

## Задача А. От матрицы к списку

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Задана матрица смежности графа. Необходимо по этой матрице вывести список смежности.

### Формат входных данных

В первой строке входных данных задано число  $n$  ( $1 \leq n \leq 100$ ). В следующих  $n$  строках задана матрица смежности. Каждая строка матрицы смежности состоит из чисел 0 или 1. 0 на пересечении  $i$ -й строки и  $j$ -го столбца означает, что существует ребро, исходящее из  $i$ -й вершины в  $j$ -ю. 0 означает, что между соответствующими вершинами ребра нет.

### Формат выходных данных

Выполните  $n$  строк. В начале каждой  $i$ -й строки выведите количество вершин, исходящих из  $i$ -й вершины. Затем в той же строке выведите номера этих вершин.

### Пример

стандартный ввод	стандартный вывод
5	3 2 4 5
0 1 0 1 1	3 1 3 5
1 0 1 0 1	4 1 2 4 5
1 1 0 1 1	1 1
1 0 0 0 0	1 4
0 0 0 1 0	

## Задача В. Классический ввод графа

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Задан неориентированный граф в виде списка ребер. Выведите его список смежности.

### Формат входных данных

В первой строке заданы два числа  $n$  и  $m$  ( $1 \leq n, m \leq 100$ ), соответственно, количество вершин и количество ребер. Далее в следующих  $n$  строках задано описание каждого ребра в виде двух вершин  $u$  и  $v$ , между которыми существует ребро графа.

### Формат выходных данных

Выведите  $n$  строк. В начале каждой  $i$ -й строки выведите количество вершин, исходящих из  $i$ -й вершины. Затем в той же строке выведите номера этих вершин.

### Пример

стандартный ввод	стандартный вывод
5 7	3 2 4 5
1 2	3 1 4 5
3 4	1 4
2 4	4 1 2 3 5
5 1	3 1 2 4
4 1	
5 2	
4 5	

## Задача С. Задача 16 (почти) для графов

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Задан неориентированный граф списком рёбер. Необходимо найти:

1. изолированные вершины этого графа;
2. максимальную степень вершины;
3. количество кратных ребер;
4. количество петель;

### Формат входных данных

В первой строке заданы два числа  $n$  и  $m$  — количество вершин и количество рёбер в графе ( $1 \leq n \leq 10^5$ ,  $0 \leq m \leq 10^5$ ). Далее в  $m$  строках заданы описания рёбер в виде двух чисел — номера вершин  $u_i$  и  $v_i$ , которые соединяет ребро ( $1 \leq u_i, v_i \leq n$ ).

### Формат выходных данных

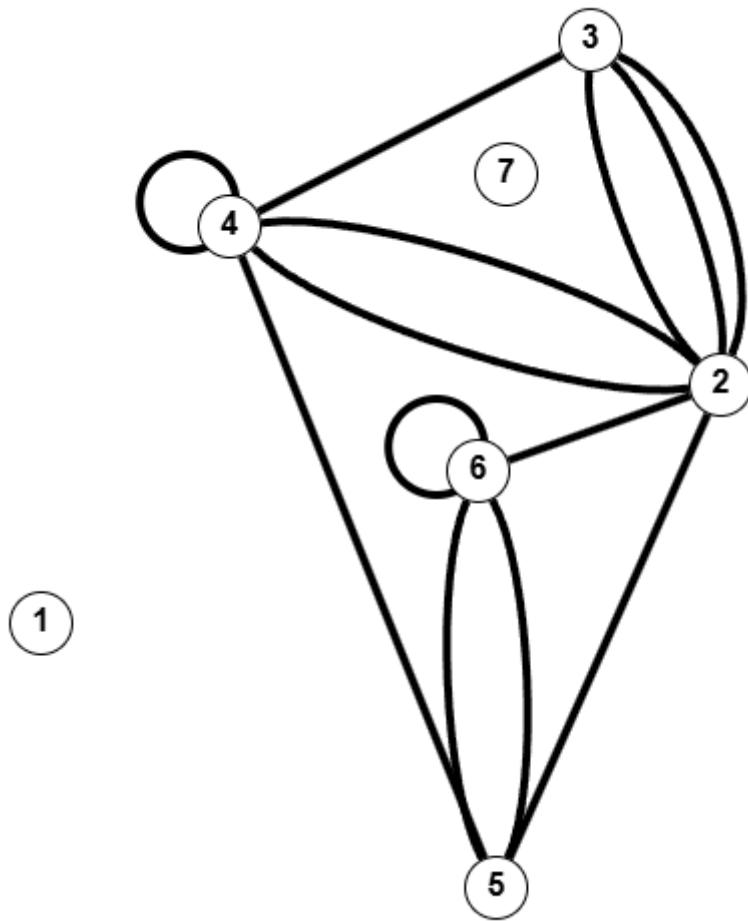
Выведите в первой строке список изолированных вершин в возрастающем порядке. Во второй строке — максимальную степень вершины, в третьей — количество кратных ребер, в четвёртой — количество петель.

