Задача 1. Недоступные области

Источник:	базовая I
Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Со времен фараоновых гробниц лабиринты строили весьма умные люди. Как правило они старались сделать так, чтобы при «посетитель» заведомо не мог попасть в самые интересные места, — скажем, в те комнаты, где хранятся главные сокровища.

В этой задаче по имеющемуся плану лабиринта нужно определить, есть ли в нем недоступные области.

После входа в лабиринт посетитель может гулять по нему как ему вздумается, **не** может разве что проходить сквозь стены и ходить вне лабиринта. На каждом шаге посетитель может переместиться из текущей клетки в любую соседнюю с ней по стороне пустую клетку. Ходить по диагонали нельзя.

Формат входных данных

В первой строке входного файла задано одно целое число N — размер лабиринта (лабиринт у нас квадратный, $3 \leqslant N \leqslant 1\,000$). Далее N строк содержат по N символов, описывающих сам лабиринт: стенки изображаются звездочками, а пустоты — пробелами.

Вход в лабиринт — это единственная пустая клетка в первой строке.

Формат выходных данных

В выходной файл нужно вывести одно целое число — количество недоступных областей лабиринта. Недоступные области считаются различными, если из одной нельзя пройти в другую.

Пример

input.txt	output.txt
9 *** ***** * * * ** * * * * * * * * * *** * * * * * * * ***** * * * *********	2

Комментарий

При рекурсивном обходе большого графа нужен довольно большой программный стек — явно больше, чем 1 мегабайт по умолчанию. При решении на Visual C, настоятельно рекомендуется увеличить стек, добавив следующую строчку в самом начале программы:

```
#pragma comment(linker, "/STACK:50000000")
```

Задача 2. BFS за линию

Источник:	базовая II
Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Дан ориентированный граф G без весов. В графе есть V вершин, занумерованных числами от 1 до N, и E дуг между некоторыми парами вершин. Требуется найти длины кратчайших путей от вершины номер 1 до всех вершин.

Формат входных данных

В первой строке входного файла находятся целые числа M и N — количество вершин и дуг в графе соответственно ($2 \leqslant N \leqslant 200\,000$, $1 \leqslant M \leqslant 200\,000$).

Следующие M строк содержат по два числа u и v ($1 \leqslant u, v \leqslant N$), означающие, что в графе есть дуга из вершины u в вершину v. По дуге можно перемещаться только в прямом направлении.

Формат выходных данных

В выходном файле требуется вывести ровно N строк. На i-й строке должно быть целое число:

- "-1", если не существует пути от вершины 1 до вершины номер i.
- Минимальное количество рёбер в пути от вершины 1 до вершины номер i.

Пример

input.txt	output.txt
5 5 1 2 2 3 3 1 2 1 3 4	0 1 2 3 -1

Задача 3. Побег из лабиринта

Источник:	базовая II
Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Для заданного лабиринта найти кратчайший путь от входа до выхода.

Формат входных данных

В первой строке входного файла находятся целые числа M и N — высота и ширина лабиринта ($1 \leqslant M, N \leqslant 100$). Каждая из следующих M строк содержит N символов, при этом символ '.' обозначает пустую клетку, символ 'X' — блоки, символ 'S' — начальную клетку, символ 'F' — конечную клетку.

Формат выходных данных

Выведите в выходной файл минимальное число шагов, за которое можно добраться от начальной клетки до конечной, каждый раз переходя на соседнюю по стороне клетку и не ступая на блоки, либо число -1, если это невозможно.

Выходить за пределы лабиринта нельзя.

Пример

input.txt	output.txt
6 7 .....X. .X..XF. ..XXXX. .X..... .X..X.. S...X..	11

Задача 4. Система неравенств

Источник:	основная I
Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Дана система из n переменных и m неравенств. Переменные этой системы обозначаются через X_k для $1 \leqslant k \leqslant n$. Каждое неравенство имеет вид: $X_i < X_j$ (для некоторых индексов i и j).

Требуется найти такое решение системы, что:

1. все неравенства системы удовлетворены;
2. значение каждой переменной является натуральным числом, не превышающим n;
3. значения всех переменных различны.

