

Задача А. Поиск подстроки

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Найти позиции всех вхождений строки T в строку S .

Формат входных данных

Первые две строки входных данных содержат строки S и T , соответственно. Длины строк больше 0 и меньше 50000, строки содержат только латинские буквы.

Формат выходных данных

Выведите номера символов, начиная с которых строка T входит в строку S , в порядке возрастания.

Примеры

стандартный ввод	стандартный вывод
ababbababa aba	0 5 7

Задача В. Равные подстроки

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дана строка $S = s_1 s_2 \dots s_n$ и множество запросов вида (l_1, r_1, l_2, r_2) . Для каждого запроса требуется ответить, равны ли подстроки $s_{l_1} \dots s_{r_1}$ и $s_{l_2} \dots s_{r_2}$.

Формат входных данных

В первой строке записана строка S , состоящая из строчных латинских букв. Эта строка непустая и имеет длину не более 100 000 символов. Во второй строке записано целое число q ($1 \leq q \leq 100\,000$) — количество запросов. В каждой из следующих q строк записаны числа l_1, r_1, l_2, r_2 ($1 \leq l_1 \leq r_1 \leq |S|$; $1 \leq l_2 \leq r_2 \leq |S|$).

Формат выходных данных

Для каждого запроса выведите «+», если соответствующие подстроки равны, и «-», в противном случае.

Примеры

стандартный ввод	стандартный вывод
abacaba 4 1 1 7 7 1 3 5 7 3 4 4 5 1 7 1 7	++-+
qa 3 1 1 1 1 2 2 2 2 1 1 2 2	++-

Задача С. Сравнение подстрок

Имя входного файла: `substrcmp.in`
Имя выходного файла: `substrcmp.out`
Ограничение по времени: 1.5 секунд
Ограничение по памяти: 256 мегабайт

Дана строка $S = s_1s_2 \dots s_n$ и множество запросов вида (l_1, r_1, l_2, r_2) . Для каждого запроса требуется ответить, какая из подстрок больше — $s_{l_1} \dots s_{r_1}$ или $s_{l_2} \dots s_{r_2}$.

Формат входных данных

В первой строке записана строка S , состоящая из строчных латинских букв. Эта строка непустая и имеет длину не более 100 000 символов. Во второй строке записано целое число q ($1 \leq q \leq 100\,000$) — количество запросов. В каждой из следующих q строк записаны числа l_1, r_1, l_2, r_2 ($1 \leq l_1 \leq r_1 \leq |S|$; $1 \leq l_2 \leq r_2 \leq |S|$).

Формат выходных данных

Для каждого запроса выведите «=», если соответствующие подстроки равны, «>», если первая подстрока больше и «<», если первая подстрока меньше.

Примеры

substrcmp.in	substrcmp.out
abacaba	=
3	<
1 3 5 7	>
1 3 3 5	
4 7 2 5	
ab	<
2	<
1 1 2 2	
1 1 1 2	

Задача D. Наибольший подпалиндром

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Дана строка длины N . Требуется найти наибольшую подстроку, являющуюся палиндромом.

Формат входных данных

В первой и единственной строке входного файла дана последовательность заглавных и строчных букв английского алфавита. Длина последовательности $1 \leq N \leq 1000$.

Формат выходных данных

В первой строке выведите длину наибольшего подпалиндрома. Во второй строке выведите подпалиндром. Если наибольших подпалиндромов несколько, выведите первый из них.

Примеры

стандартный ввод	стандартный вывод
abaca	3 aba

Задача Е. Максимальная общая подстрока

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Дана строка $A = a_1a_2 \dots a_n$ и строка $B = b_1b_2 \dots b_m$. Нужно узнать длину их максимальной общей подстроки.

Формат входных данных

В первых двух строках записаны строки A и B , состоящие из строчных латинских букв. Эти строки непустые и имеют длину не более 30 000 символов.

Формат выходных данных

В выходной файл выведите длину их максимальной общей подстроки.

Примеры

стандартный ввод	стандартный вывод
abacaba acabaca	5

Задача F. Движущаяся строка

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Поликарп нашёл строку s и перестановку p . Их длины оказались одинаковы и равны n .

Перестановка из n элементов — это массив длины n , в котором каждое целое число от 1 до n встречается ровно по одному разу. Например, $[1, 2, 3]$ и $[4, 3, 5, 1, 2]$ — это перестановки, но $[1, 2, 4]$, $[4, 3, 2, 1, 2]$ и $[0, 1, 2]$ — это не перестановки.

За одну операцию он может умножить s на p , то есть заменить строку s на строку new , в которой для каждого i от 1 до n верно, что $new_i = s_{p_i}$. Например, при $s = wmbc$ и $p = [3, 1, 4, 2]$, после применения операции строка превратится в $s = s_3s_1s_4s_2 = bwem$.

Поликарпу стало интересно, через сколько операций строка впервые вернётся к своему первоначальному виду. Так как это может занять слишком много времени, он просит вашей помощи в этом вопросе.

Можно доказать, что искомое количество операций всегда существует. Оно может оказаться очень большим, используйте 64-битный целочисленный тип.

Формат входных данных

В первой строке входных данных записано целое число t ($1 \leq t \leq 5000$) — количество наборов входных данных в тесте.

Первая строка каждого набора содержит целое число n ($1 \leq n \leq 200$) — длину строки и перестановки.

Вторая строка каждого набора содержит строку s длины n , состоящую из строчных латинских букв.

