

Задача А. Обход в глубину

Ограничение по времени: 2 секунда
Ограничение по памяти: 64 мегабайта

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.

Примеры

dfs.in	dfs.out
3 1 0 1 0 1 0 0 0 0 0	2

Задача В. Компоненты связности

Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

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

Формат входных данных

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 -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 выходных данных

Вывести одно целое число — искомое количество компонент связности графа.

Пример

comp.in	comp.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 С. Яблоки от яблони...

Ограничение по времени: 2 секунда
Ограничение по памяти: 64 мегабайта

У Пети в саду растет яблоня. Воодушевленный историей об Исааке Ньютоне, который, как известно, открыл закон всемирного тяготения после того, как ему на голову упало яблоко, Петя с целью повысить свою успеваемость по физике часто сидит под яблоней.

Однако, поскольку по физике у Пети твердая тройка, яблоки с его яблони падают следующим образом. В какой-то момент одно из яблок отрывается от ветки, на которой оно висит, и начинает падать строго вниз. Если в некоторый момент оно задевает другое яблоко, то то тоже отрывается от своей ветки и начинает падать вниз, при этом первое яблоко не меняет направление своего падения. Вообще, если любое падающее яблоко заденет другое яблоко на своем пути, то оно также начнет падать.

Таким образом, в любой момент каждое яблоко либо висит на ветке, либо падает строго вниз, причем все яблоки кроме первого, чтобы начать падать, должны сначала соприкоснуться с каким-либо другим падающим яблоком.

Выясните, какие яблоки упадут с Петиной яблони.

Формат входных данных

В первой строке вводится число N — количество яблок на Петиной яблоне ($1 \leq N \leq 200$). Следующие N строк содержат описание яблок. Будем считать все яблоки шарами. Каждое яблоко задается координатами своей самой верхней точки (той, где оно исходно прикреплено к дереву, длиной черенка пренебрежем) x_i , y_i и z_i и радиусом r_i ($-10000 \leq x_i, y_i, z_i \leq 10000$, $1 \leq r_i \leq 10000$, все числа целые). Гарантируется, что изначально никакие яблоки не пересекаются (даже не соприкасаются). Ось OZ направлена вверх.

Формат выходных данных

В первой строке выведите количество яблок, которые упадут с яблони, если начнет падать первое яблоко. В следующей строке выводите номера упавших яблок. Яблоки нумеруются, начиная с 1, в том порядке, в котором они заданы во входных данных. Номера яблок выведите в порядке возрастания.

Примеры

apples.in	apples.out
4	3
0 0 10 4	1 2 4
5 0 3 1	
-7 4 7 1	
0 1 2 6	

Задача D. Бюрократия

Ограничение по времени: 2 секунда
Ограничение по памяти: 64 мегабайта

Некогда гордая и свободолюбивая республика Акадия совсем погрязла в коррупции. Депутаты покупают и перепродают голоса друг друга, чтобы затем принять или отклонить какие-то законы. К принятым законам довольно быстро появляются поправки, их аннулирующие. Эти поправки, в свою очередь, аннулируются следующими поправками, и так далее. Секретарь премьер-министра хочет разобраться, какие законы в данный момент действуют, а какие нет.

С незапамятных времён в Акадии всего два типа законов:

- прямой закон, устанавливающий новые правовые нормы;
- поправка, аннулирующая какой-то из предыдущих законов.

Закон считается действующим тогда и только тогда, когда нет действующего закона, который бы являлся поправкой, его аннулирующей.

В распоряжении секретаря — все законы Акадии с момента становления республики в порядке их принятия. Помогите ему выяснить, какие из законов — действующие.

Формат входных данных

В первой строке ввода записано n ($1 \leq n \leq 100000$) — количество принятых законов. Следующие n строк описывают сами законы. Каждое описание имеет следующий вид:

- `declare`, если данный закон — это прямой закон.
- `cancel i`, если данный закон — это поправка, аннулирующая закон с номером i .

Законы нумеруются, начиная с единицы.

Формат выходных данных

В первой строке вывода выведите одно число — количество действующих законов. Далее выведите номера всех действующих законов в порядке возрастания.

Примеры

bureaucr.in	bureaucr.out
5 declare cancel 1 declare cancel 2 cancel 3	3 1 4 5