Следует заметить, что в данной задаче ноль не считается натуральным числом.

ВАЖНО: Ваше решение должно работать за время $O(m + n)$.

Формат входных данных

В первой строке содержатся два целых числа n и m, где n — количество переменных, m — количество неравенств ($2 \leqslant n \leqslant 10^5$, $1 \leqslant m \leqslant 2 \cdot 10^5$).

В следующих m строках записаны неравенства, по одному в строке. Для каждого неравенства записано два целых числа i и j ($1 \leqslant i, j \leqslant n$), что обозначает неравенство: $X_i < X_j$.

Формат выходных данных

Если искомое решение существует, то в первой строке должно быть записано слово `YES`, а во второй — само решение. Решение выводится в виде n целых чисел, определяющих значения переменных X_1, X_2, X_3, …, X_n. Если решений нет, выведите слово `NO` в единственной строке.

Если подходящих решений несколько, выведите любое из них.

Примеры

input.txt	output.txt
6 5 1 2 3 2 1 3 6 5 5 4	YES 1 3 2 6 5 4
6 5 1 2 2 3 3 1 6 5 5 4	NO
3 2 2 3 3 1	YES 3 1 2

Задача 5. Найти цикл

Источник:	основная I
Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Дан ориентированный граф. Требуется найти в этом графе простой цикл, если он существует.

Цикл — это замкнутый маршрут, идущий по рёбрам графа. Цикл называется простым, если он проходит через каждую вершину не более одного раза (за один круг).

Формат входных данных

В первой строке содержатся два целых числа N и M, где N — количество вершин, M — количество рёбер ($1 \leqslant N, M \leqslant 2 \cdot 10^5$).

В следующих M строках записаны ориентированные рёбра, по одному в строке. Для каждого ребра записано два целых числа: u_j — номер начальной вершины и v_j — номер конечной вершины ($1 \leqslant u_j \neq v_j \leqslant N$).

Формат выходных данных

Если в графе нет циклов, выведите одно целое число `-1`.

Иначе выведите **любой** простой цикл в виде последовательности вершин. В первой строке требуется записать K — количество вершин в цикле ($2 \leqslant K \leqslant N$). Во втором строке должно быть записано K различных целых чисел X_i — номера вершин цикла в порядке их прохождения ($1 \leqslant X_i \leqslant N$).

Примеры

input.txt	output.txt
6 5 1 2 3 2 1 3 6 5 5 4	-1
6 5 1 2 2 3 3 1 6 5 5 4	3 2 3 1

Задача 6. Фиолетовое такси

Источник:	основная II
Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

«Фиолетовое такси» предложило клиентам новую услугу — теперь можно узнать кратчайшее время проезда для любых мест отправления и назначения. Однако диспетчеры не успевают отвечать на все запросы. Напишите программу, которая поможет диспетчерам быстро находить минимальное время пути. Учтите, что в редких случаях клиента интересует не только время, но и подробное описание предполагаемого маршрута.

Формат входных данных

В первой строке входного файла записано четыре целых числа: N — количество пунктов, в которых люди садятся и выходят из такси, M — количество дорог, их соединяющих, P — количество запросов на поиск кратчайшего пути, K — количество запросов на поиск минимального времени ($1 \leqslant N, P \leqslant 300$, $1 \leqslant M, K \leqslant 50\,000$).

Далее в M строках описываются дороги, по три целых числа в каждой строке — номера двух пунктов, соединенных этой дорогой и время L_i проезда по дороге ($1 \leqslant L_i \leqslant 10^6$).

Пункты пронумерованы числами от 1 до N. Все дороги двусторонние. Пару пунктов **может** соединять несколько дорог. Гарантируется, что все пункты соединены между собой.

Далее следуют запросы, по одному запросу в строке. Сначала записаны запросы на поиск кратчайшего пути (P штук), затем запросы на поиск минимального времени (K штук). Каждый запрос описан двумя целыми числами S_j и T_j — номер пункта отправления и номер пункта назначения соответственно ($1 \leqslant S_j \neq T_j \leqslant N$).

