

Задача A. Вася и Циклические Сдвиги

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

Однажды Василий проголодался и пошел на кухню подкрепиться. Но каково было его удивление, когда, открыв холодильник, он обнаружил там не привычные продукты, а строку! Причем не просто строку, а заикленную, прямо как бублик (ассоциации с едой часто приходят Васе в голову, когда он голоден). Загадочная строка заинтересовала любопытного, но все еще голодного Василия, и он начал ее вертеть в руках. Например, если вертеть строку `abacaba`, то можно получить следующие строки:

- `abacaba`
- `bacabaa`
- `acabaab`
- `cabaaba`
- `abaabac`
- `baabaca`
- `aabacab`

И тут Василия осенило: если он сможет посчитать, сколько раз в процессе кручения строки получается лексикографически минимальная строка, то еда магическим образом появится в холодильнике (странные идеи часто приходят Васе в голову, когда он голоден). Помогите Васе, иначе он так и будет сидеть голодным.

Более формально, вам дана строка. *Циклическим сдвигом* строки s длины n называется строка, полученная из исходной путем отбрасывания первых $0 \leq k < n$ символов и приписывания их в конец. Необходимо посчитать, сколько раз среди всех циклических сдвигов строки встречается лексикографически минимальный циклический сдвиг.

Формат входных данных

Единственная строка входного файла содержит строку S , найденную Василием. Она непуста, состоит из маленьких латинских букв, и её длина не превосходит 10 000 000.

Формат выходных данных

Выведите единственное число — искомое количество минимальных циклических сдвигов.

Примеры

стандартный ввод	стандартный вывод
<code>aaaa</code>	4
<code>abacaba</code>	1

Замечание

В первом примере минимальным циклическим сдвигом является строка `aaaa`. Во втором примере минимальным циклическим сдвигом является строка `aabacab`.

Задача В. Неточное совпадение

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

Даны строки p и t . Требуется найти все вхождения строки p в строку t в качестве подстроки с точностью до возможного несовпадения одного символа.

Формат входных данных

Первая строка входного файла содержит p , вторая — t ($1 \leq |p|, |t| \leq 10^6$). Строки состоят из букв латинского алфавита.

Формат выходных данных

В первой строке выведите количество вхождений строки p в строку t . Во второй строке выведите в возрастающем порядке номера символов строки t , с которых начинаются вхождения p . Символы нумеруются с единицы.

Пример

стандартный ввод	стандартный вывод
aaaa	4
Caaabdaaaa	1 2 6 7

Задача С. Шок контент

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

Вам даны две последовательности длины N : $a = \{a_0, \dots, a_{N-1}\}$ и $b = \{b_0, \dots, b_{N-1}\}$, состоящие из целых положительных чисел.

Ёж выберет целое число k , такое что $0 \leq k < N$ и целое положительное число x . После этого он создаст новую последовательность длины N $a' = \{a'_0, \dots, a'_{N-1}\}$ так, что $a'_i = a_{(i+k) \bmod N} \oplus x$.

Найдите все пары (k, x) такие, что a' будет равняться b .

Формат входных данных

Первая строка входных данных содержит целое число N ($1 \leq N \leq 2 \cdot 10^5$) — количество элементов в последовательностях.

Вторая строка содержит N целых чисел a_i ($0 \leq a_i \leq 2^{30}$) — элементы последовательности a .

Третья строка содержит N целых чисел b_i ($0 \leq b_i \leq 2^{30}$) — элементы последовательности b .

Формат выходных данных

Выведите все пары (k, x) , что a' будет равняться b .

На каждой строке вывода должна содержаться ровно 1 пара.

Пары должны идти в порядке возрастания k (в случае равенства k в порядке возрастания x)

Если подходящих пар не существует, то вывод должен быть пустым.

Примеры

стандартный ввод	стандартный вывод
3 0 2 1 1 2 3	1 3
5 0 0 0 0 0 2 2 2 2 2	0 2 1 2 2 2 3 2 4 2
6 0 1 3 7 6 4 1 5 4 6 2 3	2 2 5 5
2 1 2 0 0	

Задача D. Лотерея

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

На одном из телеканалов каждую неделю проводится следующая лотерея. В течение недели участники делают свои ставки. Каждая ставка заключается в назывании какого-либо M -значного числа в системе счисления с основанием K (то есть, по сути, каждый участник называет M цифр, каждая из которых лежит в диапазоне от 0 до $K - 1$). Ведущие нули в числах допускаются.

