

Задача А. Количество ребер в неориентированном графе

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

Простой неориентированный граф задан матрицей смежности. Найти количество ребер в графе.

Формат входных данных

В первой строке входного файла дано число N ($1 \leq N \leq 100$). Затем идут N строк по N элементов в каждой — описание матрицы смежности.

Формат выходных данных

В выходной файл выведите единственное число — количество ребер в графе.

Примеры

<code>edges.in</code>	<code>edges.out</code>
3 0 1 0 1 0 1 0 1 0	2

Задача В. Количество ребер в ориентированном графе

Имя входного файла: edges2.in
Имя выходного файла: edges2.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Ориентированный граф задан матрицей смежности. Найдите количество ребер в графе.

Формат входных данных

В первой строке входного файла дано число N ($1 \leq N \leq 100$). Затем идут N строк по N элементов в каждой — описание матрицы смежности.

Формат выходных данных

В выходной файл выведите единственное число — количество ребер в графе.

Примеры

edges2.in	edges2.out
3 0 1 1 1 0 1 0 1 1	6

Задача С. Истоки и стоки

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

Вершина ориентированного графа называется истоком, если в нее не входит ни одно ребро, стоком, если из нее не выходит ни одного ребра. Ориентированный граф задан матрицей смежности. Найдите все вершины графа, которые являются истоками и все его вершины, которые являются стоками.

Формат входных данных

В первой строке входного файла дано число N ($1 \leq N \leq 100$). Затем идут N строк по N элементов в каждой — описание матрицы смежности.

Формат выходных данных

В первой строке выходного файла выведите число k — число истоков в графе и затем k чисел — номера вершин, которые являются истоками, в возрастающем порядке. На второй строке выведите информацию о стоках в том же формате.

Примеры

<code>flow.in</code>	<code>flow.out</code>
4	1 3
1 0 0 1	2 2 4
0 0 0 0	
1 1 0 1	
0 0 0 0	

Задача D. Петли

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

По заданной матрице смежности неориентированного графа определите, содержит ли он петли.

Формат входных данных

В первой строке входного файла дано число N ($1 \leq N \leq 100$). Затем идут N строк по N элементов в каждой — описание матрицы смежности.

Формат выходных данных

В выходной файл вывести «YES», если граф содержит петли, и «NO» в противном случае.

Примеры

loops.in	loops.out
3 0 1 1 1 0 1 1 1 0	NO
3 0 1 0 1 1 1 0 1 0	YES

Задача Е. Проверка на неориентированность

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

По заданной квадратной матрице $N \times N$ из нулей и единиц определить, может ли данная матрица быть матрицей смежности простого неориентированного графа.

Формат входных данных

В первой строке входного файла дано число N ($1 \leq N \leq 100$). Затем идут N строк по N элементов в каждой — описание матрицы смежности.

Формат выходных данных

В выходной файл вывести «YES», если граф неориентированный, и «NO» в противном случае.

Примеры

<code>orient.in</code>	<code>orient.out</code>
3 0 1 1 1 0 1 1 1 0	YES
3 0 1 1 1 0 1 0 1 0	NO

Задача F. От матрицы смежности к списку ребер

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

Простой неориентированный граф задан матрицей смежности, выведите его представление в виде списка ребер.

Формат входных данных

В первой строке входного файла дано число N ($1 \leq N \leq 100$). Затем идут N строк по N элементов в каждой — описание матрицы смежности.

Формат выходных данных

В выходной файл выведите список ребер, упорядоченный сначала по первой вершине в паре вершин, которая описывает ребро, а потом по второй.

Примеры

<code>tolist.in</code>	<code>tolist.out</code>
3	1 2
0 1 1	1 3
1 0 1	2 3
1 1 0	

Задача G. От матрицы смежности к списку ребер-2

Имя входного файла: `tolist2.in`
Имя выходного файла: `tolist2.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Ориентированный граф задан матрицей смежности, выведите его представление в виде списка ребер.

Формат входных данных

В первой строке входного файла дано число N ($1 \leq N \leq 100$). Затем идут N строк по N элементов в каждой — описание матрицы смежности.

Формат выходных данных

В выходной файл выведите список ребер, упорядоченный по первой вершине в паре вершин, которая описывает ребро, а потом по второй вершине.

Примеры

<code>tolist2.in</code>	<code>tolist2.out</code>
3	1 2
0 1 0	2 3
0 0 1	3 1
1 1 0	3 2

Задача Н. От списка ребер к матрице смежности

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

Простой неориентированный граф задан списком ребер, выведите его представление в виде матрицы смежности.

Формат входных данных

В первой строке входного файла заданы два целых числа N ($1 \leq N \leq 100$) — число вершин и M ($1 \leq M \leq N * (N - 1) / 2$) — число ребер. Далее в M строках содержатся M пар чисел, каждая из которых описывает одно ребро графа.

Формат выходных данных

В выходной файл выведите матрицу смежности графа.

Примеры

<code>tomatrix.in</code>	<code>tomatrix.out</code>
3 3	0 1 1
1 2	1 0 1
2 3	1 1 0
1 3	

Задача I. Степени вершин

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

Простой неориентированный граф задан матрицей смежности. Найдите степени всех вершин графа.

Формат входных данных

В первой строке входного файла дано число N ($1 \leq N \leq 100$). Затем идут N строк по N элементов в каждой — описание матрицы смежности.

Формат выходных данных

В выходной файл выведите N чисел — степени всех вершин.

Примеры

<code>vertexes.in</code>	<code>vertexes.out</code>
3	1
0 1 0	2
1 0 1	1
0 1 0	

Задача J. Поиск цикла

Имя входного файла: `cycle2.in`
Имя выходного файла: `cycle2.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

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

Формат входных данных

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 выходных данных

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

Примеры

<code>cycle2.in</code>	<code>cycle2.out</code>
2 2 1 2 2 1	YES 1 2
2 1 1 2	NO