

## Задача А. 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 программо  
 ч  
 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 и  
 д о о . й т  
 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 о а  
 н а 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 к ч e д к м к р  
 у ь д : x c e м a v a  
 р н o к с 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 і 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 v ы в a н ж л o d ж д к  
 к н у a  
 , e e ч и ч o т o п й н т p a d н a t c a n и t c e v ы в a ю  
 к ж щ н  
 o х e t з и y k ч e v o l c o p - 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 д

## Формат входных данных

В стандартном поточике вводика или файлике `inputik.txt` 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 выводика или в файл `outputik.txt`  $N$  строчечек. В  $i$ -ой строчечке программочка должна вывести несколько чиселок: первое чиселко — количюсик (сколько штучечек) вхожденьчек строчечки  $i$  из словарика в исходненькой, затем через пробельчик для каждого вхожденьчика выведите индексики началиков всех вхожденьчик этой строчечки в исходненькую в отсортированном порядочке. Индексики всех строчечек начинаются с единичек. Няшечки-преподавашечки гарантируют, что колчюсик вхожденьчик 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 В. Сортировка слов

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

Одна из новых возможностей текстового редактора «World XP» – это сортировка слов в предложении. Выход новой бета-версии редактора должен состояться не позднее, чем через пять часов, а заявленная функция еще не реализована.

Требуется написать программу, осуществляющую сортировку слов в предложении. При этом все символы, отличные от букв, должны сохраниться и не поменять своего положения относительно вхождений слов. Для упрощения при подаче входных данных на вход вашей программы все такие символы будут заменены на символ «.» (точка). Таким образом символ «.» имеет смысл разделителя между словами. Например, строка «.aba.a.ba» после сортировки примет вид «.a.aba.ba», а строка «с.bb.a» примет вид «a.bb.c». Слова следует сортировать лексикографически, как в словаре.

### Формат входных данных

Входной файл содержит единственную строку, содержащую только прописные латинские буквы и символ «.». Слова могут разделяться любым количеством символов «.», строка может как начинаться, так и заканчиваться последовательностью точек. Длина заданной строки не менее 1 символа и не превосходит  $10^6$  символов.

### Формат выходных данных

В выходной файл выведите строку после сортировки слов в ней.

### Примеры

стандартный ввод	стандартный вывод
. .aba.a. .ba	. .a.aba. .ba
с. .bb.a	a. .bb.c

## Задача С. Под-бор

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

Бором называется подвешенное дерево, на каждом из рёбер которого написано по символу, причём символы, написанные на рёбрах, выходящих из общей вершины-родителя, различны. Будем называть направление от родителя к детям “вниз”. Назовем вхождением строки  $s$  в бор такую вершину бора, от которой можно пройти несколько шагов вниз таким образом, что встретившиеся символы образуют строку  $s$ .

Даны бор и несколько строк, найдите сумму количеств вхождений этих строк в этот бор.

### Формат входных данных

В первой строке входного файла записано единственное число  $n$ ,  $1 \leq n \leq 100\,000$  — количество вершин бора. В следующих  $n$  строках описаны вершины бора. В  $(i + 1)$ -й строке описаны дети  $i$ -й вершины: число  $k_i$  ее детей, затем  $k_i$  пар из номера вершины-ребёнка и символа, написанного на соответствующем ребре. Номер родителя всегда меньше номера ребёнка; корнем бора является вершина номер 1.

В  $(n + 2)$ -й строке записано количество  $m$  ( $1 \leq m \leq 100\,000$ ) строк для поиска. В следующих  $m$  строках перечислены сами строки. Входные строки непусты, а их суммарная длина не превышает  $100\,000$  символов.

Все символы, написанные на рёбрах, а также все символы, составляющие строки — маленькие латинские буквы.

### Формат выходных данных

Выведите одно число — сумму количеств вхождений.

### Пример

стандартный ввод	стандартный вывод
7	9
2 2 a 4 b	
2 3 a 6 b	
0	
1 5 b	
1 7 b	
0	
0	
4	
b	
bb	
bbb	
bb	

---

## Задача D. Подозрительные строки - HARD

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
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  $k$  ( $1 \leq n < 2^{31}, 1 \leq k \leq 10$ ) — 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

стандартный ввод	стандартный вывод
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
2147483647 10 aaaaaaaaaa bbbbbbbbbb cccccccccc dddddddddd eeeeeeeeee fffffffffff gggggggggg hhhhhhhhhh xxxxxxxxxxx zzzzzzzzzz	5040

## Задача Е. 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 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 вывода должна содержать слово:

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

### Примеры

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

## Задача F. 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 G. Расшифровка ДНК

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

Ученые работают на раскопках окаменелых останков древних существ на планете соседней звездной системы. В процессе исследования ученые пытаются понять, как именно цепочки ДНК различных существ составлялись из генов.

Цепочки ДНК всех исследуемых существ представляют собой последовательности нуклеотидов. Каждый нуклеотид обозначается строчной буквой латинского алфавита. Таким образом, цепочка ДНК представляет собой строку, составленную из строчных букв латинского алфавита.

Ген также представляет собой строку из строчных букв латинского алфавита. Известно, что в любом корректном наборе генов никакая строка не является префиксом другой строки.

