

Задача А. Светофоры

Имя входного файла: `lights.in`
Имя выходного файла: `lights.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

В подземелье M тоннелей и N перекрестков, каждый тоннель соединяет какие-то два перекрестка. Мышиный король решил поставить по светофору в каждом тоннеле перед каждым перекрестком. Напишите программу, которая посчитает, сколько светофоров должно быть установлено на каждом из перекрестков. Перекрестки пронумерованы числами от 1 до N .

Формат входных данных

Во входном файле записано два числа N и M ($0 < N \leq 100$, $0 \leq M \leq \frac{N(N-1)}{2}$). В следующих M строках записаны по два числа i и j ($1 \leq i, j \leq N$), которые означают, что перекрестки i и j соединены тоннелем. Гарантируется, что никакой тоннель не соединяет перекресток сам с собой, и не существует двух различных тоннелей, соединяющих одну и ту же пару перекрестков.

Формат выходных данных

В выходной файл вывести N чисел: k -е число означает количество светофоров на k -м перекрестке.

Примеры

<code>lights.in</code>	<code>lights.out</code>
7 10 5 1 3 2 7 1 5 2 7 4 6 5 6 4 7 5 2 1 5 3	3 3 2 2 5 2 3

Задача В. От матрицы смежности к спискам смежности

Имя входного файла: `mtoal.in`
Имя выходного файла: `mtoal.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Простой ориентированный граф задан матрицей смежности. Выведите его представление в виде списков смежности.

Формат входных данных

В первой строке файла находится число N — количество вершин графа ($1 \leq N \leq 100$). Во второй строке и далее — матрица смежности. Гарантируется, что граф не содержит петель.

Формат выходных данных

Выведите N строк — списки смежности графа. В i -й строке сначала выведите количество исходящих из i -й вершины рёбер, а затем — номера вершин, в которые эти рёбра идут, упорядоченные по возрастанию.

Примеры

<code>mtoal.in</code>	<code>mtoal.out</code>
5	1 3
0 0 1 0 0	2 1 3
1 0 1 0 0	1 5
0 0 0 0 1	2 1 2
1 1 0 0 0	2 1 2
1 1 0 0 0	

Задача С. Истоки и стоки

Имя входного файла: `sourcesink.in`
Имя выходного файла: `sourcesink.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Вершина ориентированного графа называется истоком, если в нее не входит ни одно ребро, и стоком, если из нее не выходит ни одного ребра.

Ориентированный граф задан списком ребер. Найдите все его вершины-истоки и все вершины-стоки.

Формат входных данных

В первой строке записано два числа N и M — количество вершин ($1 \leq N \leq 100$) и количество ребер ($0 \leq M \leq 100000$). В следующих M строках записаны по два числа v и u ($1 \leq v, u \leq N$), которые означают, что существует ребро, которое ведет из v в u .

Формат выходных данных

В первой строке выведите K — число истоков в графе, затем номера вершин, являющихся истоками в порядке возрастания. Во второй строке выведите информацию о стоках в том же формате.

Примеры

<code>sourcesink.in</code>	<code>sourcesink.out</code>
4 2 2 4 3 1	2 2 3 2 1 4
5 7 5 5 4 3 5 4 2 1 1 4 2 4 1 2	0 1 3

Задача D. Цветной дождь

Имя входного файла: colour-rain.in
Имя выходного файла: colour-rain.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

В Банановой республике очень много холмов, соединенных мостами. На химическом заводе произошла авария, в результате чего испарилось экспериментальное удобрение «зюван». На следующий день выпал цветной дождь, причем он прошел только над холмами. В некоторых местах падали красные капли, в некоторых — синие, а в остальных — зеленые, в результате чего холмы стали соответствующего цвета. Президенту Банановой республики это понравилось, но ему захотелось покрасить мосты между вершинами холмов так, чтобы мосты были покрашены в цвет холмов, которые они соединяют. К сожалению, если холмы разного цвета, то покрасить мост таким образом не удастся. Посчитайте количество таких «плохих» мостов.

Формат входных данных

В первой строке записано два числа N и M — количество холмов ($1 \leq N \leq 100$) и количество мостов ($0 \leq M \leq \frac{N(N-1)}{2}$). В следующих M строках записаны по два числа v и u ($1 \leq v, u \leq N$), которые означают, что холмы v и u соединены мостом. Гарантируется, что никакой мост не соединит холм сам с собой, и не существует двух различных мостов, соединяющих одну и ту же пару холмов. В последней строке написаны N чисел k_1, k_2, \dots, k_N , которые обозначают цвет соответствующего холма: 1 — красный, 2 — синий, 3 — зеленый.

Формат выходных данных

Выведите количество мостов, соединяющих холмы разных цветов.