### Пример

стандартный ввод	стандартный вывод
7 13	1 7
2 4	7
4 3	3
2 5	2
3 2	
5 6	
6 6	
2 3	
2 4	
6 2	
4 4	
4 5	
2 3	
5 6	

### Замечание

Граф из примера приведен на следующем рисунке:



## Задача D. Получи дерево

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Александр решил посадить у себя в саду дерево. Он купил его в интернет-магазине, и на следующий день его обещали доставить. Но произошла какая-то ошибка, и вместо дерева Александр получил неориентированный связный граф. Оставлять в таком виде граф нельзя: в нем могут быть циклы, в том числе петли, кратные ребра — совсем некрасиво. Поэтому Александр взял топор и решил удалить некоторое количество ребер из графа.

Выведите любое дерево, которое могло получиться у Александра в результате удаления ребер из графа.

### Формат входных данных

Сначала вводятся 2 числа  $n$  и  $m$  — число вершин и ребер в графе ( $1 \leq n \leq 10^5$ ,  $1 \leq m \leq 2 \cdot 10^5$ ). Далее идут  $m$  пар чисел, задающих ребра. Гарантируется, что граф связный.

### Формат выходных данных

Выведите количество вершин в полученном дереве, а затем с новой строки — ребра, которые войдут в дерево. Ребра, как и концы каждого ребра, можно выводить в любом порядке.

### Пример

стандартный ввод	стандартный вывод
4 5	3
1 2	1 2
3 2	2 3
1 4	2 4
2 4	
4 2	

## Задача Е. Поиск цикла

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Дан ориентированный граф. Необходимо определить есть ли в нём циклы, и если есть, то вывести любой из них.

### Формат входных данных

В первой строке находятся два натуральных числа  $N$  и  $M$  ( $1 \leq N \leq 100\,000$ ,  $M \leq 100\,000$ ) — количество вершин и рёбер в графе соответственно. Далее в  $M$  строках перечислены рёбра графа. Каждое ребро задаётся парой чисел — номерами начальной и конечной вершин соответственно.

### Формат выходных данных

Если в графе нет цикла, то вывести «NO», иначе — «YES» и затем перечислить все вершины в порядке обхода цикла.

### Примеры

стандартный ввод	стандартный вывод
3 3 1 2 2 3 3 1	YES 2 3 1
3 3 1 2 1 3 2 3	NO

## Задача F. Компоненты связности

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Дан неориентированный граф. Требуется узнать количество компонент связности этого графа.

### Формат входных данных

В первой строке через пробел заданы целые числа  $N$  и  $M$  — количество вершин и ребер графа ( $1 \leq N \leq 10^5$ ,  $0 \leq M \leq 10^5$ ). В следующих  $M$  строках через пробел заданы пары целых чисел  $u$  и  $v$  — номера вершин, соединенных ребром ( $1 \leq u, v \leq N$ ).

