

Задача А. Задачка на строчечки

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

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 і-ой ма u
д о о . й t
о к - 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
о н в ч й л с рочечек нач д о 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 а
н а л о ъ а и а к с н р a
л и в н н в х де с ястю е т в ъ о м N п
ы р е е . р х е 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 ъ н e н д o x c и в и к ч e д к м к р
у ъ д : х с e м a в a
р н o к с к e d н и e т и д e в ы в a к ч и ъ н c o м
o e в o и т ч д c m
т л ы л в a к и p a в o л c з и і и к ч e ч o p k o л o
o a в e a л e ч
к м с и ч o к ъ л o x c e n и t c e в ы в a н ж л o d ж д к
к н у a
, e e ч и ч o т o п ѡ н т p a d н a t c a n и t c e в ы в a ю
к ж щ н
o х e т з и y к ч e в o л c o п - х a к ч e ч o p t c N х и a
в й
o к у б х и к c н и т a л х и к ъ н e л a m з и y k ч e ч o p t c т ё д

Формат входных данных

В стандартном поточике ввода ваша программочка найдёт строчечку из маленьких латинских буковок, которую мы назовём исходненькой. На следующей строчечке программочка найдёт числище N ($1 \leq N \leq 1\,000\,000$), а в следующих N строчечках — по словечку из тех же маленьких латинских буковок; эти словечки мы назовём словариком. Суммарненькая суммочка длинниц словечек из словарика не превосходит $1\,000\,000$.

Формат выходных данных

Ваша программочка должна вывести на стандартный поточичек вывода N строчечек. В i -й строчечке программочка должна вывести несколько чиселок: первое чиселко — количюсик (сколько штукечек) вхожденьечек строчечки i из словарика в исходненькой, затем через пробельчик для каждого вхожденьичка выведите индексики началиков всех вхожденьичек этой строчечки в исходненькую в отсортированном порядочке. Индексики всех строчечек начинаются с единичек. Няшечки-преподавашечки гарантируют, что колючюсик вхожденьичек не превосходит $1\,000\,000$.

Примеры

стандартный ввод	стандартный вывод
abracadabra	2 1 8
4	2 1 8
abra	0
ab	1 5
marazm	
cadabra	

Задача В. Множественный поиск 2

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 512 мегабайт

Дан массив строк S и строка t . Требуется для каждой строки $p \in S$ определить, сколько раз она встречается в t как подстрока.

Формат входных данных

Первая строка входного файла содержит целое число n — число элементов в S ($1 \leq n \leq 10^6$). Следующие n строк содержат по одной строке $p \in S$. Сумма длин всех строк из S не превосходит 10^6 . Последняя строка входного файла содержит t ($1 \leq t \leq 10^6$).

Все строки состоят из строчных латинских букв.

Формат выходных данных

Для каждой строки s_i выведите одно число: сколько раз она встречается в t .

Примеры

стандартный ввод	стандартный вывод
3	1
abc	0
abcdr	1
abcde	
xabcdef	

Задача С. Вирусы

Имя входного файла: `virus.in`
Имя выходного файла: `virus.out`
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 `virus.in` 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;
- **NIE** — в противном случае.

Примеры

<code>virus.in</code>	<code>virus.out</code>
3 01 11 00000	NIE
3 011 11 0000	ТАК

Задача D. Подозрительные строки

Имя входного файла:	<code>strings.in</code>
Имя выходного файла:	<code>strings.out</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Вы работаете в компании, специализирующейся в технологиях, связанных с интернетом, и ваш текущий проект — спам-фильтр. Фильтр определяет, содержит ли строка спам, используя *словарь спам-слов*. Если в строке содержится хотя бы одно слово из этого словаря как подстрока, фильтр считает, что исходная строка подозрительна.

Вы стали решать более интересную задачу: сколько существует различных подозрительных строк длины n , состоящих из строчных букв латинского алфавита для данного словаря спам-слов. Найдите ответ по модулю 10 000.