Формат выходных данных

Для каждого запроса нужно вывести ответ на отдельной строке. Во-первых, нужно вывести минимальное время пути от пункта отправления до пункта назначения. Во-вторых, для запроса на поиск кратчайшего пути нужно вывести дополнительно $(A_j + 1)$ целых чисел — описание оптимального маршрута. Первое из этих чисел должно быть равно A_j — количеству вершин в пути (включая пункты отправления и назначения), а остальные числа должны задавать номера пунктов, по которым идёт маршрут, в порядке их прохождения.

Пример

input.txt	output.txt
5 6 3 2 4 2 2 1 4 8 2 3 6 1 5 7 2 1 3 4 3 9 3 5 1 3 3 1 4 3 5 4	16 4 3 2 1 5 9 3 1 2 3 9 3 3 2 1 8 12

Задача 7. COVID-19: Карантин

Источник:	основная II
Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Вовочка грустит: из-за пандемии коронавируса его город закрыли на «самоизоляцию», и его лично — на 14-дневный карантин. Теперь он не может выходить из дома без необходимости. Если он попробует, то его поймают суровые дядечки и упекут в инфекционный стационар. А Вовочке очень хочется встретиться с друзьями и рассказать им о своей заграничной поездке! Будучи от природы человеком безответственным и безрассудным, Вовочка нашёл выход из положения.

Оказывается, выгуливать собак никто не запрещает даже тем, кто на карантине! К счастью, у Вовочки есть крупная и хорошо дрессированная собака по кличке Слоняша, с которой он может свободно гулять. Вот только до некоторых друзей идти далеко, а опыт показывает, что если Слоняша долго не делает того, что обычно должны делать выгуливаемые собаки, то наблюдатели начинают подозревать неладное и быстро отправляют Вовочку домой. Таким образом, если Вовочка хочет пойти к кому-то в гости, ему надо составить такой маршрут, чтобы на протяжении всего пути Слоняша была непрерывно занята делом. А чтобы пополнять запасы драгоценной жидкости, Слоняша может пить воду из луж — благо на дворе весна и луж на дорогах предостаточно.

Вовочка решил прибегнуть к помощи вычислительной техники для составления плана. Он смоделировал проблему следующим образом. Есть ориентированный граф, рёбра которого представляют собой дорожки на улице (да, его родной город настолько суров, что на всех пешеходных дорожках одностороннее движение). Каждому ребру приписан вес, который определяет, сколько миллилитров жидкости нужно Слоняше на его качественное преодоление. Вес ребра рассчитывается за вычетом той воды, которую можно выпить из расположенных на дорожке луж, так что он может быть отрицательным.

Вовочка хочет найти оптимальный путь от своего дома до дома каждого из своих друзей. Пусть считается оптимальным, если запасы жидкости у Слоняши в конце пути максимально возможные, а значит суммарный вес всех дорожек в пути минимально возможен. Вовочка надеется, что если умело расходовать драгоценные миллилитры, то можно будет потом сходить куда-нибудь ещё!

Помогите Вовочке решить его нечеловеческую проблему.

Формат входных данных

В первой строке входного файла записано три целых числа: N — количество вершин в графе, M — количество дорожек, K — количество друзей у Вовочки ($2 \leqslant N \leqslant 5\,000$, $1 \leqslant M \leqslant 50\,000$, $1 \leqslant K \leqslant 50$).

Во второй строке записано K различных целых чисел: номера вершин v_j, в которых живут друзья Вовочки ($2 \leqslant v_j \leqslant N$). Вовочка живёт в вершине номер один.

Далее в M строках описываются дорожки, по три целых числа в каждой строке: a_i — вершина, в которой начинается дорожка, b_i — вершина, в которой заканчивается дорожка и w_i — вес дорожки в миллилитрах ($1 \leqslant a_i, b_i \leqslant N$, $|w_i| \leqslant 10^5$).

