

Задача А. От списка ребер к матрице смежности, ориентированный вариант

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

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

Формат входных данных

На вход программы поступают числа n ($1 \leq n \leq 100$) — количество вершин в графе и m ($1 \leq m \leq n(n-1)$) — количество ребер. Затем следует m пар чисел — ребра графа.

Формат выходных данных

Выведите матрицу смежности заданного графа.

Пример

стандартный ввод	стандартный вывод
5 3	0 0 1 0 0
1 3	0 0 1 0 0
2 3	0 0 0 0 0
5 2	0 0 0 0 0
	0 1 0 0 0

Задача В. Степени вершин по списку ребер

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Неориентированный граф задан списком ребер. Найдите степени всех вершин графа.

Формат входных данных

Сначала вводятся числа n ($1 \leq n \leq 100$) — количество вершин в графе и m ($1 \leq m \leq \frac{n(n-1)}{2}$) — количество ребер. Затем следует m пар чисел — ребра графа. Каждая вершина — число от 1 до n .

Формат выходных данных

Выведите n чисел — степени вершин графа. i -ое выведенное число должно быть равно степени вершины с номером i .

Пример

стандартный ввод	стандартный вывод
5 3 1 3 2 3 2 5	1 2 2 0 1

Задача С. Обход графа

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Дан неориентированный невзвешенный граф. Для него вам необходимо найти количество вершин, лежащих в одной компоненте связности с данной вершиной (считая эту вершину).

Формат входных данных

В первой строке входных данных содержатся два числа: N и S ($1 \leq N \leq 100$; $1 \leq S \leq N$), где N — количество вершин графа, а S — заданная вершина. В следующих N строках записано по N чисел — матрица смежности графа, в которой 0 означает отсутствие ребра между вершинами, а 1 — его наличие. Гарантируется, что на главной диагонали матрицы всегда стоят нули.

Формат выходных данных

Выведите одно целое число — искомое количество вершин.

Пример

стандартный ввод	стандартный вывод
3 1 0 1 1 1 0 0 1 0 0	3

Задача D. Дерево?

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Имеется неориентированный граф, состоящий из N вершин и M ребер. Необходимо проверить, является ли граф деревом. Напомним, что дерево — это связный граф, в котором нет циклов (следовательно, между любой парой вершин существует ровно один простой путь). Граф называется связным, если от одной вершины существует путь до любой другой.

Формат входных данных

Во входном файле в первой строке содержатся два целых числа N и M ($1 \leq N \leq 100$, $0 \leq M \leq 1000$), записанные через пробел. Далее следуют M различных строк с описаниями ребер, каждая из которых содержит два натуральных числа A_i и B_i ($1 \leq A_i, B_i \leq N$), где A_i и B_i — номера вершин, соединенных i -м ребром.

Формат выходных данных

В выходной файл выведите слово «YES», если граф является деревом, или «NO» в противном случае.

Примеры

стандартный ввод	стандартный вывод
3 2 1 2 1 3	YES
4 3 1 2 1 3 2 3	NO

Задача Е. Компоненты связности

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

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

Формат входных данных

Во входном файле записано два числа N и M ($0 < N \leq 100000, 0 \leq M \leq 100000$). 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 выходных данных

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 отсортированном порядке.

Пример

стандартный ввод	стандартный вывод
6 4	3
3 1	3
1 2	1 2 3
5 4	2
2 3	4 5
	1
	6

Задача F. Долой списывание!

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

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

У Михаила Дмитриевича записаны все пары лкшат, обменявшихся записками. Требуется определить, сможет ли он разделить лкшат на две группы так, чтобы любой обмен записками осуществлялся от лкшонка одной группы лкшонку другой группы.

Формат входных данных

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

Формат выходных данных

Необходимо вывести ответ на задачу Павла Олеговича. Если возможно разделить лкшат на две группы, выведите «YES»; иначе выведите «NO».

Пример

bipartite.in	bipartite.out
3 2 1 2 2 3	YES

Задача G. Есть ли цикл?

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Дан ориентированный граф. Требуется определить, есть ли в нем цикл.

Формат входных данных

В первой строке вводится число n - количество вершин и m - количество ребер. ($1 \leq n, m \leq 10^5$).
Далее в m строках следует по 2 числа u, v - вершины графа, соединенные ребром.

Формат выходных данных

Выведите 0, если в заданном графе нет цикла, и 1, если он есть.

Примеры

стандартный ввод	стандартный вывод
4 4 1 2 2 3 3 4 4 1	1
3 2 1 2 1 3	0