

Задача А. Истоки и стоки

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

Вершина ориентированного графа называется истоком, если в нее не входит ни одно ребро, и стоком, если из нее не выходит ни одного ребра.

Ориентированный граф задан матрицей смежности. Найдите все его вершины-истоки и все вершины-стоки.

Формат входных данных

N — число вершин в графе ($1 \leq N \leq 100$), затем матрица смежности — N строк по N чисел, каждое из которых равно 0 или 1.

Формат выходных данных

В первой строке выведите K — число истоков в графе, затем номера вершин, являющихся истоками в порядке возрастания. Во второй строке выведите информацию о стоках в том же формате.

Примеры

<code>source.in</code>	<code>source.out</code>
5 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0	2 3 4 3 1 4 5

Задача В. 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 ($1 \leq N, M \leq 100000$). В следующих M строках даны пары вершин v_1, v_2 , являющимися концами ребер ($1 \leq v_1, v_2 \leq N$).

Формат выходных данных

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

Примеры

<code>maketree.in</code>	<code>maketree.out</code>
3 3 1 2 2 3 1 3	2 2 3 1 2
2 2 1 2 1 2	1 1 2

Задача С. Количество достижимых вершин

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

Вам дан ориентированный граф, заданный списками смежности. Он состоит из N вершин. Посчитайте количество вершин, достижимых из вершины с номером S . Любая вершина считается достижимой из самой себя.

Формат входных данных

В первой строке входного файла записано два числа N и S ($1 \leq N \leq 10^4$, $1 \leq S \leq N$). Далее идут N строк. В i -й строке содержится количество вершин, смежных с вершиной i , и номера этих вершин. Все вершины нумеруются натуральными числами от 1 до N . Количество рёбер в графе не превышает 10^5 .

Формат выходных данных

Выведите в выходной файл одно число — количество вершин, достижимых из вершины с номером S .

Примеры

<code>reach.in</code>	<code>reach.out</code>
4 1 1 4 2 1 4 2 1 4 1 3	3

Задача D. 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>
<pre>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</pre>	<pre>2 2 1 1 0 -1</pre>

Задача Е. Приключения шахматного коня

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

На шахматной доске $N \times N$ в клетке (x_1, y_1) стоит голодный шахматный конь. Он хочет попасть в клетку (x_2, y_2) , где растет вкусная шахматная трава. Какое наименьшее количество ходов он должен для этого сделать?

Формат входных данных

На вход программы поступает пять чисел: N , x_1 , y_1 , x_2 , y_2 ($5 \leq N \leq 20$, $1 \leq x_1, y_1, x_2, y_2 \leq N$). Левая верхняя клетка доски имеет координаты $(1, 1)$, правая нижняя - (N, N) .

Формат выходных данных

В первой строке выведите единственное число K - наименьшее необходимое число ходов коня. В каждой из следующих $K + 1$ строк должно быть записано 2 числа - координаты очередной клетки в пути коня.

Примеры

<code>knight.in</code>	<code>knight.out</code>
5	2
1 1	1 1
3 2	3 2

Задача F. Числа

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

Витя хочет придумать новую игру с числами. В этой игре от игроков требуется преобразовывать четырехзначные числа не содержащие нулей при помощи следующего разрешенного набора действий:

1. Можно увеличить первую цифру числа на 1, если она не равна 9.
2. Можно уменьшить последнюю цифру на 1, если она не равна 1.
3. Можно циклически сдвинуть все цифры на одну вправо.
4. Можно циклически сдвинуть все цифры на одну влево.

Например, применяя эти правила к числу 1234, можно получить числа 2234, 1233, 4123 и 2341 соответственно.

Точные правила игры Витя пока не придумал, но пока его интересует вопрос, как получить из одного числа другое за минимальное количество операций.

Формат входных данных

Во входном файле содержится два различных четырехзначных числа, каждое из которых не содержит нулей.

Формат выходных данных

Программа должна вывести последовательность четырехзначных чисел, не содержащих нулей. Последовательность должна начинаться первым из данных чисел и заканчиваться вторым из данных чисел, каждое последующее число в последовательности должно быть получено из предыдущего числа применением одного из правил. Количество чисел в последовательности должно быть минимально возможным.

Примеры

<code>numbers.in</code>	<code>numbers.out</code>
9876	9876
8876	8769
	8768
	8876

Задача G. Доставка кефирчика

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

Во время проведения очередной Межгалактической Летней Компьютерной Школы (МЛКШ) организаторы столкнулись с проблемой доставки кефирчика для вечерки. Дело в том, что кефирчик производят на планете под номером 1, а сами школьники живут на планете n , поэтому на доставку кефирчика тратится довольно большое время, а значит он успевает испортиться.

К счастью, галактическая транспортная система «Берендеев-Экспресс» постепенно внедряет новые кефиропроводы, способные передавать кефир со скоростью, в два раза превышающей скорость старых моделей. А именно, с любой планеты на любую по старым кефиропроводам кефир проходит за два года, а по новым — за один.

Разумеется, грешно было бы не воспользоваться инновационными технологиями, поэтому директор МЛКШ попросил вас написать программу, которая по данным о имеющихся кефиропроводах (как новых, так и старых) узнает кратчайший путь от планеты 1 до планеты n .

Формат входных данных

В первой строке входного файла даны два целых числа n и m ($1 \leq n \leq 100\,000$, $0 \leq m \leq 100\,000$) — количество планет и количество кефиропроводов соответственно. В последующих m строках даны тройки натуральных чисел u_i , v_i и c_i . Числа u_i и v_i обозначают номера планет, соединенных i -м кефиропроводом, а c_i ($c_i = 1$ или $c_i = 2$) — количество лет, которое потребуется, чтобы передать кефир с одной планеты на другую через i -й кефиропровод. Планеты во входном файле нумеруются с единицы. Кефир по трубопроводам можно передавать в обоих направлениях.

Формат выходных данных

В выходной файл требуется вывести одно число — количество лет, которое требуется, чтобы доставить кефир с планеты 1 на планету n . Если доставка невозможна, то в выходной файл требуется вывести «-1».

Примеры

<code>kefir.in</code>	<code>kefir.out</code>
3 2 1 2 2 2 3 1	3
3 1 2 3 1	-1
2 5 1 2 1 1 2 2 1 2 1 1 1 2 2 2 1	1