

Задача А. Z-функция

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

Дана непустая строка S , длина которой N не превышает 10^6 . Будем считать, что элементы строки нумеруются от 1 до N .

Требуется для всех i от 1 до N вычислить её z-функцию $z[i]$.

Формат входных данных

Одна строка длины N , $0 < N \leq 10^6$, состоящая из маленьких латинских букв.

Формат выходных данных

Выведите N чисел — значения z-функции для каждой позиции, разделённые пробелом.

Примеры

<code>z-function.in</code>	<code>z-function.out</code>
<code>abracadabra</code>	<code>11 0 0 1 0 1 0 4 0 0 1</code>

Задача В. Префикс-функция

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

Дана непустая строка S , длина которой N не превышает 10^6 . Будем считать, что элементы строки нумеруются от 1 до N .

Требуется для всех i от 1 до N вычислить её префикс-функцию $\pi[i]$.

Формат входных данных

Одна строка длины N , $0 < N \leq 10^6$, состоящая из маленьких латинских букв.

Формат выходных данных

Выведите N чисел — значения префикс-функции для каждой позиции, разделённые пробелом.

Примеры

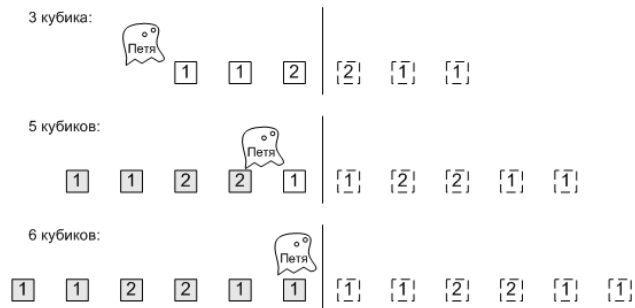
<code>prefix-function.in</code>	<code>prefix-function.out</code>
<code>abracadabra</code>	<code>0 0 0 1 0 1 0 1 2 3 4</code>

Задача С. Кубики

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

Привидение Петя любит играть со своими кубиками. Он любит выкладывать их в ряд и разглядывать своё творение. Однако недавно друзья решили подшутить над Петей и поставили в его игровой комнате зеркало. Ведь всем известно, что привидения не отражаются в зеркале! А кубики отражаются.

Теперь Петя видит перед собой N цветных кубиков, но не знает, какие из этих кубиков настоящие, а какие — всего лишь отражение в зеркале.



Помогите Пете! Выясните, сколько у него может быть кубиков. Петя видит отражение всех кубиков в зеркале и часть кубиков, которая находится перед ним. Часть кубиков может быть позади Пети, их он не видит.

Формат входных данных

Первая строка входного файла содержит два целых числа: N ($1 \leq N \leq 100\,000$) и количество различных цветов, в которые могут быть раскрашены кубики, — M ($1 \leq M \leq 100\,000$). Следующая строка содержит N целых чисел от 1 до M — цвета кубиков.

Формат выходных данных

В выходной файл выведите в порядке возрастания все такие K , что у Пети может быть K кубиков.

Примеры

<code>cubes.in</code>	<code>cubes.out</code>
6 2 1 1 2 2 1 1	3 5 6

Задача D. Неточное совпадение

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

Даны строки p и t . Требуется найти все вхождения строки p в строку t в качестве подстроки с точностью до возможного несовпадения одного символа.

Формат входных данных

Первая строка входного файла содержит p , вторая — t ($1 \leq |p|, |t| \leq 10^6$). Строки состоят из букв латинского алфавита.

Формат выходных данных

В первой строке выведите количество вхождений строки p в строку t . Во второй строке выведите в возрастающем порядке номера символов строки t , с которых начинаются вхождения p . Символы нумеруются с единицы.

Примеры

<code>inexact-matching.in</code>	<code>inexact-matching.out</code>
<code>aaaa</code>	<code>4</code>
<code>Caabdaaaa</code>	<code>1 2 6 7</code>

Задача Е. Подстроки

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

Дано K строк из маленьких латинских букв. Требуется найти их наибольшую общую подстроку.

Формат входных данных

В первой строке число K ($1 \leq K \leq 10$). В следующих K строках — собственно K строк (длины строк от 1 до 10 000).

Формат выходных данных

Наибольшая общая подстрока.

Примеры

<code>substr.in</code>	<code>substr.out</code>
3 abacaba mycabarchive acabistrue	cab

Задача F. «Abracadabra»

Имя входного файла:	<code>suprefixes.in</code>
Имя выходного файла:	<code>suprefixes.out</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Строка s называется *супрефиксом* для строки t , если t начинается с s и заканчивается на s . Например, «*abra*» является супрефиксом для строки «*abracadabra*». В частности, сама строка t является своим супрефиксом. Супрефиксы играют важную роль в различных алгоритмах на строках.

В этой задаче требуется решить обратную задачу о поиске супрефикса, которая заключается в следующем. Задан словарь, содержащий n слов t_1, t_2, \dots, t_n и набор из m строк-образцов s_1, s_2, \dots, s_m . Необходимо для каждой строки-образца из заданного набора найти количество слов в словаре, для которых эта строка-образец является супрефиксом.

Требуется написать программу, которая по заданному числу n , n словам словаря t_1, t_2, \dots, t_n , заданному числу m и m строкам-образцам s_1, s_2, \dots, s_m вычислит для каждой строки-образца количество слов из словаря, для которых эта строка-образец является супрефиксом.

Формат входных данных

Первая строка входного файла содержит целое число n ($1 \leq n \leq 200'000$).

Последующие n строк содержат слова t_1, t_2, \dots, t_n , по одному слову в каждой строке. Каждое слово состоит из строчных букв латинского алфавита. Длина каждого слова не превышает 50. Суммарная длина всех слов не превышает 10^6 . Словарь не содержит пустых слов.

Затем следует строка, содержащая целое число m ($1 \leq m \leq 200'000$).

Последующие m строк содержат строки-образцы s_1, s_2, \dots, s_m , по одной на каждой строке. Каждая строка-образец состоит из строчных букв латинского алфавита: Длина каждой строки-образца не превышает 50. Суммарная длина всех строк-образцов не превышает 10^6 . Никакая строка-образец не является пустой строкой.

Формат выходных данных

Выходной файл должен содержать m чисел, по одному на строке.

Для каждой строки-образца в порядке, в котором они заданы во входном файле, следует вывести количество слов словаря, для которых она является супрефиксом.

Примеры

<code>suprefixes.in</code>	<code>suprefixes.out</code>
4	4
abacaba	2
abracadabra	0
aa	
abra	
3	
a	
abra	
abac	

Задача G. Доминошки

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

Вам дано клетчатое поле размера $n \times m$ клеток. Некоторые клетки на этом поле отмечены как занятые. На нем необходимо разместить несколько (возможно, 0) доминошек размера 1×2 , соблюдая следующие ограничения:

- каждая клетка может быть покрыта не более чем одной доминошкой
- никакие две соседних по стороне клетки не могут быть заняты разными доминошками

Сколькими способами можно это сделать?

Формат входных данных

В первой строке входного файла содержится два целых числа: n и m ($1 \leq n \leq 6, 1 \leq m \leq 100$). В следующих n строках содержится описание поля. Символ `X` означает занятую клетку, символ `.` означает свободную клетку.

Формат выходных данных

Выведите одно целое число — количество способов разместить доминошки, взятое по модулю $10^9 + 7$.

Примеры

<code>dominoes.in</code>	<code>dominoes.out</code>
<pre>2 2</pre>	5
<pre>3 3X. ...</pre>	21
<pre>5 5X..</pre>	5182