

## Задача А. Молекулярная биология

Имя входного файла:	dict.in
Имя выходного файла:	dict.out
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Напомним, что важнейшей задачей молекулярной биологии является изучение различных белков. В общем случае один и тот же белок имеет одну из четырех форм (первичная, вторичная, третичная и четвертичная структура), простейшей с точки зрения химии является первичная, которую мы и рассмотрим.

Белок представляет собой последовательность аминокислот, соединенных пептидной связью (эта связь одинакова между любыми двумя аминокислотами, поэтому рассматриваться она не будет). Аминокислота, как следует из названия, — это органическое соединение, в молекуле которого одновременно содержатся карбоксильные и аминные группы. Для простоты будем считать, что аминокислота состоит из химических элементов, каждый элемент будем обозначать маленькой латинской буквой, таким образом, аминокислота однозначно определяется строкой из нескольких латинских букв. Безусловно, не любая последовательность элементов является аминокислотой.

Для каждого организма известен большой список аминокислот, которые могут быть синтезированы в организме такого типа. Для удобства транскрипции (это один из этапов синтеза РНК), ни одна аминокислота не является началом другой.

Одна из часто встречающихся задач заключается в том, чтобы понять, мог ли данный белок быть синтезирован в данном организме (это позволяет понять по белку, где он был синтезирован: у человека, у гриба и пр.). Для этого нужно проверить, все ли аминокислоты, из которых он состоит, могли быть синтезированы. Однако, часто белок содержит лишние элементы (например, к белку из-за обилия атомов водорода очень часто присоединяются молекулы воды), поэтому задача сводится к изучению фрагментов этого белка. Проводят как отдельные проверки каких-то фрагментов, так и целые серии проверок случайных фрагментов.

Более формально, вам дан словарь из аминокислот, каждая из которых представляет собой строку из маленьких латинских букв, и белок, который также представляет собой строку из маленьких латинских букв. Необходимо ответить на несколько вопросов вида «мог ли фрагмент белка с позиции  $l$  по  $r$  быть синтезирован из данных аминокислот», т. е. можно ли данный фрагмент разбить на слова из словаря. В случае положительного ответа, необходимо предъявить минимальное количество слов в таком разбиении.

### Формат входных данных

В первой строке входного файла содержится одно число  $N$  — количество аминокислот. Далее следует  $N$  непустых строк, каждая из которых является описанием аминокислоты. Суммарная длина строк  $S$  не превосходит  $10^6$ .

В следующей строке содержится исходный белок, длина  $L$  которого не превосходит  $10^6$ .

Далее содержится число  $M$  — количество простых запросов ( $0 \leq M \leq 10^5$ ). Каждый простой запрос описывается двумя числами —  $l$  и  $r$ , ( $1 \leq l \leq r \leq L$ ), обозначающими искомый фрагмент  $[l, r]$ , который необходимо разбить.

Далее содержится число  $K$  — количество запросов-серий ( $0 \leq K \leq 10^5$ , в некоторых группах тестов запросы-серии отсутствуют и  $K = 0$ ). Каждая серия описывается восемью целыми числами  $T, A, B, C, D, E, l_1, r_1$  ( $1 \leq A, B, C, D \leq 100$ ,  $1 \leq T, E, l_1, r_1 \leq 10^7$ ). Такая серия описывает  $T$  простых запросов. Запрос с номером  $i$  выглядит как фрагмент  $[\min(l_i \bmod L + 1, r_i \bmod L + 1), \max(l_i \bmod L + 1, r_i \bmod L + 1)]$ . Числа  $l_i$  и  $r_i$  для  $i \geq 2$  определяются следующим образом. Пусть  $ans$  обозначает ответ на запрос с номером  $i-1$  в этой серии. Тогда  $l_i = (A \cdot l_{i-1} + B) \bmod E$ ,  $r_i = (C \cdot r_{i-1} + D + ans) \bmod E$ . Суммарное число запросов в сериях не превосходит  $10^7$ .

Все строки во входном файле состоят только из маленьких латинских букв. Гарантируется, что ни одна аминокислота не является префиксом другой.

### Формат выходных данных

Для каждого простого запроса выведите единственное число — минимальное число аминокислот,

на которое разбивается соответствующий фрагмент. В случае, если разбиения не существует, ответом на запрос является число  $-1$ . Для каждой серии выведите одно число — сумму ответов на все запросы, для которых ответ существует (то есть не равен  $-1$ ), взятую по модулю  $E$ .