Гарантируется, что в графе нет циклов отрицательного веса. Видимо, идея с выгулом

собаки Вовочке не первому пришла в голову, и все подобные циклы уже тщательно выпиты.

Формат выходных данных

Для каждого из K друзей Вовочки нужно вывести оптимальный путь до его дома, в порядке задания этих домов во входных данных. Каждый путь описывается в отдельной строке. Сначала должен быть записан вес пути, который должен быть минимально возможным, затем количество вершин в пути, и наконец номера всех этих вершин в порядке прохождения по пути.

Гарантируется, что до каждого друга можно добраться по дорожкам.

Пример

input.txt	output.txt
5 9 2 2 4 1 3 1000 3 2 300 1 2 1200 1 4 500 4 5 400 5 4 0 4 3 600 3 5 -300 5 2 200	900 4 1 3 5 2 500 2 1 4

Комментарий

Учтите, что автор условия **не** разделяет легкомысленное отношение Вовочки к карантину и к жизни вообще.

Задача 8. Сплетницы

Источник:	основная II
Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	3 секунды*
Ограничение по памяти:	256 мегабайт

Домохозяйки проводят много времени за телефонными разговорами с подругами. Как только одной из них становится что-нибудь известно, она садится за телефон и сообщает свежую сплетню подругам. С каждой из своих подруг домохозяйка заканчивает разговор через разное время. Напишите программу, вычисляющую минимальное время, за которое сплетня дойдет от домохозяйки-распространительницы свежей сплетни, живущей в дома S, до другой, живущей в доме T.

Формат входных данных

В первой строке файла содержатся целые числа N, M, K — количество домохозяек, количество пар любящих поговорить домохозяек, и количество запросов ($1 \leqslant N \leqslant 3\,000$, $1 \leqslant M \leqslant 300\,000$, $1 \leqslant K \leqslant 20$).

Далее идут K строк с запросами. В каждой строке два целых числа S_i и T_i — номер дома, где зарождается сплетня, и номер дома, в которую она попадёт ($1 \leqslant S_i \neq T_i \leqslant N$).

В каждой из следующих строк располагаются по три целых числа, из которых первые два числа — это номера домов домохозяек, которые любят разговаривать по телефону друг с другом, третье число — время разговора между ними — неотрицательно и не превосходит 100 000.

Формат выходных данных

Для каждого запроса во входных данных нужно вывести ответ на задачу в отдельной строке.

Если сплетня не может дойти до адресата, то выведите лишь слово "`NO`".

В противном случае выведите слово "`YES`", затем искомое минимальное время C и количество сплетниц P в том пути, на котором это время достигается. Наконец, выведите P целых чисел — номера домов на найденном пути в том порядке, в котором по ним будет проходить сплетня.

Пример

input.txt	output.txt
7 6 4 1 7 7 1 5 3 4 6 1 2 2 1 3 1 3 6 1 3 7 2 6 7 1 2 4 1	YES 3 3 1 3 7 YES 3 4 7 6 3 1 NO YES 5 5 4 2 1 3 6

Задача 9. КСС

Источник:	основная I
Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Вам задан ориентированный граф с N вершинами и M ребрами. Возможно, в граф закрались петли и кратные ребра.

Определите компоненты сильной связности заданного графа.

Формат входных данных

Граф задан во входном файле следующим образом: первая строка содержит числа N и M ($1 \leqslant N, M \leqslant 100\,000$). Каждая из следующих M строк содержит описание ребра — два целых числа из диапазона от 1 до N — начало и конец ребра соответственно.

Формат выходных данных

На первой строке выходного файла выведите число L — количество компонент сильной связности заданного графа. На следующей строке выведите N чисел из диапазона от 1 до L — номера компонент сильной связности, которым принадлежат соответствующие вершины. Компоненты сильной связности следует занумеровать от 1 до L произвольным образом.

Пример

input.txt	output.txt
4 4 2 1 3 2 2 3 4 3	3 1 2 2 3

Задача 10. Мосты

Источник:	основная I
Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Дан неориентированный граф. Необходимо найти все ребра, удаление которых приводит к увеличению количества компонент связности.