Примеры

colour-rain.in	colour-rain.out
2 1 2 1 1 1	0
3 3 1 3 1 2 2 3 3 3 1	2

Задача Е. Переселение сыщика

Имя входного файла: `two.in`
Имя выходного файла: `two.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Ниро Вульф решил переехать в другой город. Одна седьмая тонны веса мешает ему перемещаться быстро. В работе детектива необходимо быстро оказываться на месте преступления, а впоследствии настигать преступника, пока он не успел сбежать. Поэтому Вульф ищет такой город, в котором он мог бы с одной площади попасть на другую, проехав не более чем по двум улицам.

Напишите для него программу, которая по карте города сообщала, обладает ли город нужным свойством.

Формат входных данных

В первой строке заданы два числа: n — количество площадей ($n < 100$) и m — количество улиц между площадями.

В последующих m строках содержится пара чисел от 1 до n — начало и конец улицы.

Формат выходных данных

Выведите «YES», если город пригоден для жизни, и «NO» в противном случае.

Примеры

<code>two.in</code>	<code>two.out</code>
3 2 1 2 2 3	YES
4 3 1 2 2 3 3 4	NO

Задача F. Проверка на наличие кратных ребер, ориентированный вариант

Имя входного файла: `check.in`
Имя выходного файла: `check.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Ориентированный граф задан списком ребер. Проверьте, содержит ли он кратные ребра.

Формат входных данных

N — число вершин и M — число ребер ($1 \leq N \leq 100$, $1 \leq M \leq 10\,000$), затем M пар чисел — ребра графа.

Формат выходных данных

Выведите YES, если граф содержит параллельные ребра, иначе NO.

Примеры

<code>check.in</code>	<code>check.out</code>
5 3 2 5 3 1 3 2	NO
3 5 1 2 2 3 3 1 2 3 2 1	YES

Задача G. Полный граф

Имя входного файла: `complete.in`
Имя выходного файла: `complete.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Неориентированный граф называется полным, если любая пара его различных вершин соединена хотя бы одним ребром. Для заданного списком ребер графа без петель проверьте, является ли он полным.

Формат входных данных

Программе на вход даются числа N и M , где N — число вершин ($1 \leq N \leq 100$) и M — число ребер ($1 \leq M \leq 10000$), а затем M пар чисел — ребра графа.

Формат выходных данных

Выведите YES, если граф является полным, и NO в противном случае.

Примеры

<code>complete.in</code>	<code>complete.out</code>
3 3 1 2 1 3 2 3	YES

Задача Н. Проблемы с доступом в столовую?

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

2097 год. Прогресс всё-таки добрался до столовой «Берендеевых полян». Теперь на входе стоит индикатор, автоматически пропускающий внутрь только лкшат. Задумывалось, что индикатор будет считывать логотип ЛКШ, напечатанный на бейдже, и пропускать только тех, чей граф совпадает с настоящим графом лкшонка. Однако преподаватели быстро поняли, что такая система не работает, ведь если школьник потеряет бейдж и тот попадёт в чужие руки, в столовую сможет проникнуть посторонний. Поэтому лкшатский логотип перестали печатать на бейджах, а протранслировали его всем школьникам на самой первой линейке.

Теперь, чтобы попасть в столовую, нужно нарисовать на специальной панели лкшатский граф. Казалось бы, какие проблемы могут возникнуть в этот раз? А вот какие: далеко не все школьники могут нарисовать граф так же аккуратно как в оригинале, и система не пропускает их внутрь.

Преподаватели собираются решать эту проблему: они перепишут код индикатора так, чтобы тот пропускал внутрь любого, кто нарисовал граф изоморфный лкшатскому логотипу. Графы называются изоморфными, если можно биективно (то есть с однозначным соответствием друг другу) отобразить вершины одного в вершины другого таким образом, чтобы рёбра первого графа биективно переходили при этом отображении в рёбра второго.

Пока аккуратные школьники не съели все блинчики, помогите преподавателям — проверьте, совпадает ли данный граф с графом на логотипе (он есть на вашем бейдже).

Формат входных данных

Первая строка содержит два числа n ($1 \leq n \leq 20$, $0 \leq m \leq 40$) — число вершин и рёбер в графе, нарисованном школьником.

Далее следует m строк, задающих рёбра графа. Каждая из них содержит два числа u_i и v_i ($1 \leq u_i, v_i \leq n$) — номера концов соответствующего ребра.

Гарантируется, что граф не содержит кратных рёбер и петель.

Формат выходных данных

Выведите “YES” (без кавычек), если данный граф совпадает с настоящим графом с логотипа ЛКШ, и “NO” (без кавычек) иначе.

Примеры

стандартный ввод	стандартный вывод
17 27	YES
1 2	
1 3	
1 4	
1 5	
1 10	
2 3	
2 5	
2 6	
2 7	
3 7	
3 8	
3 9	
3 10	
4 5	
4 11	
4 10	
5 6	
5 12	
6 7	
6 13	
7 8	
7 14	
8 9	
8 15	
9 10	
9 16	
10 17	

Замечание