## Примеры

dict.in	dict.out
2	1
abra	1
cadabra	-1
abracadabra	1
6	2
1 4	-1
8 11	
4 8	
5 11	
1 11	
3 6	
0	
2	3
ab	
ba	
ababab	
0	
1	
10 1 2 3 4 5 6 7	

## Замечание

Тесты состоят из четырех групп.

1. Тесты 1–2, из условия, оцениваются в 0 баллов.
2. В тестах этой группы общая длина словарных аминокислот  $S$  не превосходит 5 000, длина белка  $L$  не превосходит 5 000, число простых запросов  $M$  не превосходит 5 000, запросы-серии отсутствуют ( $K = 0$ ). Эта группа оценивается в 30 баллов, баллы начисляются только при прохождении всех тестов группы.
3. В тестах этой группы суммарная длина аминокислот  $S$  не превосходит 500 000, длина белка  $L$  не превосходит 5 000, число простых запросов  $M$  не превосходит 5 000, запросы-серии отсутствуют ( $K = 0$ ). Эта группа оценивается в 30 баллов, баллы начисляются только при прохождении всех тестов группы.
4. В тестах этой группы суммарная длина аминокислот  $S$  не превосходит 500 000, длина белка  $L$  не превосходит 100 000, число простых запросов  $M$  не превосходит 100 000, запросы-серии отсутствуют ( $K = 0$ ). Эта группа оценивается в 20 баллов, баллы начисляются только при прохождении всех тестов группы.
5. Offline-группа. Баллы за тесты этой группы начисляются только при прохождении всех тестов всех предыдущих групп. Некоторые тесты этой группы объединяются в подгруппы, тесты за каждую подгруппу ставятся только при прохождении всех тестов подгруппы.

## Задача В. 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 С. Под-бор

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

### Примеры

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

## Задача D. Динамический поиск подстрок

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 6 секунд  
Ограничение по памяти: 256 мегабайт

*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.

### Примеры

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

## Задача Е. Поиск набора образцов

Имя входного файла: `console2.in`  
Имя выходного файла: `console2.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Напишите программу, которая для каждой строки из заданного набора  $S$  проверяет, верно ли, что она содержит как подстроку одну из строк из набора  $T$ .

### Формат входных данных

Первая строка входного файла содержит натуральное число  $n$  ( $1 \leq n \leq 1000$ ) — количество строк в наборе  $T$ . Каждая из следующих  $n$  строк содержит непустую строку. Гарантируется, что суммарная длина всех строк из набора  $T$  не превышает 80 000. Оставшаяся часть файла содержит строки из набора  $S$ . Каждая строка состоит из ASCII символов с кодами от 32 до 126 включительно. Строка может быть пустой. Гарантируется, что размер входного файла не превышает 1 МБ.

### Формат выходных данных

В выходной файл выведите все строки из набора  $S$  (в том порядке, в котором они находятся во входном файле), содержащие как подстроку по крайней мере одну строку из набора  $T$ .

### Примеры

<code>console2.in</code>	<code>console2.out</code>
<code>3</code>	<code>sudislavl</code>
<code>gr</code>	<code>group b</code>
<code>sud</code>	
<code>abc</code>	
<code>lksh</code>	
<code>sudislavl</code>	
<code>kostroma</code>	
<code>summer</code>	
<code>group b</code>	

## Задача F. Задачечка 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 программо  
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  
на л о ь а и а к с н р а  
л и в н н в х де с ястю е т в ь 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 о о  
и к п д к у ч ж е а ч а г  
ю к и о и к ь н е н д о x с и в и к ч e д к м к р  
у ь д : х с е м а в а  
р н о к с к e d н и e т и д e в ы в а к ч и ь н с o м  
о e в o и т ч д с м  
т л ы л в а к и р a в o л с з и і и к ч e ч o p k o л o  
о а в e а л e ч  
к м с и ч o к ь л o x e n и т с e в ы в a н ж л o d ж д к  
к н у а  
, e e ч и ч o т o п й н т p a d н a t c a n и т с e в ы в a ю  
к ж щ н  
о х e t з и y k ч e v o l s o p - х a k ч e ч o p t c N х и а  
в й  
о к у 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 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) вхожденьчечек 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 количюсик 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	