

## Задача А. Компоненты связности

Имя входного файла: `components1.in`  
Имя выходного файла: `components1.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Дан неориентированный невзвешенный граф. Необходимо посчитать количество его компонент связности.

### Формат входного файла

В первой строке входного файла содержится одно натуральное число  $N$  ( $N \leq 100$ ) — количество вершин в графе. Далее в  $N$  строках по  $N$  чисел — матрица смежности графа: в  $i$ -й строке на  $j$ -м месте стоит 1, если вершины  $i$  и  $j$  соединены ребром, и 0, если ребра между ними нет. На главной диагонали матрицы стоят нули. Матрица симметрична относительно главной диагонали.

### Формат выходного файла

Вывести одно целое число — искомое количество компонент связности графа.

### Пример

<code>components1.in</code>	<code>components1.out</code>
6 0 1 1 0 0 0 1 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0	3

## Задача В. Баобаб

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

Дан неориентированный невзвешенный граф. Необходимо определить, является ли он деревом.

### Формат входного файла

В первой строке входного файла содержится одно натуральное число  $N$  ( $N \leq 100$ ) — количество вершин в графе. Далее в  $N$  строках по  $N$  чисел — матрица смежности графа: в  $i$ -й строке на  $j$ -м месте стоит 1, если вершины  $i$  и  $j$  соединены ребром, и 0, если ребра между ними нет. На главной диагонали матрицы стоят нули. Матрица симметрична относительно главной диагонали.

### Формат выходного файла

Вывести «YES», если граф является деревом, и «NO» иначе.

### Пример

<code>baobab.in</code>	<code>baobab.out</code>
3 0 1 0 1 0 1 0 1 0	YES

## Задача С. Долой списывание!

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

Во время контрольной работы профессор заметил, что некоторые студенты обмениваются записками. Сначала он хотел поставить им всем двойки, но в тот день профессор был добрым, а потому решил разделить студентов на две группы: списывающих и дающих списывать, и поставить двойки только первым.

У профессора записаны все пары студентов, обменявшихся записками. Требуется определить, сможет ли он разделить студентов на две группы так, чтобы любой обмен записками осуществлялся от студента одной группы студенту другой группы.

### Формат входного файла

В первой строке входного файла записаны два числа  $N$  и  $M$  — количество студентов и количество пар студентов, обменивающихся записками ( $1 \leq N \leq 100$ ,  $0 \leq M \leq \frac{N(N-1)}{2}$ ). Далее в  $M$  строках расположены описания пар студентов: два числа, соответствующие номерам студентов, обменивающихся записками (нумерация студентов идёт с 1). Каждая пара студентов перечислена не более одного раза.

### Формат выходного файла

Необходимо вывести ответ на задачу профессора. Если возможно разделить студентов на две группы, выведите «YES», иначе выведите «NO».

### Примеры

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

## Задача D. Компоненты связности — 2

Имя входного файла: `components2.in`  
Имя выходного файла: `components2.out`  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 256 мегабайт

Дан неориентированный невзвешенный граф. Необходимо посчитать количество его компонент связности и вывести их.

### Формат входного файла

Во входном файле записано два числа  $N$  и  $M$  ( $0 < N \leq 100\,000$ ,  $0 \leq M \leq 100\,000$ ). В следующих  $M$  строках записаны по два числа  $i$  и  $j$  ( $1 \leq i, j \leq N$ ), которые означают, что вершины  $i$  и  $j$  соединены ребром.

### Формат выходного файла

В первой строчке выходного файла выведите количество компонент связности. Далее выведите сами компоненты связности в следующем формате: в первой строке количество вершин в компоненте, во второй — сами вершины в произвольном порядке.

### Пример

<code>components2.in</code>	<code>components2.out</code>
6 4	3
3 1	3
1 2	1 2 3
5 4	2
2 3	4 5
	1
	6

## Задача E. Поиск цикла

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

Дан ориентированный невзвешенный граф без кратных рёбер. Необходимо определить, есть ли в нём цикл.

### Формат входного файла

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

### Формат выходного файла

Если в графе нет цикла, то вывести «NO». Иначе в первой строке выведите слово «YES»,

во второй строке выведите количество вершин в цикле, в третьей строке выведите через пробел вершины, входящие в цикл, в порядке их следования.

### Примеры

<code>cycle.in</code>	<code>cycle.out</code>
2 2	YES
1 2	2
2 1	1 2
2 1	NO
1 2	

## Задача F. TopSort

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

Дан ориентированный невзвешенный граф. Необходимо его топологически отсортировать.

### Формат входного файла

В первой строке входного файла даны два натуральных числа  $N$  и  $M$  ( $1 \leq N \leq 10^5$ ,  $1 \leq M \leq 10^5$ ) — 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 выходного файла

Вывести любую топологическую сортировку графа в виде последовательности номеров вершин. Если граф невозможно топологически отсортировать, требуется вывести  $-1$ .

### Примеры

<code>topsort.in</code>	<code>topsort.out</code>
6 6	4 6 3 1 2 5
1 2	
3 2	
4 2	
2 5	
6 5	
4 6	
3 3	-1
1 2	
2 3	
3 1	