

Задача А. Суффиксный бор

Имя входного файла: `trie.in`
Имя выходного файла: `trie.out`
Ограничение по времени: 2 секунды
Ограничение по памяти:

Постройте суффиксный бор данной строки s .

Формат входных данных

Первая строка входного файла содержит строку s ($1 \leq |s| \leq 100$). Строка состоит из строчных латинских букв.

Формат выходных данных

В первой строке выходного файла выведите два натуральных числа n и m , разделенных пробелом — число вершин и ребер в суффиксном боре соответственно. В следующих m строках выведите описания ребер в формате `<родитель> <потомок> <символ>`. Корнем бора должна быть вершина с номером 1. Вершины должны быть занумерованы натуральными числами, не превышающими n .

Примеры

<code>trie.in</code>	<code>trie.out</code>
ababb	12 11 1 2 a 1 7 b 2 3 b 3 4 a 3 11 b 4 5 b 5 6 b 7 8 a 7 12 b 8 9 b 9 10 b

Задача В. Суффиксный массив

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

Постройте суффиксный массив для заданной строки s .

Формат входных данных

Первая строка входного файла содержит строку s ($1 \leq |s| \leq 400\,000$). Строка состоит из строчных латинских букв.

Формат выходных данных

Выведите $|s|$ различных чисел — номера первых символов суффиксов строки s так, чтобы соответствующие суффиксы были упорядочены в лексикографически возрастающем порядке.

Примеры

<code>array.in</code>	<code>array.out</code>
ababb	1 3 5 2 4

Задача С. Суффиксное дерево

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

Постройте суффиксное дерево для заданной строки s .

Формат входных данных

Первая строка входного файла содержит строку s ($1 \leq |s| \leq 100\,000$). Строка состоит из строчных латинских букв.

Формат выходных данных

В первой строке выходного файла выведите два натуральных числа n и m , разделенных пробелом — число вершин и ребер в суффиксном дереве соответственно. В следующих m строках выведите описания ребер в формате $\langle \text{родитель} \rangle \langle \text{потомок} \rangle \langle l \rangle \langle r \rangle$. Эта запись означает, что на ребре написана строка $s[l..r]$, при этом значение l должно быть минимально возможным. Корнем дерева должна быть вершина с номером 1. Вершины должны быть занумерованы натуральными числами, не превышающими n .

Примеры

tree.in	tree.out
ababb	7 6 1 4 1 2 1 6 2 2 4 2 3 5 4 5 5 5 6 3 3 5 6 7 5 5

Задача D. Суффиксный массив 2

Имя входного файла: `array2.in`
Имя выходного файла: `array2.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Постройте суффиксный массив для заданной строки s , для каждой двух соседних суффиксов найдите длину максимального общего префикса.

Формат входных данных

Первая строка входного файла содержит строку s ($1 \leq |s| \leq 400\,000$). Строка состоит из строчных латинских букв.

Формат выходных данных

В первой строке выведите $|s|$ различных чисел — номера первых символов суффиксов строки s так, чтобы соответствующие суффиксы были упорядочены в лексикографически возрастающем порядке. Во второй строке выведите $|s| - 1$ чисел — длины наибольших общих префиксов.

Примеры

array2.in	array2.out
ababb	1 3 5 2 4 2 0 1 1

Задача Е. Рефрен

Имя входного файла: `refrain.in`
Имя выходного файла: `refrain.out`
Ограничение по времени: 3 секунды
Ограничение по памяти: 64 мегабайта

Рассмотрим последовательность n целых чисел от 1 до m . Подпоследовательность подряд идущих чисел называется *рефреном*, если произведение ее длины на количество вхождений в последовательность максимально.

По заданной последовательности требуется найти ее рефрен.

Формат входных данных

Первая строка входного файла содержит два целых числа: n и m ($1 \leq n \leq 150\,000$, $1 \leq m \leq 10$).

Вторая строка содержит n целых чисел от 1 до m .

Формат выходных данных

Первая строка выходного файла должна содержать произведение длины рефрена на количество ее вхождений. Вторая строка должна содержать длину рефрена. Третья строка должна содержать последовательность которая является рефреном.

Пример

<code>refrain.in</code>	<code>refrain.out</code>
9 3	9
1 2 1 2 1 3 1 2 1	3
	1 2 1

Задача F. Количество подстрок

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

Вычислите количество различных подстрок строки s .

Формат входных данных

Первая строка входного файла содержит строку s ($1 \leq |s| \leq 400\,000$). Строка состоит из строчных латинских букв.

Формат выходных данных

Выведите одно число — ответ на задачу.

