

Задача А. 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 В. А-функция

Имя входного файла: `afunction.in`
Имя выходного файла: `afunction.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Дана строка S , состоящая из N символов. Определим функцию $A(i)$ от первых i символов этой строки следующим образом:

$A(i)$ = максимально возможному k такому, что равны следующие две строки: $S[1] + S[2] + S[3] + \dots + S[k] = S[i] + S[i - 1] + S[i - 2] + \dots + S[i - k + 1]$, где $S[i]$ — i -ый символ строки S , а знак $+$ означает, что символы записываются в строчку непосредственно друг за другом.

Напишите программу, которая вычислит значения функции A для заданной строки для всех возможных значений i от 1 до N .

Формат входных данных

В первой строке входного файла записано одно число N . $1 \leq N \leq 200\,000$. Во второй строке записана строка длиной N символов, состоящая только из больших и/или маленьких латинских букв.

Формат выходных данных

В выходной файл выведите N чисел — значения функции $A(1), A(2), \dots, A(N)$.

Примеры

<code>afunction.in</code>	<code>afunction.out</code>
5 aabaa	1 2 0 1 5

Задача С. Циклическая строка

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

Строка S была записана много раз подряд, после чего из получившейся строки взяли подстроку и дали вам. Ваша задача определить минимально возможную длину исходной строки S .

Формат входных данных

На вход программы поступает строка, которая содержит только латинские буквы, длина строки не превышает 50 000 символов.

Формат выходных данных

Требуется вывести одно число — ответ на вопрос задачи.

Примеры

стандартный ввод	стандартный вывод
z	1
сac	2

Задача D. Последнее слово Джека

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

Джек недавно прочитал на заборе интересное и новое для него слово. Оно настолько понравилось Джеку, что он захотел сам придумать ещё какое-нибудь интересное слово. Но только ничего у него не вышло — все придуманные им слова состояли из префиксов исходного слова и поэтому не приносили радости. Он стал придумывать всё более и более длинные слова, но ни одно из них не было оригинальным. . .

И вот настало время Джеку сказать своё последнее слово.

Формат входных данных

Первая строка содержит интересное слово, которое было написано на заборе. Вторая строка содержит последнее слово Джека. Длины слов не превосходят 75 000, слова непустые и состоят из строчных латинских букв.

Формат выходных данных

Если Джек так ничего и не придумал своего, выведите первой строкой `No`. В этом случае покажите Джеку, как разбить его последнее слово на несколько частей, каждая из которых является исходным словом или его непустым префиксом — выведите все эти части во второй строке, разделяя их пробелом. Если же такого разбиения нет, и последнее слово было за Джеком, выведите единственной строкой `Yes`.

Примеры

<code>prefix.in</code>	<code>prefix.out</code>
<code>abracadabra abrabracada</code>	<code>No abr abracada</code>
<code>abracadabra arbadacarba</code>	<code>Yes</code>

Задача E. Максимальная общая подстрока

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Дана строка $A = a_1a_2\dots a_n$ и строка $B = b_1b_2\dots b_m$. Нужно узнать длину их максимальной общей подстроки.

Формат входных данных

В первых двух строках записаны строки A и B , состоящие из строчных латинских букв. Эти строки непустые и имеют длину не более 30 000 символов.

Формат выходных данных

В выходной файл выведите длину их максимальной общей подстроки.

Примеры

стандартный ввод	стандартный вывод
abacaba acabaca	5

Задача F. Сравнение подстрок

Имя входного файла: `substrcmp.in`
Имя выходного файла: `substrcmp.out`
Ограничение по времени: 1.5 секунд
Ограничение по памяти: 256 мегабайт

Дана строка $S = s_1 s_2 \dots s_n$ и множество запросов вида (l_1, r_1, l_2, r_2) . Для каждого запроса требуется ответить, какая из подстрок больше — $s_{l_1} \dots s_{r_1}$ или $s_{l_2} \dots s_{r_2}$.

Формат входных данных

В первой строке записана строка S , состоящая из строчных латинских букв. Эта строка непустая и имеет длину не более 100 000 символов. Во второй строке записано целое число q ($1 \leq q \leq 100\,000$) — количество запросов. В каждой из следующих q строк записаны числа l_1, r_1, l_2, r_2 ($1 \leq l_1 \leq r_1 \leq |S|$; $1 \leq l_2 \leq r_2 \leq |S|$).

Формат выходных данных

Для каждого запроса выведите «=», если соответствующие подстроки равны, «>», если первая подстрока больше и «<», если первая подстрока меньше.

Примеры

substrcmp.in	substrcmp.out
abacaba	=
3	<
1 3 5 7	>
1 3 3 5	
4 7 2 5	
ab	<
2	<
1 1 2 2	
1 1 1 2	

Задача G. Палиндромы

Имя входного файла: `palindrome.in`
Имя выходного файла: `palindrome.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Строка называется палиндромом, если она одинаково читается как слева направо, так и справа налево. Например, `abba` — палиндром, а `омах` — нет. Для строки α будем обозначать $\alpha[i..j]$ ее подстроку длины $j - i + 1$ с i -й по j -ю позицию включительно (позиции нумеруются с единицу). Для заданной строки α длины N ($1 \leq N \leq 100\,000$) требуется подсчитать число q пар (i, j) , $1 \leq i < j \leq n$, таких что $\alpha[i..j]$ является палиндромом.