Формат входных данных

В первой строке содержатся два числа N и M ($1 \leqslant N, M \leqslant 100\,000$), где N — количество вершин графа, M — количество ребер. В следующих M строках содержится по два числа: u, v ($1 \leqslant u, v \leqslant N$), означающих, что в графе есть ребро между вершинами u и v.

Возможны кратные рёбра и петли.

Формат выходных данных

В первой строке выведите количество найденных рёбер. Во второй строке выведите номера рёбер в порядке возрастания. Рёбра нумеруются в том порядке, в каком они заданы во входном файле.

Пример

input.txt	output.txt
6 5 1 2 3 2 4 5 1 3 4 6	2 3 5

Задача 11. Точки сочленения

Источник:	повышенной сложности I
Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Дан неориентированный граф. Необходимо найти все вершины, удаление которых приводит к увеличению количества компонент связности.

Формат входных данных

В первой строке содержатся два числа N и M ($1 \leqslant N, M \leqslant 100\,000$), где N — количество вершин графа, M — количество ребер. В следующих M строках содержится по два числа: u, v ($1 \leqslant u, v \leqslant N$), означающих, что в графе есть ребро между вершинами u и v.

Возможны кратные рёбра и петли.

Формат выходных данных

В первой строке выведите количество найденных вершин. Во второй строке выведите номера вершин в порядке возрастания.

Пример

input.txt	output.txt
9 12 1 2 2 3 4 5 2 6 2 7 8 9 1 3 1 4 1 5 6 7 3 8 3 9	3 1 2 3

Задача 12. Дорожная реформа

Источник:	повышенной сложности II
Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

В Берляндии произошла реформа дорожного движения, направленная на устранение пробок. Планируется достичь этого за счёт того, чтобы сделать поток машин более равномерно распределённым по дорогам страны. Для популярных дорог установили плату за проезд, надеясь, что это отпугнёт автомобилистов, а на непопулярных дорогах стали выдавать проезжающим подарки, купоны со скидками, билеты в кино, и так далее.

Опытный авантюрист по имени Остап хочет нажиться за счёт реформы. Он уже посчитал для каждой дороги общие затраты на проезд, включая цену на бензин и выручку с подарков. Для некоторых непопулярных дорог затраты получились отрицательными, то есть стоимость подарков перекрывает все платы. Теперь он хочет узнать для каждого города в стране, сколько нужно минимально потратить денег, чтобы до него добраться. Или если посмотреть на это с другой стороны, сколько максимально денег можно заработать на пути до него.

Формат входных данных

Первая строка входного файла содержит три целых числа: N — количество городов, M — количество дорог и s — номер города, в котором живёт Остап ($2 \leqslant N \leqslant 2\,000$, $1 \leqslant M \leqslant 6\,000$). Города пронумерованы числами от 1 до N.

Следующие M строк содержат описание дорог. Каждая дорога задаётся номером города, их которого она выходит, номером города, в который она приходит, и затратами на проезд. Затраты на проезд — целое число, не превосходящее 10^{15} **по модулю**. Все дороги в Берляндии с односторонним движением.

Формат выходных данных

Для каждого города u выведите минимальные затраты, с которыми можно до него добраться из s, в отдельной строке. Если в город u доехать нельзя, то выведите '`*`' вместо длины. Если Остап может по пути заработать неограниченно много денег, то выведите '`-`' вместо длины.

Пример

input.txt	output.txt
7 8 1 1 2 10 2 3 5 1 3 100 3 5 7 5 4 10 4 3 -18 6 1 -1 2 7 -50	0 10 - - - * -40

Задача 13. Авиаперелёты

Источник:	повышенной сложности II (**обязательна на «отлично»**)
Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	3 секунды*
Ограничение по памяти:	256 мегабайт

В результате пандемии коронавируса закрываются границы и отменяются рейсы. К сожалению, не все люди успели вернуться домой вовремя, и оставшимся за границей теперь приходится нелегко.

Группа туристов находится в аэропорту A и отчаянно хочет вернуться домой в аэропорт B. Поскольку некоторые регулярные рейсы были отменены, теперь они не могут вернуться домой за ту цену, на которую они рассчитывали. Они согласны лететь через сколько угодно стран и аэропортов, лишь бы оказаться дома, потратив на это минимальную сумму денег. В довершение всего, ни у кого из них нет интернета под рукой, чтобы купить билеты на сайте, зато есть прайслист всех авиарейсов на сегодня.

Помогите туристам составить оптимальный маршрут для возвращения домой!

Формат входных данных

В первой строке файла содержатся целые числа N, M, K — количество аэропортов, количество авиарейсов, и количество запросов ($1 \leqslant N \leqslant 100\,000$, $1 \leqslant M \leqslant 200\,000$, $1 \leqslant K \leqslant 20$).

Далее идут K строк с запросами. В каждой строке два целых числа A_i и B_i — номер аэропорта, в котором находится группа туристов, и номер аэропорта, в который она хочет попасть ($1 \leqslant A_i \neq B_i \leqslant N$).

В остальных M строках описываются доступные авиарейсы. В каждой строке три целых числа: номер аэропорта отправления, номер аэропорта прибытия и стоимость перелёта соответственно. Стоимость перелёта неотрицательна и не превышает 300 000. Учтите, что все авиарейсы летят только в одну сторону. В эти безумные времена авиарейс может даже прилетать в тот аэропорт, из которого вылетел.

Все аэропорты пронумерованы числами от 1 до N, а все авиарейсы — числами от 1 до M в порядке описания.

Формат выходных данных

Выведите ответ на каждый запрос в отдельной строке.

Если группа не может долететь домой, то выведите лишь слово “`DOOMED`”.

В противном случае выведите слово “`quarantine`”, затем искомую минимальную сумму денег T и количество авиарейсов F в оптимальном маршруте. Наконец, выведите F целых чисел — номера авиарейсов, на которые надо купить билеты, в порядке полёта по ним.

Пример

input.txt	output.txt
7 7 3 1 7 7 1 3 4 1 2 2 1 3 1 3 6 1 3 7 2 6 7 1 2 4 1 7 2 3	quarantine 3 2 2 4 DOOMED quarantine 6 3 4 7 6

Задача 14. Vim

Источник:	космической сложности II
Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	5 секунд
Ограничение по памяти:	256 мегабайт

Описанные в данной задаче команды могут отличаться от команд настоящего редактора Vim. Пожалуйста, внимательно читайте условие.

В редакторе Vim текст отображается построчно, при этом строки текста могут быть разной длины. Текст состоит из строчных букв латинского алфавита и пробелов.

Для навигации по тексту используется курсор. В редакторе Vim курсор находится **не** между символами, а на символе. Его положение в тексте задаётся парой координат: номером строки и позицией в строке. Положение курсора всегда соответствует реальному символу в тексте: номер строки не может быть меньше 1 или больше количества строк в тексте, а позиция не может быть меньше 1 или больше длины текущей строки.

Положение в тексте A считается расположенным раньше положения B, если:

- номер строки A меньше номера строки B;
- оба положения имеют одинаковый номер строки, но позиция A меньше позиции B.

Соответственно положение A считается расположенным позже положения B, если B расположено раньше A.

Словом в тексте называется последовательность букв латинского алфавита. Слова разделяются одним или несколькими пробелами и переводами строки. Началом слова называется первый символ слова, а концом — последний символ слова.

Vim имеет несколько различных режимов работы, один из которых — режим команд. В режиме команд можно использовать следующие клавиши для навигации по тексту:

- Клавиша '`l`' сдвигает курсор на одну позицию вправо. Если курсор стоял на последнем символе строки, то он никуда не перемещается.
- Клавиша '`h`' сдвигает курсор на одну позицию влево. Если курсор стоял на первом символе строки, то он никуда не перемещается.
- Клавиша '`w`' перемещает курсор в начало следующего слова в тексте: наименьшее положение, соответствующее началу какого-либо слова, и которое идёт позже текущего положения курсора. Если такого начала слова нет, то курсор никуда не перемещается.
- Клавиша '`b`' перемещает курсор в начало текущего слова: наибольшее положение, соответствующее началу какого-либо слова и которое идёт раньше текущего положения курсора. Если такого начала слова нет, то курсор никуда не перемещается.
- Клавиша '`$`' перемещает курсор на последнюю позицию текущей строки.
- Клавиша '`0`' (символ нуля) перемещает курсор на первую позицию текущей строки.
- Клавиша '`j`' перемещает курсор на строчку вниз. Если строка последняя, то курсор не перемещается.
- Клавиша '`k`' перемещает курсор на строчку вверх. Если строка первая, то курсор не перемещается.

При перемещении вверх и вниз курсор запоминает, с какой позиции было начато перемещение между строками — назовём эту позицию стартовой. Если курсор перемещается на строку, длина которой больше или равна номеру стартовой позиции, то в новой строке положение курсора будет иметь стартовую позицию. Если курсор перемещается на строку, длина которой меньше стартовой позиции, то курсор перемещается на последнюю позицию новой строки. При дальнейшем нажатии на клавиши '`k`' или '`j`' курсор так же выбирает позицию

в новой строке исходя **не** из текущей позиции, а из стартовой позиции, с которой началось перемещение между строками. Память о стартовой позиции стирается после нажатия на клавишу, отличную от ‘`k`’ или ‘`j`’.

Рассмотрим работу клавиши ‘`k`’ на примере. Пусть есть три строки длиной 16, 6 и 11, и курсор находится в третьей строке в позиции 9. Если первый раз нажать на «вверх», то курсор переместится из положения (3, 9) в положение (2, 6). Если второй раз нажать на «вверх», то курсор переместится из положения (2, 6) в положение (1, 9), т.к. движение между строками началось с позиции 9. Если же между первым и вторым нажатием сделать перемещение курсора влево, то перещения курсора будут: $(3, 9) \to (2, 6) \to (2, 5) \to (1, 5)$.

Вы — эффективный программист, потому в качестве основного редактора выбрали Vim. Поскольку вы эффективны, то хотите все действия выполнять за минимальный объём приложенных усилий. Перед вами, как перед программистом, часто стоит задача переместить курсор из одного положения в другое и, конечно, вы хотите это сделать за минимальное количество нажатий на клавиши. Помогите себе: напишите программу, которая по заданному тексту и заданным начальному и конечному положениям курсора найдёт кратчайшую последовательность нажатий на клавиши, перемещающую курсор из начального положения в конечное.

Формат входных данных

В первой строке содержится единственное целое число N ($N \geqslant 1$) — количество строк в тексте.

Следующие N строк содержат текст. Для удобства все символы пробела в тексте заменены на символ точки (ASCII 46). Текст состоит из строчных букв латинского алфавита и пробелов (заменённых на точки). Каждая строка текста содержит как минимум один символ, однако в ней могут отсутствовать буквы. Длина текста (суммарная длина всех строк) не превосходит 10^5.

Следующая строка содержит число Q ($1 \leqslant Q \leqslant 10^3$) — количество запросов на перемещение курсора.

Каждый запрос описан на отдельной строке. Запрос состоит из 4 чисел, описывающих начальное и конечное положение курсора: row_s, col_s, row_f и col_f — соответственно номер строки и позиция в этой строке начального положения курсора и номер строки и позиция в этой строке конечного положения курсора. Нумерация строк и позиций начинается с 1. Гарантируется, что данные координаты корректны и соответствуют реальным позициям в тексте. Начальное и конечное положения курсора не совпадают.

Произведение длины текста на количество запросов не превышает 10^7.

Формат выходных данных

Ответом для каждого запроса является кратчайшая последовательность нажатий клавиш, перемещающая курсор из начального положения в конечное. Для каждого запроса выведите ответ на него на отдельной строке. Если существует несколько кратчайших последовательностей, выведите любую из них.

Пример

input.txt	output.txt
3 abc.def q qewrewqr 4 3 7 3 5 1 2 1 5 1 2 1 6 1 1 3 5	hh w $h wjj