

Задача А. К-я строка

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

Реализуйте структуру данных, которая поддерживает следующие операции:

- добавить в словарь строку S ;
- найти в словаре k -ю строку в лексикографическом порядке.

Известно, что изначально словарь пуст.

Формат входных данных

Первая строка входного файла содержит натуральное число N — количество команд ($N \leq 10^5$). Последующие N строк содержат по одной команде каждая.

Команда записывается либо в виде числа k , либо в виде строки S , которая может состоять только из строчных латинских букв. Гарантируется, что при запросе k -й строки она существует. Также гарантируется, что сумма длин всех добавляемых строк не превышает 10^5 .

Формат выходных данных

Для каждого числового запроса k выходной файл должен содержать k -ю в лексикографическом порядке строчку из словаря на момент запроса. Гарантируется, что суммарная длина строк в выходном файле не превышает 10^5 .

Примеры

<code>kthstr.in</code>	<code>kthstr.out</code>
7	tolstoy
pushkin	gogol
lermontov	
tolstoy	
gogol	
gorkiy	
5	
1	

Задача В. Type Printer

Имя входного файла:	<code>printer.in</code>
Имя выходного файла:	<code>printer.out</code>
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Вам нужно напечатать N слов на *Movable Type Printer*. *Movable Type Printers* — это старые принтеры, для работы которых требуется ставить маленькие металлические кусочки (каждый из кусочков содержит одну букву) в определенном порядке, образуя таким образом слова. Потом все они вдавливаются в лист бумаги. Таким образом печатается одно слово. Ваш принтер позволяет делать следующие операции:

- Добавить букву в конец слова, находящегося сейчас на принтере.
- Удалить последнюю букву из слова, находящегося сейчас на принтере. Эту операцию можно делать, только если слово содержит хотя бы одну букву.
- Напечатать слово, находящееся на принтере (при этом слово никуда не исчезает, можно печатать его ещё раз и ещё раз).

Изначально на принтере содержится пустое слово. В конце печати на принтере можно оставить непустое слово. Слова, которые вам даны, вы можете печатать в произвольном порядке.

Каждая из трёх операций занимает одну единицу времени. Вам нужно найти последовательность операций, которая печатает данные N слов за минимальное время. Если минимальных последовательностей несколько, выведите любую.

Формат входных данных

В первой строке содержится число N ($1 \leq N \leq 25\,000$). В следующих N строках содержатся слова (по одному на строке), состоящие из маленьких букв латинского алфавита. Длина каждого слова не превышает 20. Все слова различны.

Формат выходных данных

Ваша программа должна вывести следующие данные:

- На первой строке число M — число операций.
- На следующих M строках по одному символу — описание операций. Каждая операция описывается одним символом:
 - Добавление символа обозначается собственно символом.
 - Удаление символа обозначается символом «-» (минус, ASCII-код 45).
 - Операция «напечатать текущее слово» обозначается символом «P» (заглавная латинская буква P).

Примеры

printer.in	printer.out
3 print the poem	20 t h e P - - - P o e m P - - - r i n t P

Задача С. Жужжащий профессор

Имя входного файла: `prof.in`
Имя выходного файла: `prof.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

В одном очень известном университете один очень известный профессор очень быстро произносил свои лекции — так, что ничего невозможно было разобрать. Студенты шутили по этому поводу, что он не говорит, а жужжит. Естественно, что про загадочного профессора никто абсолютно ничего не знал.

Но вот недавно Петя Булочкин решил предпринять исследование по изучению словарного запаса профессора. С этой целью он даже посетил одну лекцию и записал все сказанное на ней на диктофон. Затем, прокручивая дома запись с десятикратным замедлением, Петя смог записать все, что сказал профессор. Но вот незадача — профессор говорил так быстро, что, даже прослушивая замедленную запись, нельзя было точно сказать, где он делал паузы между словами. Таким образом, у Пети есть некоторый текст S , состоящий только из маленьких латинских букв — лекция, которая была прочитана профессором.

Петя решил, что те слова, которые профессор употреблял только один раз во время своей лекции, его не интересуют. Кроме того, понятно, что если профессор употреблял некоторое слово два или более раз, то существуют два неперекрывающихся вхождения этого слова в текст S . Назовем непустую строку T кандидатом в слова, если существуют два неперекрывающихся вхождения T в S . Теперь Петя хочет найти все строки, которые являются кандидатами в слова. И поможете ему в этом вы.