Будем говорить, что цепочку ДНК  $d$  можно расшифровать с использованием набора генов  $G$ , если  $d$  можно представить как результат последовательной записи одного или нескольких генов:  $d = g_1g_2 \dots g_k$ , где  $g_i$  — гены из набора  $G$ . Один и тот же ген может входить в расшифровку ДНК несколько раз.

Для обработки информации ученым требуется разработать компьютерную систему, которая будет поддерживать корректный набор генов  $G$  и массив цепочек ДНК существ  $D$ . По мере анализа останков, ученые могут добавлять новый ген в набор  $G$  или добавлять новую цепочку ДНК в массив  $D$ . Гарантируется, что ни в какой момент времени не существует двух генов, один из которых является префиксом другого.

После каждой операции ученые хотят знать, какие цепочки ДНК в массиве  $D$  можно расшифровать, используя текущий набор генов  $G$ . После  $i$ -й операции система должна сообщать  $k_i$  — количество цепочек ДНК, находящихся в массиве  $D$ , которые впервые стало можно расшифровать после  $i$ -й операции, а затем  $k_i$  чисел — номера этих цепочек. Результат очередной операции должен быть получен до того, как станет известна следующая операция.

Помогите ученым разработать такую систему.

### Формат входных данных

В первой строке находится число  $n$  — количество операций, которые необходимо выполнить ( $1 \leq n \leq 100\,000$ ).

В следующих  $n$  строках находятся описания операций,  $i$ -я строка начинается с символа «+», если эта операция — добавление нового гена в набор  $G$ , или с символа «?», если эта операция — добавление цепочки ДНК в конец массива  $D$ . Далее через пробел находится строка  $x_i$ , состоящая из строчных латинских букв, которую необходимо использовать, чтобы получить строку  $s_i$ , которая задает добавляемый в этой операции ген или цепочку ДНК.

Для получения строки  $s_i$  из строки  $x_i$ , необходимо выполнить следующие действия. Если  $i = 1$ , то  $s_i = x_i$ . Иначе пусть число впервые расшифрованных цепочек ДНК после предыдущей операции равно  $k_{i-1}$ . Выполним  $k_{i-1}$  раз следующее действие: перенесем первый символ  $x_i$  в конец. Иначе говоря, выполним циклический сдвиг строки  $x_i$  влево на  $k_{i-1}$ . Получившаяся строка равна  $s_i$  — ген или цепочка ДНК, которую необходимо добавить на  $i$ -й операции.

Все строки не пусты, суммарный размер строк во всех операциях не превышает  $10^6$ .

Гарантируется, что ни в какой момент времени не существует двух генов, один из которых является префиксом другого.

### Формат выходных данных

Выведите  $n$  строк.

В  $i$ -й строке выведите сначала число  $k_i$  — количество цепочек ДНК, находящихся в массиве  $D$ , которые впервые стало можно расшифровать после  $i$ -й операции, а затем  $k_i$  чисел — номера этих цепочек. Цепочки нумеруются с единицы в порядке добавления в массив  $D$ . Номера цепочек в одной строке можно выводить в любом порядке.

**Пример**

стандартный ввод	стандартный вывод
5	0
? abcabd	0
+ abc	1 2
? abcabc	0
? dabdab	2 1 3
+ abd	

**Замечание**

В первых трех операциях  $s_1$ ,  $s_2$  и  $s_3$  совпадают с соответствующими строками во вводе. Поскольку  $k_3 = 1$ , то для четвертой операции  $s_4$  получается из строки  $x_4 = \text{«dabdab»}$  циклическим сдвигом влево на 1, таким образом, в четвертой операции в массив  $D$  добавляется строка  $s_4 = \text{«abdabd»}$ . Наконец,  $k_4 = 0$ , поэтому  $s_5 = x_5$ .

## Задача Н. 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	

## Задача I. Divljak

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	768 мегабайт

Это история про  $N$  варваров и Тарзана. Варвары занумерованы числами от 1 до  $N$ , и у каждого на табличке написана строка из строчных букв латинского алфавита. Игра состоит из  $Q$  раундов, каждый из которых может быть одного из двух типов:

1. Тарзан показывает слово  $P$  варварам.
2. Тарзан спрашивает варвара номер  $i$ , сколько слов из показанных им до текущего момента таковы, что слово на табличке варвара является их подстрокой.

Варвары не очень любят задачи на строчки, поэтому помогите им ответить на все запросы Тарзана.

### Формат входных данных

Первая строка содержит число варваров  $N$  ( $1 \leq N \leq 10^5$ ).

Каждая из следующих  $N$  строк содержит слово на табличке очередного варвара.

Далее идет число запросов  $Q$  ( $1 \leq Q \leq 10^5$ ).

Каждая из следующих  $Q$  строк описывает очередной запрос. Сперва записано число  $t$  — тип запроса. Если  $t$  равно 1, то далее записано слово  $P$ , которое показывает Тарзан. Иначе  $t$  равно 2 и далее записан номер варвара  $i$  ( $1 \leq i \leq N$ ), которого спрашивает Тарзан.

Суммарная длина слов на табличках варваров  $\leq 2 \cdot 10^6$ .

Суммарная длина показанных Тарзаном слов  $\leq 2 \cdot 10^6$ .

### Примеры

стандартный ввод	стандартный вывод
3 a bc abc 3 1 abca 2 1 2 3	1 1
7 abba bbaa b bbaa abba a ba 7 1 aaabbabbaab 2 7 1 baabaaa 1 aabbbab 2 3 1 aabba 2 3	1 3 4