В некоторый момент прием ставок на текущий розыгрыш завершается, и после этого ведущий в телеэфире называет выигравшее число (это также M -значное число в K -ичной системе счисления). После этого те телезрители, у кого первая цифра их числа совпала с первой цифрой числа, названного ведущим, получают выигрыш в размере A_1 рублей. Те, у кого совпали первые две цифры числа — получают A_2 рублей (при этом если у игрока совпала вторая цифра, но не совпала первая, он не получает ничего). Аналогично угадавшие первые три цифры получают A_3 рублей. И так далее. Угадавшие все число полностью получают A_M рублей. При этом если игрок угадал t первых цифр, то он получает A_t рублей, но не получает призы за угадывание $t - 1, t - 2$ и т.д. цифр. Если игрок не угадал первую цифру, он не получает ничего.

Напишите программу, которая по известным ставкам, сделанным телезрителями, находит число, которое должна назвать телеведущая, чтобы фирма-организатор розыгрыша выплатила в качестве выигрышей минимальную сумму. Для вашего удобства ставки, сделанные игроками, уже упорядочены по неубыванию.

Формат входных данных

В первой строке задаются числа N (количество телезрителей, сделавших свои ставки, $1 \leq N \leq 100\,000$), M (длина чисел, $1 \leq M \leq 10$) K (основание системы счисления $2 \leq K \leq 10$). В следующей строке записаны M чисел A_1, A_2, \dots, A_M , задающих выигрыши в случае совпадения только первой, первых двух, ..., всех цифр ($1 \leq A_1 \leq A_2 \leq \dots \leq A_M \leq 100\,000$). В каждой из следующих N строк записано по одному M -значному K -ичному числу. Числа идут в порядке неубывания.

Формат выходных данных

В первой строке выведите искомое число (если решений несколько — выведите минимальное), а во второй строке — сумму, которую при назывании телеведущей первого числа придется выплатить в качестве выигрыша.

Примеры

стандартный ввод	стандартный вывод
10 3 2 1 3 100 000 000 001 010 100 100 100 100 110 111	011 6
1 1 10 100 0	1 0

Задача E. Удаление

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 128 мегабайт

Дано n различных строк s_i . Требуется найти количество таких пар (i, j) , что $i \neq j$, а так же s_j можно получить из s_i каждый раз удаляя только первый или второй символ из s_i .

Например из строки `abcxux` можно получить строку `сux`: `abcxux` \rightarrow `асxux` \rightarrow `сxux` \rightarrow `сux`.

Формат входных данных

Первая строка содержит число n ($2 \leq n \leq 200000$) — количество строк.

i -я из следующих n строк содержит строку s_i ($1 \leq |S|$). Гарантируется, что s_i состоит из строчных букв латинского алфавита.

Так же гарантируется, что $|s_1| + \dots + |s_n| \leq 10^6$.

Примеры

стандартный ввод	стандартный вывод
3 abcxux сux abc	1
6 b a abc c d ab	5

Задача F. Задачка на строчечки

Имя входного файла: inputik.txt
 Имя выходного файла: outputik.txt
 Ограничение по времени: 2 секунды
 Ограничение по памяти: 256 мегабайтов