Третья строка каждого набора содержит n целых чисел — перестановку p ($1 \leq p_i \leq n$), все p_i различны.

Формат выходных данных

Выведите t строк, каждая из которых содержит ответ на соответствующий набор входных данных. В качестве ответа выведите единственное число — минимальное количество операций, после которого строка s станет такой же, какой была до их применения.

Пример

стандартный ввод	стандартный вывод
3	1
5	6
ababa	12
3 4 5 2 1	
5	
ababa	
2 1 4 5 3	
10	
codeforces	
8 6 1 7 5 2 9 3 10 4	

Замечание

В первом наборе входных данных применение операции не изменяет строку, поэтому она станет равной самой себе после 1 операции.

Во втором наборе входных данных строка будет меняться следующим образом:

- $s = babaa$
- $s = abaab$

- $s = \text{baaba}$
- $s = \text{abbaa}$
- $s = \text{baaab}$
- $s = \text{ababa}$

Задача G. Дерево и строки

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Есть дерево состоящее всего из одной вершины с номером 1. К дереву постепенно добавляются новые вершины. В i -ю секунду в дерево добавляется вершина с номером $i+1$, которая подвешивается в качестве сына к вершине p_i , а на ребре между вершинами $i+1$ и p_i записывается буква c_i .

Каждому пути из корня дерева до вершины v соответствует некоторая строка, получающаяся путем выписывания символов, записанных на ребрах текущего пути в порядке следования от корня к вершине v .

После каждого добавления новой вершины посчитайте количество уникальных строк, начинающихся в корне дерева (вершине с номером 1), и заканчивающихся в какой-либо другой вершине.

Формат входных данных

В первой строке записано число n ($1 \leq n \leq 3 \cdot 10^5$) — количество запросов на добавление новой вершины в дерево.

В следующих n строках описаны запросы добавления вершин. i -й запрос описывается параметрами p_i ($1 \leq p_i \leq i$) и c_i (строчная буква латинского алфавита), которые означают, что добавленная вершина с номером $i+1$ подвешивается к вершине с номером p_i в качестве потомка, а на полученном ребре записывается символ c_i .

Формат выходных данных

Выведите n строк. В i -й строке выведите количество уникальных строк после добавления $(i+1)$ -й вершины.

Примеры

стандартный ввод	стандартный вывод
2 1 b 2 p	1 2
3 1 o 1 o 2 j	1 1 2

Задача Н. Автодополнение

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

На занятиях по «Алгоритмам и структурам данных» Сергей узнал как эффективно обрабатывать строковые данные. Теперь он хочет написать собственную функцию, которая заменит ему встроенный T9 на его собственный интеллектуальный помощник.

Сергей хочет написать программу, которая эффективно и правильно обрабатывает два вида запросов:

- добавить слово S в словарь;
- вывести число слов в словаре, начинающихся на префикс S .

Сергей уверен, что самостоятельно напишет такую программу, которая будет обрабатывать такие запросы не дольше 2 сек. А вы сможете написать такую программу?

Формат входных данных

Первая строка содержит целое число n ($1 \leq n \leq 10\,000$) — количество слов в словаре телефона.

Следующие n строк описывают слова из словаря. Каждая из них содержит последовательность, состоящую из строчных латинских букв.

Следующая строка содержит целое число m ($1 \leq m \leq 10\,000$) — количество запросов.

Следующие m строк описывают запросы. Каждый запрос состоит из числа, означающего тип запроса и строки:

- 1 — запрос на добавление слово в словарь;
- 2 — запрос на подсчёт количество слов, начинающихся с данного префикса;

Все слова словаря и запросов состоят из строчных латинских букв из не превышают по длине 500 символов. Все добавляемые в словарь буквенные последовательности различны. Суммарная длина буквенных последовательностей во входных данных не превосходит $8 \cdot 10^6$ символов. Гарантируется, что есть хотя бы один запрос вида 2.

Формат выходных данных

Выведите одну или более строк, каждая из которых содержит одно целое число — для каждого запроса типа 2 количество слов из словаря, начинающихся на заданный префикс.

Примеры

стандартный ввод	стандартный вывод
2 codeforces contest 4 2 a 2 code 1 cf 2 c	0 1 3
3 programming week weekends 3 2 week 1 program 2 prog	2 2

Задача I. 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 pushkin lermontov tolstoy gogol gorkiy 5 1	tolstoy gogol

Задача J. Цепочка слов

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.25 секунд
Ограничение по памяти: 256 мегабайт

Цепочкой слов длины n назовем последовательность слов $w_1, w_2, \dots, w_n$ такую, что для $1 \leq i < n$ слово w_i является собственным префиксом слова w_{i+1} .

Напомним, что слово u длины k называется собственным префиксом слова v длины l , если $l > k$ и первые k букв слова v совпадают со словом u .

Задано множество слов $S = \{s_1, s_2, \dots, s_m\}$. Найдите максимальную длину цепочки слов, которую можно построить, используя (возможно, не все) слова этого множества.

Формат входных данных

Первая строка ввода содержит целое число m ($1 \leq m \leq 255$). Каждая из последующих m строк содержит по одному слову из множества S .

Все слова не пусты, имеют длину, не превосходящую 255 символов, и состоят только из строчных букв латинского алфавита.

Формат выходных данных

Выведите ответ на задачу.

Примеры

стандартный ввод	стандартный вывод
3 a ab abc	3
5 a ab bc bcd add	2