

## Задача А. 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

<code>condense2.in</code>	<code>condense2.out</code>
4 4 2 1 3 2 2 3 4 3	2

## Задача В. 2-SAT

Имя входного файла: 2sat.in  
Имя выходного файла: 2sat.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Формулировка 2-SAT: нужно подобрать значения  $n$  булевых переменных так, чтобы все  $m$  утверждений вида  $x_{i_1} = e_1 \vee x_{i_2} = e_2$  обратились в истину. В данной задаче вам гарантируется, что решение существует.

### Формат входных данных

Входной файл состоит из одного или нескольких тестов.

Каждый тест описывается следующим образом. На первой строке число переменных  $n$  и число утверждений  $m$ . Каждая из следующих  $m$  строк содержит числа  $i_1, e_1, i_2, e_2$ , задает утверждение  $x_{i_1} = e_1 \vee x_{i_2} = e_2$  ( $0 \leq i_j < n$ ,  $0 \leq e_j \leq 1$ ). Ограничения: сумма всех  $n$  не больше 100 000, сумма всех  $m$  не больше 300 000.

### Формат выходных данных

Для каждого теста выведите строку из  $n$  нулей и единиц — значения переменных. Если у данной задачи 2-SAT есть несколько решений, выведите любое.

### Примеры

2sat.in	2sat.out
1 0	0
2 2	01
0 0 1 0	000
0 1 1 1	
3 4	
0 1 1 0	
0 0 2 1	
1 1 2 0	
0 0 0 1	

## Задача С. Цикл отрицательного веса

Имя входного файла: `negcycle.in`  
Имя выходного файла: `negcycle.out`  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 256 мегабайта

Дан ориентированный граф. Определите, есть ли в нем цикл отрицательного веса, и если да, то выведите его.

### Формат входных данных

Во входном файле в первой строке число  $N$  ( $1 \leq N \leq 100$ ) — количество вершин графа. В следующих  $N$  строках находится по  $N$  чисел — матрица смежности графа. Все веса ребер не превышают по модулю 10 000. Если ребра нет, то соответствующее число равно 100 000.

### Формат выходных данных

В первой строке выходного файла выведите «YES», если цикл существует или «NO» в противном случае. При его наличии выведите во второй строке количество вершин в искомом цикле и в третьей строке — вершины входящие в этот цикл в порядке обхода.

### Примеры

<code>negcycle.in</code>	<code>negcycle.out</code>
2	YES
0 -1	2
-1 0	2 1

## Задача D. Pink Floyd

Имя входного файла: floyd.in  
Имя выходного файла: floyd.out  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Группа Pink Floyd собирается отправиться в новый концертный тур по всему миру. По предыдущему опыту группа знает, что солист Роджер Уотерс постоянно нервничает при перелетах. На некоторых маршрутах он теряет вес от волнения, а на других — много ест и набирает вес.

Известно, что чем больше весит Роджер, тем лучше выступает группа, поэтому требуется спланировать перелеты так, чтобы вес Роджера на каждом концерте был максимально возможным.

Группа должна посещать города в том же порядке, в котором она дает концерты. При этом между концертами группа может посещать промежуточные города.

### Формат входных данных

Первая строка входного файла содержит три натуральных числа  $n$ ,  $m$  и  $k$  — количество городов в мире, количество рейсов и количество концертов, которые должна дать группа соответственно ( $n \leq 100$ ,  $m \leq 10\,000$ ,  $2 \leq k \leq 10\,000$ ). Города пронумерованы числами от 1 до  $n$ .

Следующие  $m$  строк содержат описание рейсов, по одному на строке. Рейс номер  $i$  описывается тремя числами  $b_i$ ,  $e_i$  и  $w_i$  — номер начального и конечного города рейса и предполагаемое изменение веса Роджера в миллиграммах ( $1 \leq b_i, e_i \leq n$ ,  $-100\,000 \leq w_i \leq 100\,000$ ).

Последняя строка содержит числа  $a_1, a_2, \dots, a_k$  — номера городов, в которых проводятся концерты ( $a_i \neq a_{i+1}$ ). В начале концертного тура группа находится в городе  $a_1$ .

Гарантируется, что группа может дать все концерты.

### Формат выходных данных

Первая строка выходного файла должна содержать число  $l$  — количество рейсов, которые должна сделать группа. Вторая строка должна содержать  $l$  чисел — номера используемых рейсов.

Если существует такая последовательность маршрутов между концертами, что Роджер будет набирать вес неограниченно, то первая строка выходного файла должна содержать строку “infinitely kind”.

### Примеры

floyd.in	floyd.out
4 8 5	6
1 2 -2	5 6 5 7 2 3
2 3 3	
3 4 -5	
4 1 3	
1 3 2	
3 1 -2	
3 2 -3	
2 4 -10	
1 3 1 2 4	

## Задача Е. Раскраска в три цвета

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

Петя нарисовал на бумаге  $n$  кружков и соединил некоторые пары кружков линиями. После этого он раскрасил каждый кружок в один из трех цветов — красный, синий или зеленый.

Теперь Петя хочет изменить их раскраску. А именно — он хочет перекрасить каждый кружок в некоторый другой цвет так, чтобы никакие два кружка одного цвета не были соединены линией. При этом он хочет обязательно перекрасить каждый кружок, а перекрашивать кружок в тот же цвет, в который он был раскрашен исходно, не разрешается.

Помогите Пете решить, в какие цвета следует перекрасить кружки, чтобы выполнялось указанное условие.

### Формат входных данных

Первая строка содержит два целых числа  $n$  и  $m$  — количество кружков и количество линий, которые нарисовал Петя, соответственно ( $1 \leq n \leq 1000$ ,  $0 \leq m \leq 20000$ ).

Следующая строка содержит  $n$  символов из множества  $\{‘R’, ‘G’, ‘B’\}$  —  $i$ -й из этих символов означает цвет, в который раскрашен  $i$ -й кружок (‘R’ — красный, ‘G’ — зеленый, ‘B’ — синий).

Следующие  $m$  строк содержат по два целых числа — пары кружков, соединенных отрезками.

### Формат выходных данных

Выведите в выходной файл одну строку, состоящую из  $n$  символов из множества  $\{‘R’, ‘G’, ‘B’\}$  — цвета кружков после перекраски. Если решений несколько, выведите любое.

Если решения не существует, выведите в выходной файл слово “Impossible”.

### Примеры

<code>color.in</code>	<code>color.out</code>
4 5 RRRG 1 3 1 4 3 4 2 4 2 3	GGBR
4 5 RGRR 1 3 1 4 3 4 2 4 2 3	Impossible

## Задача F. Противопожарная безопасность

Имя входного файла: `firesafe.in`  
Имя выходного файла: `firesafe.out`  
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  $n$  ( $1 \leq n \leq 3000$ ). 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	