### Формат выходных данных

Выведите единственное число — количество компонент связности этого графа.

### Пример

стандартный ввод	стандартный вывод
5 4 1 2 2 3 1 3 4 5	2

## Задача G. Занесло снегом!

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

В некотором районе располагается  $n$  поселков и  $m$  дорог, которые соединяют какие-то два поселка. После сильного снегопада все эти дороги были занесены снегом. Для того, чтобы освободить  $i$ -ю дорогу, потребуется  $h_i$  часов. Время уборки одной дороги не зависит от времени уборки других дорог.

Какое минимальное количество часов должно пройти, чтобы все города были соединены? То есть для любой пары городов найдется маршрут между двумя городами, который проходит через одну или несколько уже открытых дорог.

Гарантируется, что по прошествии достаточного количества часов для использования всех дорог между каждой парой городов появится такой маршрут.

### Формат входных данных

В первой строке заданы два натуральных числа  $n$  и  $m$  ( $2 \leq n, m \leq 10^5$ ). Затем следуют  $m$  строк, каждая  $i$ -я строка содержит три натуральных числа  $u_i, v_i, h_i$  ( $1 \leq u_i, v_i \leq n, 1 \leq h_i \leq 10^6$ ) и описывает дорогу из  $u_i$  поселка в  $v_i$ , для разблокировки от снега которой потребуется  $h_i$  часов.

### Формат выходных данных

Выведите единственное число — минимальное количество времени, через которое будет возможно попасть из любого поселка в любой другой.

### Примеры

стандартный ввод	стандартный вывод
3 3 1 2 10 1 3 20 2 3 30	20
4 3 1 2 10 1 3 20 1 4 30	30

## Задача Н. Хорошая раскраска

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Миша и Андрей — большие любители хороших раскрасок, а также задач, с ними связанных. В лагере, куда они поехали имеется некоторое количество таких раскрасок.

Раскраска состоит из  $n \times m$  клеток. Каждая клетка раскраски может быть окрашена в один из 10 цветов, которые пронумерованы числами от 0 до 9. Окрашенные клетки могут соединяться в рисунки. Хорошим рисунком назовем такое изображение, которое состоит из клеток одного цвета, при этом между любыми двумя клетками можно провести линию, состоящую из клеток того же цвета причем только одну.

Ребята выбрали одну из хороших раскрасок. Теперь их интересует, сколько хороших рисунков изображено на этой раскраске.

### Формат входных данных

В первой строке записано два натуральных числа  $n$  и  $m$  ( $1 \leq n, m \leq 500$ ). В следующих  $n$  строках дано по  $m$  чисел через пробел — описание раскраски. Числа в этом описании лежат в диапазоне от 0 до 9.

### Формат выходных данных

В первой строке выведите число — количество хороших рисунков.

### Примеры

стандартный ввод	стандартный вывод
7 5 1 3 4 4 4 1 3 3 4 4 1 1 3 4 4 2 1 3 3 4 1 1 1 3 4 2 2 1 3 4 2 2 2 3 4	3
2 2 1 2 3 4	4

## Задача I. Конденсация графа

Имя входного файла: condense2.in

Имя выходного файла: condense2.out

Ограничение по времени: 0.65 секунда

Ограничение по памяти: 256 мегабайт

Требуется найти количество рёбер в конденсации ориентированного графа. Примечание: конденсация графа не содержит кратных рёбер и петель.

### Формат входных данных

Первая строка входного файла содержит два натуральных числа  $n$  и  $m$  — количество вершин и рёбер графа соответственно ( $n \leq 10\,000, m \leq 100\,000$ ). Следующие  $m$  строк содержат описание рёбер, по одному на строке. Ребро номер  $i$  описывается двумя натуральными числами  $b_i, e_i$  — началом и концом ребра соответственно ( $1 \leq b_i, e_i \leq n$ ). В графе могут присутствовать кратные рёбра и петли.