Формат входных данных

В первой строке содержатся два числа n и k ($1 \leq n \leq 100, 1 \leq k \leq 10$) — требуемая длина слов и количество слов в словаре спам-слов соответственно.

Следующие k строк являются строками словаря. Гарантируется, что они состоят из строчных латинских букв, они не пустые, и их длина не превосходит 10 символов.

Формат выходных данных

Выведите ответ по модулю 10 000.

Примеры

strings.in	strings.out
1 1 x	1
2 2 ab bb	2
5 2 ab bb	6350
5 2 aab bba	4054
5 9 xxxxxx xxx x уухуу xxxуxxx у уx ху zzzzzzzzzz	8752
100 10 aaaaaaaaaa bbbbbbbbbb cccccccccc dddddddddd eeeeeeeeee ffffffffff gggggggggg hhhhhhhhhh xxxxxxxxxx zzzzzzzzzz	9350

Задача Е. Мультимножество Василия

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

У автора уже закончились истории про Василия, поэтому он просто написал формальную постановку задачи.

У вас есть q запросов и мультимножество A , изначально содержащее только число 0. Запросы бывают трёх видов:

- «+ x » — добавить в мультимножество A число x .
- «- x » — удалить одно вхождение числа x из мультимножества A . Гарантируется, что хотя бы одно число x в этот момент присутствует в мультимножестве.
- «? x » — вам даётся число x , требуется вычислить $\max_{y \in A} x \oplus y$, то есть максимальное значение побитового исключающего ИЛИ (также известно как XOR) числа x и какого-нибудь числа y из мультимножества A .

Мультимножество — это множество, в котором разрешается несколько одинаковых элементов.

Формат входных данных

В первой строке входных данных содержится число q ($1 \leq q \leq 200\,000$) — количество запросов, которые требуется обработать Василию.

Каждая из последующих q строк входных данных содержит один трёх символов «+», «-» или «?» и число x_i ($1 \leq x_i \leq 10^9$). Гарантируется, что во входных данных встречается хотя бы один запрос «?».

Обратите внимание, что число 0 всегда будет присутствовать в мультимножестве.

Формат выходных данных

На каждый запрос типа «?» выведите единственное целое число — максимальное значение побитового исключающего ИЛИ для числа x_i и какого-либо числа из мультимножества A .

Примеры

стандартный ввод	стандартный вывод
10	11
+ 8	10
+ 9	14
+ 11	13
+ 6	
+ 1	
? 3	
- 8	
? 3	
? 8	
? 11	

Замечание

После первых пяти операций в мультимножестве A содержатся числа 0, 8, 9, 11, 6 и 1.

Ответом на шестой запрос будет число $11 = 3 \oplus 8$ максимальное из чисел $3 \oplus 0 = 3$, $3 \oplus 9 = 10$, $3 \oplus 11 = 8$, $3 \oplus 6 = 5$ и $3 \oplus 1 = 2$.

Задача F. Электронное правительство

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

В рамках проекта «Электронное правительство» лучшим программистам страны Распиляндии поручили создание системы автоматизации сбора статистики и анализа прессы.

Известно, что членами правительства Распиляндии может стать любой из k граждан. Их фамилии — a_1, a_2, \dots, a_k . Все фамилии различны. Изначально в правительство входят все k граждан из этого списка. Система должна поддерживать следующие возможности:

- Включить гражданина a_i в состав правительства.
- Исключить гражданина a_i из состава правительства.
- По заданному тексту статьи определить, насколько она политизирована. Для этого, для каждого действующего члена правительства, вычисляется количество раз, которое его фамилия встречается в тексте как подстрока. Считаются все вхождения, в том числе и пересекающиеся. Степень политизированности текста определяется как сумма этих количеств по всем действующим членам правительства.

Реализуйте эту систему.

Формат входных данных

В первой строке через пробел записаны числа n и k ($1 \leq n, k \leq 10^5$) — количество запросов к системе и количество потенциальных членов правительства.

