

Задача А. 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>
aaaa	4
Caabdaaaaa	1 2 6 7

Задача В. Палиндромы

Имя входного файла: `palindrome.in`
Имя выходного файла: `palindrome.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

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 С. Дана строка

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

Даже больше — дано две строки, α и β . Вам требуется узнать, где в строке α можно найти строку β как подстроку и выписать все такие позиции.

Формат входных данных

В первой строке входного файла содержится строка α , во второй — строка β . Строки состоят только из строчных латинских букв (`a–z`), их длины не превосходят 100 000.

Формат выходных данных

В первой строке выходного файла выведите одно число — количество вхождений строки β в строку α . Во второй строке для каждого вхождения выведите номер символа в строке α , где начинается очередная строка β . Вхождения нужно выводить в возрастающем порядке.

Примеры

<code>search.in</code>	<code>search.out</code>
abacaba	2
aba	1 5

Задача D. Строка

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

Дед Мороз составил важное сообщение, которое нужно передать другим Дедам Морозам. На всякий случай он зашифровал его с помощью шифра замены: заменил все буквы в сообщении, одинаковые буквы на одинаковые, разные на разные. На всякий случай Дед Мороз зашифровал свое сообщение два раза.

Потом он ненадолго отвлекся, а в это время один из эльфов прибежал в дом и решил пошалить. Он взял и приписал к шифровкам спереди и сзади какую-то абракадабру.

Теперь Деду Морозу надо найти наибольшую подстроку, которая могла бы быть его шифровкой. Помогите ему.

Например, пусть сообщение Деда Мороза имело вид `'xxz'`. Он его зашифровал первый раз как `'bba'`, второй раз — как `'aab'`. После этого эльф приписал к первому сообщению спереди букву `a`, сзади — ничего; ко второму — спереди ничего, сзади `b`. В итоге получились строки `'abba'` и `'aabb'`.

Формат входных данных

Входной файл содержит две строки, состоящие из символов с кодами от 33 до 127 и длиной не более 5000. Гарантируется, что обе строки во входном файле непустые.

Формат выходных данных

Выведите две строки — предполагаемые шифровки, в том виде, в котором они были в первой и во второй исходных строках.

Примеры

<code>substring.in</code>	<code>substring.out</code>
<code>abba</code> <code>aabb</code>	<code>bba</code> <code>aab</code>
<code>abab</code> <code>baba</code>	<code>abab</code> <code>baba</code>
<code>k555kk5555</code> <code>5kkk5kk5kk</code>	<code>k555k</code> <code>5kkk5</code>