Задача A. Игра в пьяницу

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

В игре в пьяницу карточная колода раздаётся поровну двум игрокам. Далее они вскрывают по одной верхней карте, и тот, чья карта старше, забирает себе обе вскрытые карты, которые кладутся под низ его колоды. Тот, кто остаётся без карт — проигрывает.

Для простоты будем считать, что все карты различны по номиналу, а также, что самая младшая карта побеждает самую старшую карту («шестерка берет туза»).

Игрок, который забирает себе карты, сначала кладёт под низ своей колоды карту первого игрока, затем карту второго игрока (то есть карта второго игрока оказывается внизу колоды).

Напишите программу, которая моделирует игру в пьяницу и определяет, кто выигрывает. В игре участвует n карт, имеющих значения от 0 до $n-1$, большая карта побеждает меньшую, карта со значением 0 побеждает карту $n-1$.

Формат входного файла

Программа получает на вход три строки. В первой строке содержится целое чётное число n ($2 \leqslant n \leqslant 100\,000$). Вторая строка содержит $\frac{n}{2}$ чисел — карты первого игрока, а третья — $\frac{n}{2}$ карт второго игрока. Карты перечислены сверху вниз, то есть каждая строка начинается с той карты, которая будет открыта первой. Гарантируется, что каждая из карт встречается в колодах игроков ровно один раз.

Формат выходного файла

Программа должна определить, кто выигрывает при данной раздаче, и вывести слово «`first`» или «`second`», после чего вывести количество ходов, сделанных до выигрыша. Если на протяжении $2 \cdot 10^5$ ходов игра не заканчивается, программа должна вывести слово «`draw`».

Примеры

card-game.in	card-game.out
10 1 3 5 7 9 2 4 6 8 0	second 5

Задача B. Обход в ширину

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

Дан неориентированный граф. В нём необходимо найти расстояние от одной заданной вершины до другой.

Формат входного файла

В первой строке входного файла содержатся три натуральных числа N, S и F ($1 \leqslant S, F \leqslant N \leqslant 100$) — количество вершин в графе и номера начальной и конечной вершин соответственно. Далее в N строках задана матрица смежности графа. Если значение в j-м элементе i-й строки равно 1, то в графе есть направленное ребро из вершины i в вершину j.

Формат выходного файла

В единственной строке должно находиться минимальное расстояние от начальной вершины до конечной. Если пути не существует, выведите 0.

Примеры

bfs.in	bfs.out
5 5 3 0 0 1 1 0 0 0 0 1 0 1 0 0 0 1 1 1 0 0 1 0 0 1 1 0	1

Задача C. 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 входного файла

В первой строке входного файла даны два целых числа n и m ($1 \leqslant n \leqslant 100\,000$, $0 \leqslant m \leqslant 100\,000$) — 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 выходного файла

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

kefir.in	kefir.out
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

Задача D. Числа

Имя входного файла:	numbers.in
Имя выходного файла:	numbers.out
Ограничение по времени:	2 секунды
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 входного файла

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

Формат выходного файла

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

numbers.in	numbers.out
1234 4321	1234 2234 3234 4323 4322 4321
2345 2344	2345 2344