В следующих k строках заданы фамилии a_1, a_2, \dots, a_k , по одной на строку. Все фамилии попарно различны.

В следующих n строках заданы запросы к системе, по одному на строку. Каждый запрос состоит из символа, определяющего операцию, и аргумента операции, записанных подряд без пробела.

Операции «включить в правительство» соответствует символ «+», операции «исключить» — «-». Аргументом этих операций является целое число от 1 до k — номер гражданина, которого она затрагивает. Любой гражданин может быть включен и исключен из правительства произвольное количество раз в любом порядке. Включение в правительство гражданина, который в него уже входит, или исключение гражданина, который и так не входил, ничего не меняет.

Операции «подсчитать политизированность» соответствует символ «?». Её аргументом является текст.

Все строки, как фамилии, так и тексты, являются непустыми последовательностями строчных букв латинского алфавита. Суммарная длина всех фамилий не превышает 10^6 , суммарная длина всех текстов не превышает 10^6 .

Формат выходных данных

Для каждой операции «подсчитать политизированность» выведите на отдельной строке степень политизированности заданного в ней текста. Для остальных операций ничего выводить не нужно.

Примеры

стандартный ввод	стандартный вывод
7 3	6
a	4
aa	3
ab	6
?aaab	
-2	
?aaab	
-3	
?aaab	
+2	
?aabbaa	

Задача G. Идеальная пара

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 5 секунд
Ограничение по памяти: 1024 мегабайта

В мире Барби важно, чтобы любая пара Барби-Кен была идеальна, ведь нет ничего прекрасней идеального. Но не всем везет в поиске партнера, поэтому был создан «Клуб Поиска Идеального Партнера».

Для того что бы облегчить поиск партнера, каждый впервые пришедший должен был пройти опрос, а после, в зависимости от ответов, каждой Барби и каждому Кену выдавалась строка, означающая его тип. Она так же размещалась на специальной доске в здании клуба, где любой желающий мог подойти и присмотреть себе подходящего партнера.

К несчастью, никто не хотел переводить специальные строки в более понятные характеристики, поэтому партнеров стали выбирать так: брали строчку Барби и склеивали ее со строчкой Кена, и если получившаяся строчка оказывалась палиндромом, то пара считалась идеальной, ведь не существует строки идеальнее палиндрома.

В скором времени для того, чтобы собирать статистику по возможным идеальным парам, решили после каждого обновления доски выводить, сколько идеальных пар может быть создано из анкет, закрепленных на доске. Доска представляет из себя два поля: поле со строками Кенов и поле со строками Барби. Изначально на доске нет ни одной строки. И каждый раз, когда кто-то проходит тест, после его окончания результат закрепляется на доску. Также иногда кто-то из участников клуба может найти себе партнера (из клуба или снаружи), после чего его строку снимают с доски.

Для каждого изменения доски выведите, сколько идеальных пар можно составить из текущего набора строк.

Формат входных данных

В первой строке ввода дано целое число t — количество изменений доски ($1 \leq t \leq 10^6$).

Изменения задаются следующим образом:

- « $1 + b$ » — Барби пришла в клуб, и по результатам теста получила строку b .
- « $1 - b$ » — Барби со строкой b нашла себе партнера и ушла из клуба. Гарантируется, что такая строка находилась на доске.
- « $2 + k$ » — Кен пришел в клуб, и по результатам теста получил строку k .
- « $2 - k$ » — Кен со строкой k нашел себе партнера и ушел из клуба. Гарантируется, что такая строка находилась на доске.

Гарантируется, что суммарная длина строк по всем запросам первого и третьего типа (новая строка) не превосходит $5 \cdot 10^6$.

Формат выходных данных

После каждого изменения доски выведите, сколько идеальных пар можно составить из текущих строк.

Пример

стандартный ввод	стандартный вывод
6	0
1 + ken	1
2 + nek	1
1 + barbie	2
2 + ibrab	1
1 - ken	0
1 - barbie	