В стандартном поточике вводика или файлик
 е
 кой. На следующей строчечке программочк
 ь а і
 н к; эти словечки мы назовём словарик п
 е о о н р
 н в utputik.txt N строчечек. В i-ой ма u
 до о . й t
 ок - количюсик (сколько штучеч с д і
 х у к е т В ё k
 с б и о ем через пробельчик для к р а т .
 и л к т) о ш t
 х й л а в всех вхожденъечек к ч а ч х
 м и а е з о а в е и t
 ё к ф с к ртированном про э ж х ч п с
 в с и , и о я т д о к р е в
 он в ч й л с рочечек нач до о же о л а
 з и о а т т и о й г д г к ш
 а т и е к ч о с ничек. н ч о е п р o a
 на л о ь а и а к с н р a
 л и в н н в х де с ястю е т в ь о м N п
 ы р е е . р х e г м , р
 м х а е н и ю св и к и с к е д н и о о ч р o o
 и к п д к у ч ж e a ч a г
 ю к и о и к ь н e н д o x c и в и k ч e д k m k p
 у ь д : x c e м a v a
 р н o k c к e d н и e т и д e в ы в a к ч и ь н c o m
 o e v o и т ч d c m
 т л ы л в a к и p a в o л c з и i и k ч e ч o p k o l o
 o a v e a л e ч
 к м с и ч o к ь л o x c e n и t c e в ы в a н ж л o d ж d k
 к н у a
 , e e ч и c o т o п й н т p a d н a т c a n и t c e в ы в a ю
 к ж щ н
 o х e t з и y k ч e в o л c o n - х a k ч e ч o p t c N х и a
 в й
 o k y б х и k c н и т a л х и k ь н e л a m з и y k ч e ч o p t c т ё d

Фoрмaт входных данных

В стандартном потoчике ввода или файлике `inputik.txt` ваша программочка найдёт стрoчечку из маленьких латинских буквoк, которую мы назовём исходненькой. На следующей стрoчечке программочка найдёт числище N ($1 \leq N \leq 1\,000\,000$), а в следующих N стрoчечках — по словечку из тех же маленьких латинских буквoк; эти словечки мы назовём словарикoм. Суммарненькая суммочка длинниц словечек из словарика не превосходит $1\,000\,000$.

Фoрмaт выходных данных

Ваша программочка должна вывести на стандартный потoчичек вывода или в файл `outputik.txt` N стрoчечек. В i -oй стрoчечке программочка должна вывести несколько чиселoк: первое чиселко — количюсик (сколько штучечек) вхожденьечек стрoчечки i из словарика в исходненькой, затем через пробельчик для каждого вхожденьичка выведите индексики началиков всех вхожденьичек этой стрoчечки в исходненькую в отсортированном порядoчке. Индексики всех стрoчечек начинаются с единичек. Няшечки-преподавашечки гарантируют, что колoчюсик вхожденьичек не превосходит $1\,000\,000$.

Пример

<code>inputik.txt</code>	<code>outputik.txt</code>
<code>abrachkacadabrachka</code>	<code>2 1 12</code>
<code>4</code>	<code>1 9</code>
<code>abrachka</code>	<code>2 1 12</code>
<code>cadabrachka</code>	<code>0</code>
<code>ab</code>	
<code>marazmik</code>	

Задача G. Подстрока

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

Недавно разведка перехватила зашифрованное сообщение — строку s . Все ресурсы аналитического центра, в котором вы работаете, были брошены на его декодирование. Ваш отдел занимается шифрами нового поколения. На данный момент известно всего n таких шифров. Для каждого из них есть три характерных параметра — целые числа l, r и строка t . Пусть строка g была получена в результате применения этого метода. Тогда строка $g_l g_{l+1} \dots g_{r-1} g_r$ (здесь g_i — это i -й символ строки g) содержит t как подстроку.

Вам поручено определить для каждого типа шифрования, могло ли сообщение s быть получено в результате его применения.

Формат входных данных

Первая строка входного файла содержит строку s ($1 \leq |s| \leq 100\,000$, где $|s|$ — длина строки s).

Вторая строка входного файла содержит целое число n — количество типов шифрования ($1 \leq n \leq 100\,000$). Последующие n строк содержат по два целых числа l_i, r_i и строку t_i , разделенные пробелами — характерные параметры i -го метода шифрования ($1 \leq l_i \leq r_i \leq |s|$).

Все строки состоят из строчных букв латинского алфавита. Суммарная длина всех t_i не превосходит 100 000.

Формат выходных данных

Выведите одну строку — для каждого типа шифрования «+», если сообщение s могло быть получено в результате его применения, или «-» в противном случае.

Пример

стандартный ввод	стандартный вывод
frommarsiam 3 6 10 i 2 11 am 1 9 human	++-

Задача Н. Чтение строк

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

Дано дерево T состоящее из n вершин (пронумерованных целыми числами от 1 до n). В каждой вершине записана некоторая буква. Корень дерева расположен в вершине 1.