Формат входных данных

Входной файл содержит одну строку α длины N , состоящую из маленьких латинских букв.

Формат выходных данных

В выходной файл выведите искомое число q .

Примеры

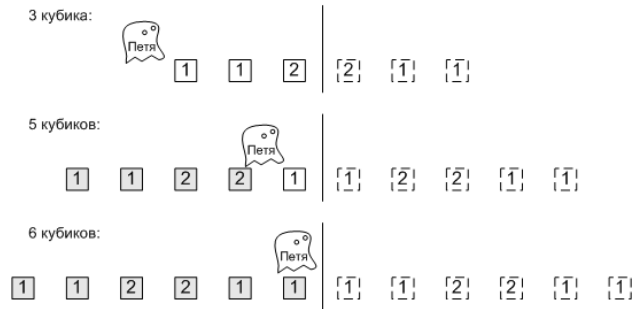
<code>palindrome.in</code>	<code>palindrome.out</code>
<code>aaa</code>	3
<code>abba</code>	2
<code>омах</code>	0

Задача Н. 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 I. 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 J. Подпалиндромы

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

Дано слово и запросы двух типов:

- заменить i -ю букву в слове на букву c ;
- проверить, является ли подстрока $s_j \dots s_k$ палиндромом.

Формат входных данных

В первой строке записано слово из n строчных латинских букв. Во второй строке записано целое число m — количество запросов ($5 \leq n, m \leq 10^5$). Следующие m строк содержат запросы. Каждый запрос имеет вид «change i a » или «palindrome? j k », где i, j, k — целые числа ($1 \leq i \leq n; 1 \leq j \leq k \leq n$), а символ c — строчная латинская буква.

Формат выходных данных

На все запросы второго типа выведите «Yes», если подслово $s_j \dots s_k$ является палиндромом, и «No» в противном случае.

Примеры

<code>substring-palindromes.in</code>	<code>substring-palindromes.out</code>
<code>abcda</code>	<code>No</code>
<code>5</code>	<code>Yes</code>
<code>palindrome? 1 5</code>	<code>Yes</code>
<code>palindrome? 1 1</code>	<code>Yes</code>
<code>change 4 b</code>	
<code>palindrome? 1 5</code>	
<code>palindrome? 2 4</code>	

Задача К. Движущаяся строка

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Поликарп нашёл строку s и перестановку p . Их длины оказались одинаковы и равны n .

Перестановка из n элементов — это массив длины n , в котором каждое целое число от 1 до n встречается ровно по одному разу. Например, $[1, 2, 3]$ и $[4, 3, 5, 1, 2]$ — это перестановки, но $[1, 2, 4]$, $[4, 3, 2, 1, 2]$ и $[0, 1, 2]$ — это не перестановки.

За одну операцию он может умножить s на p , то есть заменить строку s на строку new , в которой для каждого i от 1 до n верно, что $new_i = s_{p_i}$. Например, при $s = wmbc$ и $p = [3, 1, 4, 2]$, после применения операции строка превратится в $s = s_3s_1s_4s_2 = bwcm$.

Поликарпу стало интересно, через сколько операций строка впервые вернётся к своему первоначальному виду. Так как это может занять слишком много времени, он просит вашей помощи в этом вопросе.

Можно доказать, что искомое количество операций всегда существует. Оно может оказаться очень большим, используйте 64-битный целочисленный тип.

Формат входных данных

В первой строке входных данных записано целое число t ($1 \leq t \leq 5000$) — количество наборов входных данных в тесте.

Первая строка каждого набора содержит целое число n ($1 \leq n \leq 200$) — длину строки и перестановки.

Вторая строка каждого набора содержит строку s длины n , состоящую из строчных латинских букв.

Третья строка каждого набора содержит n целых чисел — перестановку p ($1 \leq p_i \leq n$), все p_i различны.

Формат выходных данных

Выведите t строк, каждая из которых содержит ответ на соответствующий набор входных данных. В качестве ответа выведите единственное число — минимальное количество операций, после которого строка s станет такой же, какой была до их применения.

Пример

стандартный ввод	стандартный вывод
3	1
5	6
ababa	12
3 4 5 2 1	
5	
ababa	
2 1 4 5 3	
10	
codeforces	
8 6 1 7 5 2 9 3 10 4	

Замечание

В первом наборе входных данных применение операции не изменяет строку, поэтому она станет равной самой себе после 1 операции.

Во втором наборе входных данных строка будет меняться следующим образом:

- $s = babaa$
- $s = abaab$

- $s = baaba$
- $s = abbaa$
- $s = baaab$
- $s = ababa$