Формат входных данных

Единственная строка входного файла содержит от 1 до 3000 маленьких латинских букв. Это и есть текст S , который прочитал профессор на лекции.

Формат выходных данных

Единственная строка выходного файла должна содержать одно число, равное количеству строк, являющихся кандидатами в слова.

Примеры

<code>prof.in</code>	<code>prof.out</code>
<code>bbaabbbabb</code>	<code>7</code>

Задача D. Поиск набора образцов

Имя входного файла: `console2.in`
Имя выходного файла: `console2.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Напишите программу, которая для каждой строки из заданного набора S проверяет, верно ли, что она содержит как подстроку одну из строк из набора T .

Формат входных данных

Первая строка входного файла содержит натуральное число n ($1 \leq n \leq 1000$) — количество строк в наборе T . Каждая из следующих n строк содержит непустую строку. Гарантируется, что суммарная длина всех строк из набора T не превышает 80 000. Оставшаяся часть файла содержит строки из набора S . Каждая строка состоит из ASCII символов с кодами от 32 до 126 включительно. Строка может быть пустой. Гарантируется, что размер входного файла не превышает 1 МБ.

Формат выходных данных

В выходной файл выведите все строки из набора S (в том порядке, в котором они находятся во входном файле), содержащие как подстроку по крайней мере одну строку из набора T .

Примеры

<code>console2.in</code>	<code>console2.out</code>
<code>3</code>	<code>sudislavl</code>
<code>gr</code>	<code>group b</code>
<code>sud</code>	
<code>abc</code>	
<code>lksh</code>	
<code>sudislavl</code>	
<code>kostroma</code>	
<code>summer</code>	
<code>group b</code>	

Задача Е. Мобильные телефоны

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

В области Тампере существуют станции обслуживания мобильных телефонов четвертого поколения, которые работают следующим образом. Вся область разделена на квадраты. Квадраты формируют матрицу $S \times S$, строки и столбы которой нумеруются с 0 до $S - 1$. Каждый квадрат содержит одну станцию обслуживания. Количество активных мобильных телефонов внутри квадрата может меняться, потому что телефон может перемещаться из одного квадрата в другой или выключаться. Временами каждая станция обслуживания передает сведения об изменении в количестве активных телефонов на главную станцию обслуживания (вместе с номером строки и столбца матрицы).

Напишите программу, которая принимает эти сведения и отвечает на запросы о текущем количестве активных мобильных телефонов в любой прямоугольной области.

Формат входных данных

Каждый запрос к программе задан на отдельной строке и состоит из одного целого числа, описывающего команду, и последовательности чисел, являющиеся параметрами команды.

Команда	Параметры	Значение
0	S	Инициализирует матрицу размером $S \times S$, состоящую только из нулей. Эта команда поступает один раз и является самой первой командой.
1	$X Y A$	Добавить число A к количеству активных мобильных телефонов в квадрате $(X; Y)$. Число A может быть как положительным, так и отрицательным.
2	$L B R T$	Выдать текущую сумму количеств активных мобильных телефонов во всех квадратах $(X; Y)$, таких что $L \leq X \leq R, B \leq Y \leq T$.
3	\emptyset	Означает конец входных данных. Эта команда поступает ровно один раз и является самой последней командой.

S не превосходит 1024, количество команд — 60 002, значение ячейки в любой момент времени — 2^{15} , количество активных телефонов во всей таблице — 2^{30} . Величина обновления не превосходит по модулю 2^{15} , также ни в один момент времени число активных телефонов в квадрате не становится меньше нуля.

Все индексы нумеруются начиная с нуля.

Формат выходных данных

На запросы типов 0, 1, 3 отвечать не надо. Для каждого запроса второго типа выведите строку, содержащую одно целое число — ответ на него.

Примеры

<code>mobile-phone.in</code>	<code>mobile-phone.out</code>
0 4	3
1 1 2 3	4
2 0 0 2 2	
1 1 1 2	
1 1 2 -1	
2 1 1 2 3	
3	