Рассмотрим поддерево дерева T_v некоторой вершины v . Вдоль любого просто пути, начинающегося в v и заканчивающегося в некоторой вершине $u \in T_v$ (возможно, в самой v), можно прочитать некоторую строку. Обозначим количество **различных** строк, которые можно прочитать таким способом как $\text{dif}(v)$.

Дополнительно: для каждой вершины v дано целое число c_v . Нас интересуют вершины, в которых значение $\text{dif}(v) + c_v$ как можно больше.

Вы должны вычислить две величины — максимальное значение $\text{dif}(v) + c_v$ и количество вершин v с максимальным $\text{dif}(v) + c_v$.

Формат входных данных

В первой строке входных данных записано единственное целое число n ($1 \leq n \leq 300\,000$) — количество вершин в дереве T .

Во второй строке записано n целых чисел c_i ($0 \leq c_i \leq 10^9$).

В третьей строке записана строка s , состоящая из n строчных букв английского алфавита, — i -й символ этой строки соответствует букве, записанной в вершине i .

Далее следует $n - 1$ строка с описанием рёбер дерева T . Каждая из них содержит два целых числа u и v ($1 \leq u, v \leq n$), обозначающих ребро, которое соединяет вершины u и v .

Гарантируется, что входные данные описывают дерево.

Формат выходных данных

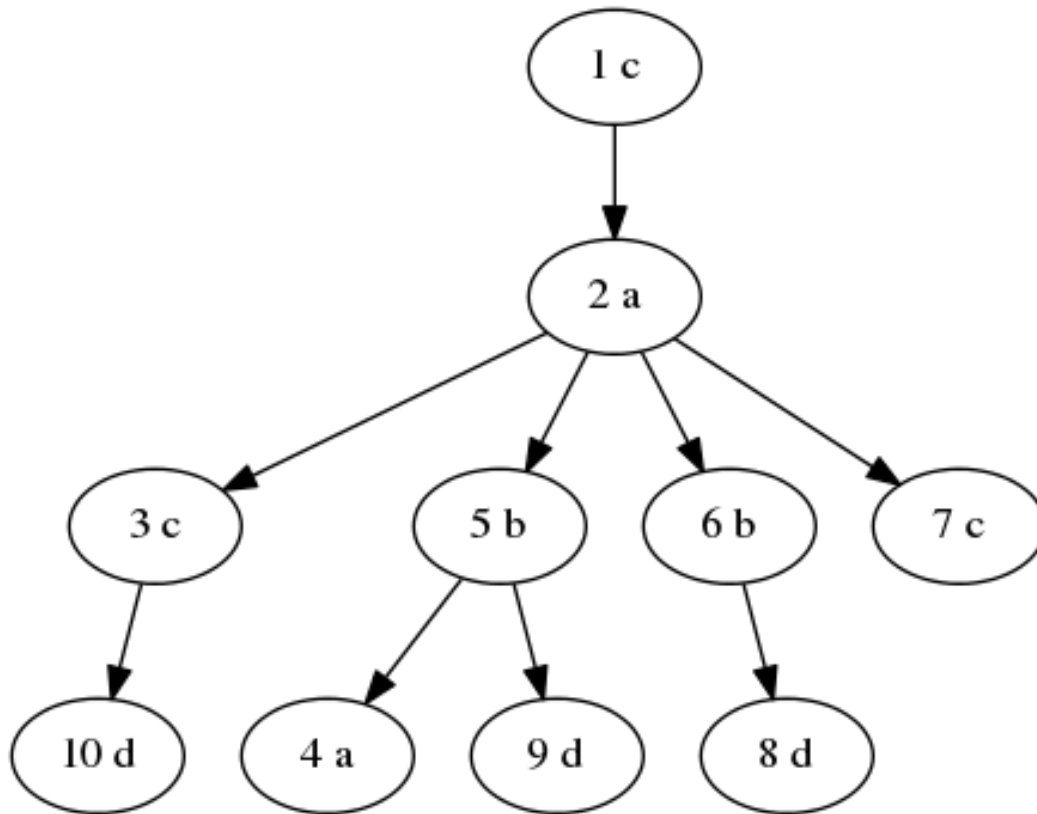
Выведите два числа — значение $m = \max(\text{dif}(i) + c_i)$ для всех $1 \leq i \leq n$ и количество вершин v , для которых $m = \text{dif}(v) + c_v$.