### Формат выходных данных

Первая строка выходного файла должна содержать одно число — количество рёбер в конденсации графа.

### Примеры

condense2.in	condense2.out
4 4	
2 1	
3 2	
2 3	
4 3	2

## Задача J. Построение

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Группа солдат-новобранцев прибыла в армейскую часть №666. После знакомства с прaporщиком стало очевидно, что от работ на кухне по очистке картофеля спаси солдат может только чудо.

Прaporщик, будучи не в состоянии запомнить фамилии, пронумеровал новобранцев от 1 до  $N$ . После этого он велел им построиться по росту (начиная с самого высокого). С этой несложной задачей могут справиться даже совсем необученные новобранцы, да вот беда, прaporщик уверил себя, что знает про некоторых солдат, кто из них кого выше, и это далеко не всегда соответствует истине.

После трех дней обучения новобранцам удалось выяснить, что знает (а точнее, думает, что знает) прaporщик. Помогите им, используя эти знания, построиться так, чтобы товарищ прaporщик остался доволен.

### Формат входных данных

Сначала на вход программы поступают числа  $N$  и  $M$  ( $2 \leq N \leq 10^5$ ,  $1 \leq M \leq 2 \cdot 10^5$ ) — количество солдат в роте и количество пар солдат, про которых прaporщик знает, кто из них выше.

Далее идут эти пары чисел  $A$  и  $B$  по одной на строке ( $1 \leq A, B \leq N$ ), что означает, что, по мнению прaporщика, солдат  $A$  выше, чем  $B$ .

Не гарантируется, что все пары чисел во входных данных различны.

### Формат выходных данных

В первой строке выведите «Yes» (если можно построиться так, чтобы прaporщик остался доволен) или «No» (если нет).

После ответа «Yes» на следующей строке выведите  $N$  чисел, разделенных пробелами, — одно из возможных построений.

### Примеры

стандартный ввод	стандартный вывод
2 2 1 2 2 1	No
3 7 1 2 2 3 1 3 2 3 1 2 1 2 1 3	Yes 1 2 3

## Задача К. Противопожарная безопасность

Имя входного файла:	<code>firesafe.in</code>
Имя выходного файла:	<code>firesafe.out</code>
Ограничение по времени:	0.5 секунда
Ограничение по памяти:	256 мегабайт

В Судиславле  $n$  домов. Некоторые из них соединены дорогами с односторонним движением.

В последнее время в Судиславле участились случаи пожаров. В связи с этим жители решили построить в посёлке несколько пожарных станций. Но возникла проблема: едущая по вызову пожарная машина, конечно, может игнорировать направление движения текущей дороги, однако возвращающаяся с задания машина обязана следовать правилам дорожного движения (жители Судиславля свято чтут эти правила!).

Ясно, что, где бы ни оказалась пожарная машина, у неё должна быть возможность вернуться на ту пожарную станцию, с которой она выехала. Но строительство станций стоит больших денег, поэтому на совете посёлка было решено построить минимальное количество станций таким образом, чтобы это условие выполнялось. Кроме того, для экономии было решено строить станции в виде пристроек к уже существующим домам.

Ваша задача — написать программу, рассчитывающую оптимальное положение станций.

### Формат входных данных

В первой строке входного файла задано число  $n$  ( $1 \leq n \leq 3\,000$ ). Во второй строке записано количество дорог  $m$  ( $1 \leq m \leq 100\,000$ ). Далее следует описание дорог в формате  $a_i$   $b_i$ , означающее, что по  $i$ -й дороге разрешается движение автотранспорта от дома  $a_i$  к дому  $b_i$  ( $1 \leq a_i, b_i \leq n$ ).

### Формат выходных данных

В первой строке выведите минимальное количество пожарных станций  $K$ , которое необходимо построить. Во второй строке выведите  $K$  чисел в произвольном порядке — дома, к которым необходимо пристроить станции. Если оптимальных решений несколько, выведите любое.

### Примеры

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