a pergunta para cada par de cidades exatamente uma vez, num total de $r = n(n-1)/2$ perguntas. Mei-Yu ganha o jogo se, depois de obter respostas para as primeiras i perguntas para algum $i < r$, ela pode inferir se a rede é ou não conexa, ou seja, se é possível viajar de avião entre qualquer par de cidades (direta ou indiretamente). Caso contrário, isto é, se ela precisar de todas as r perguntas, então o vencedor será Jian-Jia.

Para deixar o jogo mais divertido para Jian-Jia, os amigos concordaram que ele não precisa levar em conta a rede aérea real de Taiwan, e pode inventar a sua rede à medida que o jogo progride, escolhendo suas respostas com base nas perguntas já feitas por Mei-Yu. Sua tarefa é ajudar Jian-Jia a ganhar o jogo, decidindo como ele deve responder às perguntas.

Exemplos

Vamos explicar as regras do jogo através de três exemplos. Cada exemplo tem $n = 4$ cidades e $r = 6$ rodadas de perguntas e respostas.

No primeiro exemplo (tabela abaixo), Jian-Jia *perde* porque depois da rodada 4, Mei-Yu tem certeza que se pode viajar de avião entre quaisquer duas cidades, quaisquer que sejam as respostas de Jian-Jia para as perguntas 5 e 6.

rodada	pergunta	resposta
1	0, 1	sim
2	3, 0	sim
3	1, 2	não
4	0, 2	sim
-----	--------	------
5	3, 1	não
6	2, 3	não

No próximo exemplo Mei-Yu pode provar, depois da rodada 3, que não importa quais as respostas de Jian-Jia para as perguntas 4, 5 e 6, *não se pode* viajar de avião entre as cidades 0 e 1, de forma que Jian-Jia perde também.

rodada	pergunta	resposta
1	0, 3	não
2	2, 0	não
3	0, 1	não
-----	--------	------
4	1, 2	sim
5	1, 3	sim
6	2, 3	sim

No último exemplo Mei-Yu não pode determinar se é possível viajar de avião entre quaisquer duas cidade antes que todas as seis perguntas sejam respondidas, de forma que Jian-Jia *ganha* o jogo. Especificamente, dado que Jian-Jia respondeu *sim* na última pergunta (veja a tabela abaixo), então é possível viajar entre qualquer par de cidades. Contudo, se Jian-Jia tivesse respondido *não* para a última questão, então seria impossível.

rodada	pergunta	resposta
1	0, 3	não
2	1, 0	sim
3	0, 2	não
4	3, 1	sim
5	1, 2	não
6	2, 3	sim

Tarefa

Favor escrever um programa que ajude Jian-Jia a ganhar o jogo. Note que nem Mei-Yu nem Jian-Jia conhecem a estratégia do outro. Mei-Yu pode fazer perguntas sobre pares de cidades em qualquer ordem, e Jian-Jia precisa respondê-las imediatamente, sem saber nada sobre as perguntas seguintes. Você precisa implementar as seguintes duas funções.

- `initialize(n)` -- Vamos chamar `initialize` primeiro. O parâmetro n é o número de cidades.
- `hasEdge(u, v)` -- Depois vamos chamar `hasEdge` $r = n(n-1)/2$ vezes. Essas chamadas representam as perguntas de Mei-Yu, na ordem em que ela as faz. Você precisa responder se existe um voo direto entre as cidades u e v. Mais precisamente, o valor retornado deve ser 1 se existe um voo direto, e 0 caso contrário.

Subtarefas

Cada subtarefa consiste de vários jogos. Você somente ganhará pontos para um subtarefa se seu programa ganha todos os jogos para Jian-Jia.

subtarefa	pontos	n
1	15	$n = 4$
2	27	$4 \leq n \leq 80$
3	58	$4 \leq n \leq 1500$

Detalhes de implementação

Você tem que submeter exatamente um arquivo, chamado `game.c`, `game.cpp` ou `game.pas`. Esse arquivo implementa os subprogramas descritos acima, usando as seguintes assinaturas.

Programas em C/C++

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

Programas em Pascal

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

Avaliador exemplo

O avaliador exemplo lê a entrada no seguinte formato:

- linha 1: `n`
- as r linhas seguintes: cada linha contém dois inteiros `u` and `v` que descrevem a pergunta relativa às cidades u e v.