

Задача А. К-я строка

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

Реализуйте структуру данных, которая поддерживает следующие операции:

- добавить в словарь строку S ;
- найти в словаре k -ю строку в лексикографическом порядке.

Известно, что изначально словарь пуст.

Формат входных данных

Первая строка входного файла содержит натуральное число N — количество команд ($N \leq 10^5$). Последующие N строк содержат по одной команде каждая.

Команда записывается либо в виде числа k , либо в виде строки S , которая может состоять только из строчных латинских букв. Гарантируется, что при запросе k -й строки она существует. Также гарантируется, что сумма длин всех добавляемых строк не превышает 10^5 .

Формат выходных данных

Для каждого числового запроса k выходной файл должен содержать k -ю в лексикографическом порядке строчку из словаря на момент запроса. Гарантируется, что суммарная длина строк в выходном файле не превышает 10^5 .

Примеры

<code>kthstr.in</code>	<code>kthstr.out</code>
7	tolstoy
pushkin	gogol
lermontov	
tolstoy	
gogol	
gorkiy	
5	
1	

Задача В. 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	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 С. 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 D. 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 E. Contain Us All

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

Даны строки p_1, \dots, p_n из заглавных английских букв. Надо найти строку минимальной длины, которая содержит каждую из строчек p_1, \dots, p_n как подстроку. Если таких строчек несколько, нужно выбрать лексикографически минимальную.

Формат входных данных

Первая строчка содержит число n ($1 \leq n \leq 12$) — количество строчек.

Следующие n строк содержат данные строчки p_i . Гарантируется, что каждая строчка состоит только из заглавных букв, а также что длина каждой строчки не превосходит 50

Формат выходных данных

Выведите ответ на задачу.

Примеры

стандартный ввод	стандартный вывод
2 ВАВ АВА	АВАВ

Задача F. x-простые подстроки

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

Огромное спасибо Михаилу MikeMirzayanov Мирзаянову за невероятные системы Polygon и CodeForces, откуда и взята была эта замечательная задача.

Дано целое число x и строка s , состоящая из цифр от 1 до 9 включительно.
Подстрокой строки называется последовательная подпоследовательность этой строки.
Пусть $f(l, r)$ будет равно сумме цифр в подстроке $s[l..r]$.
Назовем подстроку $s[l_1..r_1]$ x -простой, если

- $f(l_1, r_1) = x$;
- не существует таких значений l_2, r_2 , что
 - $l_1 \leq l_2 \leq r_2 \leq r_1$;
 - $f(l_2, r_2) \neq x$;
 - x делится на $f(l_2, r_2)$.

Разрешено удалять любые символы из строки. Если вы удаляете символ, то две полученные части строки склеиваются, не меняя порядок.

Какое минимальное количество символов надо удалить из строки, чтобы она не содержала x -простых подстрок? Если x -простых подстрок нет в данной строке s , то выведите 0.

Формат входных данных

В первой строке записана строка s ($1 \leq |s| \leq 1000$). s содержит только цифры от 1 до 9 включительно.

Во второй строке записано одно целое число x ($1 \leq x \leq 20$).

Формат выходных данных

Выведите одно целое число — минимальное количество символов, которые надо удалить из строки, чтобы она не содержала x -простых подстрок? Если x -простых подстрок нет в данной строке s , то выведите 0.

Примеры

стандартный ввод	стандартный вывод
116285317 8	2
314159265359 1	2
13 13	0
3434343434 7	5

Замечание

В первом примере в строке две 8-простых подстроки «8» и «53». Можно удалить данные символы, чтобы избавиться от обеих: «116285317». Полученная строка «1162317» не содержит 8-простых подстрок. Также можно удалить такие символы: «116285317».

Во втором примере необходимо просто удалить обе единицы.

В третьем примере нет 13-простых подстрок. В нем вообще нет подстрок с суммой цифр равной 13.

В четвертом примере в строке не должно быть ни «34», ни «43». Поэтому необходимо удалить либо все тройки, либо все четверки. Их по 5 штук, поэтому можно удалить любые из них.

Задача G. 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	