



Игра

Маленькому мальчику Джан-Джи нравится играть. Когда ему задают вопрос, вместо простого ответа он предлагает сыграть в игру.

Джан-Джи встретил свою подружку Мей-Ю и рассказал ей о том, как устроена сеть авиарейсов в Тайване. В частности, в Тайване есть n городов, пронумерованных от 0 до $(n - 1)$. Между некоторыми из них действуют авиарейсы, каждый из них действует в обе стороны.

Мей-Ю спросила у Джан-Джи, правда ли, что возможно добраться из любого города в любой другой, используя только авиарейсы. Джан-Джи вместо ответа на вопрос предложил сыграть в игру.

Мей-Ю может задавать вопросы типа: «Соединены ли города x и y авиарейсом напрямую?». Джан-Джи должен сразу отвечать на каждый из вопросов. Мей-Ю спрашивает про каждую пару городов ровно один раз, то есть, задает всего $r = \frac{n(n-1)}{2}$ вопросов. Считается, что Мей-Ю победила в игре, если после получения ответов на первые i вопросов для некоторого $i < r$, она может определить, правда ли, что сеть авиарейсов связна. Это означает, что возможно добраться из любого города в любой другой, используя только авиарейсы. Если ей понадобилось задать все r вопросов, то выиграл Джан-Джи.

Чтобы сделать игру более интересной, друзья договорились, что они забудут о действующей сети авиарейсов в Тайване, а придумают свою сеть в процессе игры, выбирая ответ на текущий вопрос в зависимости от предыдущих вопросов Мей-Ю. Необходимо написать программу, которая поможет Джан-Джи выиграть игру, решая, как ему следует отвечать на каждый вопрос.

Примеры

Мы рассмотрим правила игры на следующих трех примерах. В каждом из них $n = 4$, $r = 6$.

В первом примере (таблица ниже) Джан-Джи *проигрывает*, потому что после 4-го вопроса, Мей-Ю уверена, что между любыми двумя городами можно долететь, используя только авиарейсы, не зависимо от того, какими будут ответы на вопросы 5 и 6.

номер	вопрос	ответ
1	0, 1	да
2	3, 0	да
3	1, 2	нет
4	0, 2	да
-----	-----	-----
5	3, 1	нет
6	2, 3	нет

В следующем примере Мей-Ю после третьего вопроса может определить, что, как бы Джан-Джи не отвечал дальше на остальные вопросы, *нельзя* добраться из города 0 в город 1, используя только авиарейсы. В этом случае Джан-Джи проигрывает снова.

номер	вопрос	ответ
1	0, 3	нет
2	2, 0	нет
3	0, 1	нет
-----	-----	-----
4	1, 2	да
5	1, 3	да
6	2, 3	да

В последнем примере Мей-Ю не может определить ответ на вопрос до тех пор, пока не получит ответы на все 6 вопросов, поэтому Джан-Джи *выигрывает*. В частности, так как Джан-Джи ответил "да" на последний вопрос, то долететь из любого города в любой другой возможно. Если бы он ответил "нет", то долететь нельзя.

номер	вопрос	ответ
1	0, 3	нет
2	1, 0	да
3	0, 2	нет
4	3, 1	да
5	1, 2	нет
6	2, 3	да

Постановка задачи

Напишите программу, которая поможет Джан-Джи победить в игре. Обратите внимание, что Мей-Ю и Джан-Джи не знают стратегию друг друга. Мей-Ю может спрашивать про пары городов в любом порядке, и Джан-Джи должен отвечать на каждый вопрос сразу, не дожидаясь остальных вопросов. Вы должны реализовать следующие две функции:

- `initialize(n)` — в начале будет вызвана функция `initialize`. Параметр n задает число городов;
- `hasEdge(u, v)` — потом будет вызвана функция `hasEdge` $r = \frac{n(n-1)}{2}$ раз. Каждый вызов этой функции — это очередной вопрос Мей-Ю. Вы должны выбрать ответ: имеется ли прямой авиарейс между городами u и v . В частности, возвращаемое значение должно быть равно 1, если прямой авиарейс есть, и 0 — в противном случае.

Подзадачи

Каждая подзадача состоит из нескольких игр. Вы получите баллы за подзадачу только в случае, если вы выиграете все игры.

подзадача	баллы	n
1	15	$n = 4$
2	27	$4 \leq n \leq 80$
3	58	$4 \leq n \leq 1500$

Детали реализации

Вы должны послать ровно один файл, названный `game.c`, `game.cpp` или `game.pas`. В этом файле должны быть реализованы функции, описанные выше с указанным ниже прототипом.

Язык C/C++

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

Язык Pascal

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

Пример проверяющего модуля

Предоставленный пример проверяющего модуля имеет следующий формат ввода данных:

- строка 1: n ;
- следующие r строк: каждая строка содержит два числа u и v , которые описывают вопрос про города u и v .

