Задача A. От матрицы смежности к спискам смежности

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 списков смежности.

Формат входных данных

В первой строке файла находится число N — количество вершин графа ($1 \leqslant N \leqslant 100$). Во второй строке и далее — матрица смежности. Гарантируется, что граф не содержит петель.

Формат выходных данных

Выведите N строк — списки смежности графа. В i-й строке сначала выведите количество исходящих из i-й вершины рёбер, а затем — номера вершин, в которые эти рёбра идут, упорядоченные по возрастанию.

Примеры

mtoal.in	mtoal.out
5 0 0 1 0 0 1 0 1 0 0 0 0 0 0 1 1 1 0 0 0 1 1 0 0 0	1 3 2 1 3 1 5 2 1 2 2 1 2

Задача B. 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 входных данных

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. Гарантируется, что никакой тоннель не соединяет перекресток сам с собой, и не существует двух различных тоннелей, соединяющих одну и ту же пару вершин.

Формат выходных данных

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 C. Сделай дерево

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

Дан неориентированный связный граф с кратными ребрами. Найдите максимальный подграф, являющийся деревом.

Формат входных данных

В первой строке даны количество вершин N и ребер M ($1 \leqslant N, M \leqslant 100000$). В следующих M строках даны пары вершин $v1, v2$, являющимися концами ребер($1 \leqslant v1, v2 \leqslant N$).

Формат выходных данных

В первой строке выведите количество ребер в дереве. В каждых последующих строках выведите список ребер в формате, аналогичному входному файлу.

Примеры

maketree.in	maketree.out
3 3 1 2 2 3 1 3	2 2 3 1 2
2 2 1 2 1 2	1 1 2

Задача D. 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 ($1 \leqslant N \leqslant 100\,000$, $M \leqslant 100\,000$) — 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 выходных данных

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

Примеры

cycle2.in	cycle2.out
2 2 1 2 2 1	YES 1 2
2 1 1 2	NO

Задача E. Топологическая сортировка

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

Дан ориентированный невзвешенный граф. Необходимо его топологически отсортировать.

Формат входных данных

В первой строке входного файла даны два целых числа N и M ($1 \leqslant N \leqslant 100\,000, 0 \leqslant M \leqslant 100\,000$) — количества вершин и рёбер в графе соответственно. Далее в M строках перечислены рёбра графа. Каждое ребро задаётся парой чисел — номерами начальной и конечной вершин соответственно.

Формат выходных данных

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

Задача F. Кратчайший путь

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

Дан ориентированный взвешенный ациклический граф. Требуется найти в нем кратчайший путь из вершины s в вершину t.

Формат входных данных

Первая строка входного файла содержит четыре целых числа n, m, s и t — количество вершин, дуг графа, начальная и конечная вершина соответственно. Следующие m строк содержат описания дуг по одной на строке. Ребро номер i описывается тремя натуральными числами b_i, e_i и w_i — началом, концом и длиной дуги соответственно ($1 \leqslant b_i, e_i \leqslant n$, $|w_i| \leqslant 1000$).

Входной граф не содержит циклов и петель.

$1 \leqslant n \leqslant 100\,000$, $0 \leqslant m \leqslant 200\,000$.

Формат выходных данных

Первая строка выходного файла должна содержать одно целое число — длину кратчайшего пути из s в t. Если пути из s в t не существует, выведите «Unreachable».

Примеры

dag-shortpath.in	dag-shortpath.out
2 1 1 2 1 2 -10	-10
2 1 2 1 1 2 -10	Unreachable