

Задача А. Строки

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

Вася посещает занятия по программированию. К сожалению, недавно он заболел и не смог прийти на лекцию. Он смог узнать лишь, что на лекции проходили поиск подстроки в строке. Вася боится, что на следующей тренировке ему придётся искать подстроку в строке, а он до сих пор не знает, как это делается. Помогите ему!

Формат входных данных

В первых двух строчках входного файла даны две строки \mathcal{T} и \mathcal{S} ($1 \leq |\mathcal{T}| \leq 1\,000\,000$, $1 \leq |\mathcal{S}| \leq 1\,000\,000$). Строки состоят только из строчных букв 'a'–'z' латинского алфавита.

Формат выходных данных

Выведите через пробел все начала вхождений строки \mathcal{S} в строку \mathcal{T} в порядке возрастания. Вхождение начинается в позиции k , если $\mathcal{T}_k = \mathcal{S}_1, \mathcal{T}_{k+1} = \mathcal{S}_2, \dots, \mathcal{T}_{k+|\mathcal{S}|-1} = \mathcal{S}_{|\mathcal{S}|}$. Если вхождений нет, выведите единственное слово "none" вместо списка вхождений.

Примеры

strings.in	strings.out
ababcabc abc	3
qwerty asdfgh	none

Задача В. Различные подстроки

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

Подстрокой строки $\mathcal{S} = s_1s_2 \dots s_n$ называется непрерывная подпоследовательность символов этой строки $s_i s_{i+1} s_{i+2} \dots s_{j-1} s_j$.

Дана строка. Сколько различных подстрок, не считая пустой, она содержит?

Формат входных данных

В первой строке ввода задана строка длины от 1 до 100 символов, включительно. Строка состоит из маленьких букв английского алфавита.

Формат выходных данных

В первой строке выведите одно число — количество различных подстрок данной строки, не считая пустой.

Примеры

unequal.in	unequal.out
aab	5
dabux	15

Задача С. Различные подстроки 2

Имя входного файла: `unequal2.in`
Имя выходного файла: `unequal2.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Подстрокой строки $S = s_1s_2 \dots s_n$ называется непрерывная подпоследовательность символов этой строки $s_i s_{i+1} s_{i+2} \dots s_{j-1} s_j$.

Дана строка. Сколько различных подстрок, не считая пустой, она содержит? Выведите все эти подстроки по одному разу в лексикографическом порядке.

Формат входных данных

В первой строке ввода задана строка длины от 1 до 100 символов, включительно. Строка состоит из маленьких букв английского алфавита.

Формат выходных данных

В первой строке выведите одно число r — количество различных подстрок данной строки, не считая пустой. В следующих r строках выведите сами эти подстроки в лексикографическом порядке.

Примеры

<code>unequal2.in</code>	<code>unequal2.out</code>
aab	5 a aa aab ab b
da	3 a d da

Задача D. Различные подстроки 3

Имя входного файла: `unequal3.in`
Имя выходного файла: `unequal3.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Подстрокой строки $S = s_1s_2 \dots s_n$ называется непрерывная подпоследовательность символов этой строки $s_i s_{i+1} s_{i+2} \dots s_{j-1} s_j$.

Дана строка. Сколько различных подстрок, не считая пустой, она содержит? Выведите все эти подстроки по одному разу в лексикографическом порядке. Рядом с каждой подстрокой выведите, сколько раз она встречается в строке S .

Формат входных данных

В первой строке ввода задана строка длины от 1 до 100 символов, включительно. Строка состоит из маленьких букв английского алфавита.

Формат выходных данных

В первой строке выведите одно число r — количество различных подстрок данной строки, не считая пустой. В следующих r строках выведите сами эти подстроки в лексикографическом порядке. После каждой подстроки через пробел должно следовать целое число — сколько раз эта подстрока встречается в строке S .

Примеры

<code>unequal3.in</code>	<code>unequal3.out</code>
aab	5 a 2 aa 1 aab 1 ab 1 b 1
da	3 a 1 d 1 da 1

Задача Е. