

Задача 14. Vim

Источник:	космической сложности II
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
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 'к' или 'j'.

Рассмотрим работу клавиши 'к' 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 перемещения курсора будут: $(3, 9) \rightarrow (2, 6) \rightarrow (2, 5) \rightarrow (1, 5)$.

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 ($N \geq 1$) — 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 ($1 \leq Q \leq 10^3$) — 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