

Задача А. Обрати меня!

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

Мальчик Вася очень любит разворачивать ориентированные графы. Помогите ему в этом.

Формат входных данных

Во входном файле записано число N ($1 \leq N \leq 50\,000$) — количество вершин в графе. В следующих N строках записан граф в виде списков смежности: в i -й строке, начале записано количество вершин, смежных с вершиной под номером i , затем в порядке возрастания записаны номера вершин, в которые идут рёбра из i -й вершины. Нумерация начинается с единицы. Гарантируется, что рёбер в графе не более 50 000.

Формат выходных данных

Выведите развёрнутый граф в том же формате, что и исходный.

Примеры

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

Задача В. Кратчайшее расстояние

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

Дан ориентированный граф. Найдите расстояния от вершины x до всех остальных вершин графа.

Формат входных данных

В первой строке входного файла содержатся два натуральных числа N и x ($1 \leq N \leq 1000$, $1 \leq x \leq N$) — количество вершин в графе и стартовая вершина соответственно. 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.

Формат выходных данных

Выведите через пробел числа d_1, d_2, \dots, d_n , где d_i — это -1 , если путей между x и i нет, и минимальное расстояние между x и i в противном случае.

Примеры

<code>mindist.in</code>	<code>mindist.out</code>
6 5 0 1 1 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 0 0 1 0 0 0 0	2 2 1 1 0 -1

Задача С. Кратчайший путь

Имя входного файла: mindist2.in
Имя выходного файла: mindist2.out
Ограничение по времени: 3 секунды
Ограничение по памяти: 64 мегабайта

Вам дан неориентированный граф. Найдите кратчайший путь от вершины a до вершины b .

Формат входных данных

В первой строке входного файла идут целые числа n и m ($1 \leq n \leq 50\,000$, $1 \leq m \leq 100\,000$) — количества вершин и рёбер соответственно. Во второй строке идут целые числа a и b — стартовая и конечная вершины соответственно. Далее идут m строк, описывающих рёбра.

Формат выходных данных

Если пути между a и b нет, выведите единственное число -1 . Иначе выведите в первой строке число l — длину кратчайшего пути между этими двумя вершинами в рёбрах, а во второй строке выведите $l + 1$ число — вершины этого пути.

Примеры

mindist2.in	mindist2.out
4 5 1 4 1 3 3 2 2 4 2 1 2 3	2 1 2 4
4 4 2 3 2 1 2 4 4 3 1 3	2 2 1 3

Задача D. Шайтан-машинка

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

У Ибрагима есть магическая чёрная шайтан-машинка. На ней есть три кнопки и табло. Табло может показывать не более чем четырёхзначные числа. Каждая из кнопок меняет число некоторым образом: первая домножает его на 3, вторая прибавляет к нему сумму его цифр, а третья вычитает из него 2. В случае, если число становится отрицательным или превосходит 9 999, шайтан-машинка ломается. Ибрагим может нажимать кнопки в любом порядке. Он хочет узнать, как получить на табло число b после некоторой последовательности нажатий, если сейчас шайтан-машинка показывает a . Помогите ему найти минимальное необходимое число нажатий.

Формат входных данных

Единственная строка входного файла содержит два натуральных числа a и b , разделённых пробелом ($1 \leq a, b \leq 9\,999$).

Формат выходных данных

Выведите одно число — минимальное необходимое количество действий.

Примеры

<code>crazycalc.in</code>	<code>crazycalc.out</code>
14 45	3
18 12	3
14 29	2

Задача Е. Детали

Имя входного файла: `details.in`
Имя выходного файла: `details.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 64 мегабайта

Предприятие «Авто-2005» выпускает двигатели для известных во всём мире автомобилей. Двигатель состоит ровно из n деталей, пронумерованных от 1 до n , при этом деталь с номером i изготавливается за p_i секунд. Специфика предприятия «Авто-2005» заключается в том, что там одновременно может изготавливаться лишь одна деталь двигателя. Для производства некоторых деталей необходимо иметь предварительно изготовленный набор других деталей.

Генеральный директор «Авто-2005» поставил перед предприятием амбициозную задачу — за наименьшее время изготовить деталь с номером 1, чтобы представить её на выставке.

Требуется написать программу, которая по заданным зависимостям порядка производства между деталями найдёт наименьшее время, за которое можно произвести деталь с номером 1.

Формат входных данных

Первая строка написано натуральное число n ($1 \leq n \leq 100\,000$). Вторая — n натуральных чисел p_1, p_2, \dots, p_n , определяющих время изготовления каждой детали в секундах ($1 \leq p_i \leq 1\,000\,000\,000$).

Каждая из последующих n строк входного файла описывает характеристики производства деталей. Здесь i -я строка содержит в начале количество деталей, которые требуются для производства детали с номером i , затем через пробел номера самих деталей. В списке нет повторяющихся номеров деталей. Сумма длин всех списков не превосходит 200 000.

Известно, что не существует циклических зависимостей в производстве деталей.

Формат выходных данных

В единственной строке выходного файла должно содержаться одно число: минимальное время (в секундах), необходимое для скорейшего производства детали с номером 1.

Примеры

details.in	details.out
3 100 200 300 1 2 0 2 2 1	300