Примеры

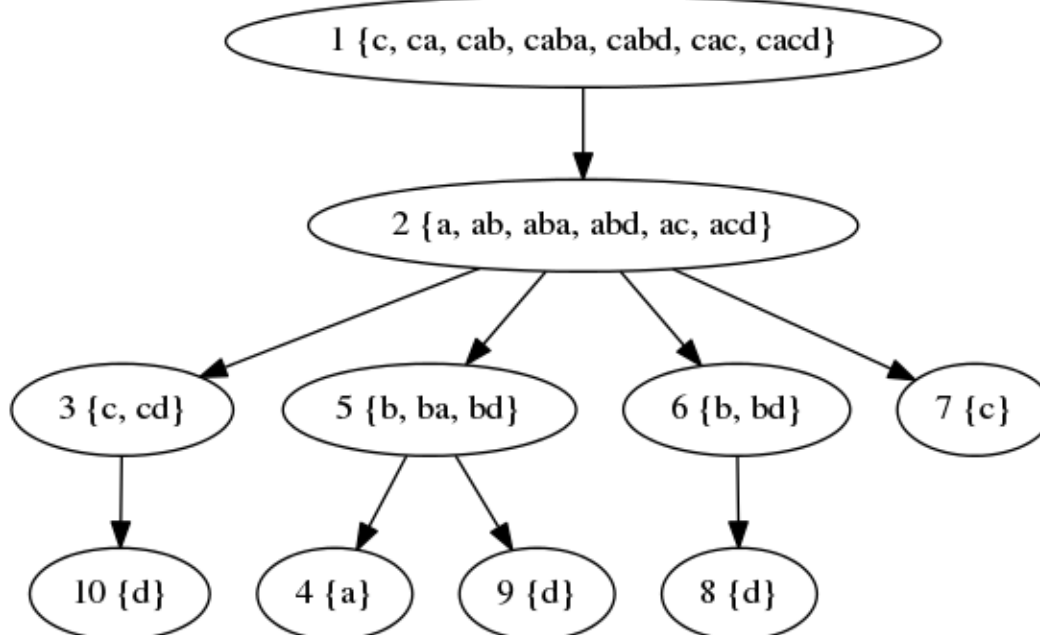
стандартный ввод	стандартный вывод
<pre>10 1 2 7 20 20 30 40 50 50 50 cacabbccddd 1 2 6 8 7 2 6 2 5 4 5 9 3 10 2 5 2 3</pre>	<pre>51 3</pre>
<pre>6 0 2 4 1 1 1 raaaba 1 2 2 3 2 4 2 5 3 6</pre>	<pre>6 2</pre>

Замечание

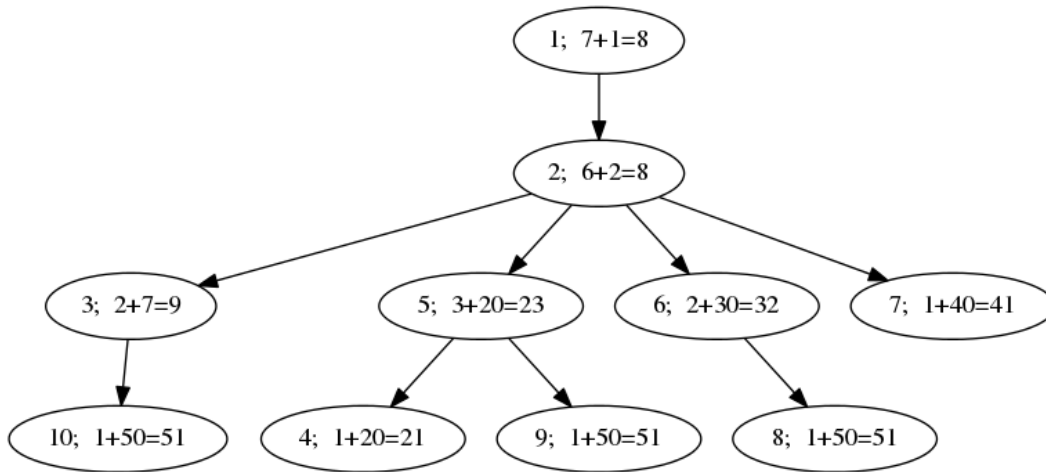
В первом примере дерево выглядит следующим образом:



Наборы строк, которые могут быть прочитаны из вершин:



Наконец, значения $\text{dif}(v) + c_v$ таковы:



Во втором примере значения $\text{dif}(1..n)$ таковы: $(5, 4, 2, 1, 1, 1)$. Различные строки, которые можно прочитать из вершины 2 таковы: a, aa, aaa, ab ; обратите внимание, что aa может быть прочитано как на пути до вершины 3, так и на пути до вершины 4.

Задача I. Вирусы

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

Комитет По Исследованию Бинарных Вирусов обнаружил, что некоторые последовательности единиц и нулей являются кодами вирусов. Комитет изолировал набор кодов вирусов. Последовательность из единиц и нулей называется безопасной, если никакой ее подотрезок (т.е. последовательность из соседних элементов) не является кодом вируса. Сейчас цель комитета состоит в том, чтобы установить, существует ли бесконечная безопасная последовательность из единиц и нулей.

Формат входных данных

Первая строка входного файла содержит одно целое число N , равное количеству всех вирусных кодов. Каждая из следующих n строк содержит непустое слово, составленное из символов 0 и 1 — код вируса. Суммарная длина всех слов не превосходит 30000.

Формат выходных данных

Первая и единственная строка выходного файла должна содержать слово:

- **ТАК** — если бесконечная, безопасная последовательность из нулей и единиц существует;
- **НIE** — в противном случае.

Примеры

стандартный ввод	стандартный вывод
3 01 11 00000	NIE
3 011 11 0000	ТАК

Задача J. Тандемные повторы

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

Дана строка s длины n .

Тандемным повтором в ней называются два вхождения какой-либо подстроки подряд. Иными словами, тандемный повтор описывается парой индексов $i < j$ такими, что подстрока $s[i \dots j]$ — это две одинаковые строки, записанные подряд.

От вас требуется посчитать количество пар индексом $i < j$ таких, что подстрока $s[i \dots j]$ является тандемным повтором.

Формат входных данных

Во входном файле находятся не более 30 тестов. Каждый тест состоит из единственной непустой строки, состоящей из символов **A,C,G,T**. Длина строки не превосходит 10^5 . Входной файл заканчивается строкой **0**.

Формат выходных данных

Для каждого теста выведите единственное число — количество тандемных повторов. Числа разделяйте переводами строк.

Пример

стандартный ввод	стандартный вывод
AGGA	1
AGAG	1
ATTTCGATTTCGATTTCG	9
AAAA	4
0	

Задача К. Словарь

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	1024 мегабайта

Словарь — это множество слов. Вы должны уметь обрабатывать запросы трех типов:

- «+ word» — добавить слово **word** в словарь, если оно в нем не присутствует.
- «- word» — удалить слово **word** из словаря, если оно там присутствует.
- «? text» — вычислить суммарное количество вхождений всех слов из словаря в текст **text**, при этом, если слово входит в текст несколько раз, то необходимо учесть каждое вхождение.

Гарантируется, что любое слово или текст являются непустыми строками, состоящими из букв **a**, **b** и **c**, суммарная длина которых не превосходит L . Однако, для упрощения задачи перед выполнением каждого запроса необходимо поступить следующим образом: пусть x обозначает ответ на последний запрос **?**, или 0, если таких запросов еще не было. Тогда необходимо очередную строку (**word** или **text**) циклически сдвинуть x раз. Напомним, что циклическим сдвигом строки $s = s_0s_1 \dots s_{|s|}$ называется строка $s' = s_1 \dots s_{|s|}s_0$.

Формат входных данных

В первой строке дано одно число Q — число запросов. В следующих Q строках находятся запросы. Суммарная длина строк во всех запросах не превосходит L ($L \leq 5\,000\,000$)

Формат выходных данных

Для каждого запроса «?» выведите одно число — ответ на него.

Пример

стандартный ввод	стандартный вывод
11	0
+ a	6
+ a	5
- a	7
- ab	
? abca	
+ ab	
+ a	
? abaaabb	
? aaabbab	
+ baa	
? babaca	