

## Задача А. Задачка на строчечки

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

### Формат входных данных

В стандартном поточике ввода ваша программочка найдёт строчечку из маленьких латинских букв, 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 букв; 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;
- **НIE** — 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 ух ху zzzzzzzzzz	8752
100 10 aaaaaaaaaa bbbbbbbbbb cccccccccc dddddddddd eeeeeeeeee fffffffffff gggggggggg hhhhhhhhhh xxxxxxxxxxx zzzzzzzzzz	9350

## Задача Е. Пароли

Имя входного файла: `cofnsubstrings.in`  
Имя выходного файла: `cofnsubstrings.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Павлу очень часто нужно придумывать различные новые секретные пароли. Каждый пароль должен иметь длину ровно  $l$  символов и может состоять только из строчных латинских букв.

Для того, чтобы не забывать придуманные пароли, Павел решил прятать в них некоторые ключевые слова. Для этого он подготовил множество из  $n$  ключевых слов, которые должны встречаться в пароле как подстроки. При этом он решил, что в придуманном пароле должны встречаться ровно  $c$  из ключевых слов, каждое возможно несколько раз.

Павел просит вас посчитать сколько всего существует различных подходящих паролей.

### Формат входных данных

Первая строка содержит единственное целое число  $n$  ( $1 \leq n \leq 6$ ) — количество ключевых слов.

Следующие  $n$  строк содержат по одному непустому ключевому слову в каждой строке. Каждое слово состоит из не более чем 50 символов.

Последняя строка содержит два целых числа  $c$  и  $l$  ( $0 \leq c \leq n, 1 \leq l \leq 50$ ) — количество требуемых слов из множества и длина пароля, соответственно.

### Формат выходных данных

Выведите количество паролей по модулю 1 000 000 009.

### Примеры

<code>cofnsubstrings.in</code>	<code>cofnsubstrings.out</code>
4 a aa aaa aaaa 2 3	50
1 abcdefgh 0 7	31810104

## 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	