Примеры

<code>count.in</code>	<code>count.out</code>
ababb	11

Задача G. Наибольшая общая подстрока

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

Найдите наибольшую общую подстроку строк s и t .

Формат входных данных

Первая строка входного файла содержит строку s , вторая — t ($1 \leq |s|, |t| \leq 100,000$). Строки состоят из строчных латинских букв.

Формат выходных данных

Выведите одну строку — наибольшую общую подстроку строк s и t . В случае, если ответ не единственный, выведите минимальный лексикографически.

Примеры

<code>common.in</code>	<code>common.out</code>
ababb abacabba	aba

Задача Н. Название теоремы

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

Пришла октябрьская пора новых теорем и вечных двигателей. Очередной такой теореме нужно придумать название, и, естественно, ответственным за это является Евгений.

Авторов теоремы n человек, которые по очереди, начиная с 1-го и заканчивая n -м, подходят к Евгению и говорят, какую букву нужно дописать в конец к текущему названию. В самом начале, до того, как кто-нибудь из авторов подойдет, название является пустой строкой.

Некоторые авторы так и не определились какую букву они хотят добавить, поэтому они предлагают Евгению выбрать её самому. Чтобы легче было запомнить название теоремы, необходимо, чтобы после добавления выбранной Евгением буквы количество различных подстрок строки текущего названия было минимально. Если таких букв несколько, то нужно выбрать ту, которая лексикографически меньше.

Если строка $S_k = s_1s_2 \dots s_{k-1}s_k$ — текущее название, то ее подстрокой является любая строка $S_{ij} = s_is_{i+1} \dots s_{j-1}s_j$, такая что $1 \leq i \leq j \leq k$. Две подстроки являются различными, если у них различны длины или они отличаются хотя бы в одном символе.

Помогите Евгению составить название новой теоремы.

Формат входных данных

Входные данные в первой строке содержат одно целое число n — количество авторов теоремы. В следующих n строках дано по одному символу c_i — строчная буква латинского алфавита, если i -й автор определился какую букву, он хочет поставить, и знак вопроса, если нет ($1 \leq n \leq 10^5$). Гарантируется, что во входных данных присутствует хотя бы один знак вопроса.

Формат выходных данных

Для каждого знака вопроса выведите букву, которую необходимо приписать к текущему названию, чтобы количество различных подстрок названия после добавления этой буквы было минимально, а так же через пробел само число различных подстрок.

Примеры

стандартный ввод	стандартный вывод
8 l a g r ? n g e	a 14
4 x ? d ?	x 2 d 8
6 a b c a b ?	c 15
3 ? ? ?	a 1 a 2 a 3

Задача I. Ненокку

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

Очень известный автор не менее известной книги решил написать продолжение своего произведения. Он писал все свои книги на компьютере, подключенном к интернету. Из-за такой неосторожности мальчику Ненокку удалось получить доступ к еще ненаписанной книге. Каждый вечер мальчик залазил на компьютер писателя и записывал на свой компьютер новые записи. Ненокку, записав на свой компьютер очередную главу, заинтересовался, а использовал ли хоть раз писатель слово “книга”. Но он не любит читать книги (он лучше ползает в интернете), и поэтому он просит вас узнать есть ли то или иное слово в тексте произведения. Но естественно его интересует не только одно слово, а достаточно много.

Формат входных данных

В каждой строчке входного файла записано одна из двух записей.

1. ? <слово> (<слово> - это набор не более 50 латинских символов);
2. A <текст> (<текст> - это набор не более 10^5 латинских символов).

1 означает просьбу проверить существование подстроки <слово> в произведение.

2 означает добавление в произведение <текст>.

Писатель только начал работать над произведением, поэтому он не мог написать более 10^5 символов. А входной файл содержит не более 15 мегабайт информации.

Формат выходных данных

Выведите на каждую строчку типа 1 “YES”, если существует подстрока <слово>, и “NO” в противном случае. Не следует различать регистр букв.

Пример

стандартный ввод	стандартный вывод
? love	NO
? is	NO
A Loveis	YES
? love	NO
? WHO	YES
A Whoareyou	
? is	

Задача J. Помогите, спасите!

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Дана строка. Найдите для каждого её префикса количество различных подстрок в нём.

Формат входных данных

В единственной строке входных данных содержится непустая строка S , состоящая из N ($1 \leq N \leq 2 \cdot 10^5$) маленьких букв английского алфавита.

Формат выходных данных

Выведите N строк, в i -й строке должно содержаться количество различных подстрок в i -м префиксе строки S .

Примеры

стандартный ввод	стандартный вывод
aabab	1 2 5 8 11
atari	1 3 5 9 14