Задача A. 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 входного файла

Во входном файле записано два числа N и M ($0 < N \leqslant 100$, $0 \leqslant M \leqslant \frac{N(N-1)}{2}$). В следующих M строках записаны по два числа i и j ($1 \leqslant i, j \leqslant N$), которые означают, что перекрестки i и j соединены тоннелем.

Формат выходного файла

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

Примеры

lights.in	lights.out
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 B. Компоненты связности

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

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

Формат входного файла

В первой строке входного файла содержится одно натуральное число N ($N \leqslant 100$) — 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.

Примеры

matrix.in	matrix.out
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 C. Компоненты связности - 2

Имя входного файла:	`matrix2.in`
Имя выходного файла:	`matrix2.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

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 ($0 < N \leqslant 100\,000$), $0 \leqslant M \leqslant 100\,000$). В следующих M строках записаны по два числа i и j ($1 \leqslant i, j \leqslant N$), 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.

Примеры

matrix2.in	matrix2.out
6 4 3 1 1 2 5 4 2 3	3 3 1 2 3 2 4 5 1 6

Задача D. От матрицы смежности к списку смежности, ориентированный вариант

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 списка смежности.

Формат входного файла

В первой строке файла — число N ($1 \leqslant N \leqslant 100$). Во второй и далее — матрица смежности.

Формат выходного файла

Выведите N строк: в i-й строке нужно через пробел записать номера всех соседей i-й вершины, упорядоченные по возрастанию. Вершины в задаче нумеруются от 1 до N.

Примеры

mtoal.in	mtoal.out
3 0 0 1 1 0 1 1 0 0	3 1 3 1

Задача E. Лесопосадки

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

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

Формат входного файла

В первой строке входного файла содержится одно натуральное число N ($N \leqslant 100$) — количество вершин в графе. Далее в N строках по N чисел — матрица смежности графа: в i-ой строке на j-ом 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, «`NO`» иначе.

Примеры

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

Задача F. Поиск цикла

Имя входного файла:	`cycle3.in`
Имя выходного файла:	`cycle3.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

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 ($1 \leqslant N \leqslant 100\,000$, $M \leqslant 100\,000$) — 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 — «`YES`».

Примеры

cycle3.in	cycle3.out
2 2 1 2 2 1	YES
2 1 1 2	NO

Задача G. TopSort

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

Дан ориентированный невзвешенный граф. Необходимо его топологически отсортировать.

Формат входного файла

В первой строке входного файла два натуральных числа N и M ($1 \leqslant N \leqslant 10^5, 1 \leqslant M \leqslant 10^5$) — количество вершин и рёбер в графе соответственно. Далее в M строках перечислены рёбра графа. Каждое ребро задается 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, вывести -1.

Примеры

topsort.in	topsort.out
6 6 1 2 3 2 4 2 2 5 6 5 4 6	4 6 3 1 2 5
3 3 1 2 2 3 3 1	-1