Задача A. Очередь

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

Для каждой операции изъятия элемента выведите ее результат.

На вход программе подаются строки, содержащие команды. Каждая строка содержит одну команду. Команда — это либо "`+ N`", либо "`-`". Команда "`+ N`" означает добавление в очередь числа N, по модулю не превышающего 10^9. Команда "`-`" означает изъятие элемента из очереди.

Формат входных данных

В первой строке содержится количество команд — M ($1 \leqslant M \leqslant 10^6$). В последующих строках содержатся команды, по одной в каждой строке.

Формат выходных данных

Выведите числа, которые удаляются из очереди, по одному в каждой строке. Гарантируется, что изъятий из пустой очереди не производится.

Примеры

queue.in	queue.out
4 + 1 + 10 - -	1 10

Замечание

Разрешается использовать deque

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 входных данных

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 ребро из вершины i в вершину j.

Формат выходных данных

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

Примеры

bfs.in	bfs.out
5 4 2 0 1 0 1 1 1 0 1 0 0 0 1 0 1 0 1 0 1 0 0 1 0 0 0 0	2

Задача C. 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 входных данных

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 выходных данных

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 D. Путь

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

В неориентированном графе требуется найти минимальный путь между двумя вершинами.

Формат входных данных

Первым на вход программе поступает целое число N — количество вершин в графе ($1 \leqslant N \leqslant 100$). Затем записана матрица смежности («0» обозначает отсутствие ребра, «1» — наличие ребра). Далее задаются номера двух вершин — начальной и конечной. Гарантируется, что матрица смежности симметрична относительно главной диагонали, а граф не содержит петель.

Формат выходных данных

Требуется вывести сначала L — длину кратчайшего пути (количество рёбер, которые нужно пройти), а затем $L + 1$ число — путь от одной вершины до другой, заданный своими вершинами. Если пути не существует, выведите «-1».

Примеры

path.in	path.out
5 0 1 0 0 1 1 0 1 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 3 5	3 3 2 1 5

Задача E. Путь конём

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

На шахматной доске 8×8 указаны две различные клетки. Найдите кратчайший маршрут коня из первой клетки во вторую.

Формат входных данных

Во входном файле записаны координаты двух клеток. Каждая координата представлена двумя символами, где сначала указана одна строчная буква от `a` до `h`, а после буквы (без пробела) цифра от `1` до `8`, например `h8`. Каждая клетка записана в отдельной строке. Гарантируется, что координаты клеток различны.

Формат выходных данных

Программа должна вывести последовательность клеток, первая из которых совпадает с первой данной, а последняя совпадает со второй данной. Две соседние клетки должны быть соединены ходом коня, при этом количество клеток в последовательности должно быть минимально возможным. Если существует несколько возможных ответов на задачу, разрешается выводить любой.

Примеры

knight.in	knight.out
a1 b1	a1 b3 d2 b1

Замечание

При решении задачи полезно знать про функции `ord()` и `chr()`:
`ord(c)` - возвращает код символа `c`
`chr(x)` - возвращает символ с кодовым значением `x`
Например: `ord('a') = 97`, `chr(98) = 'b'`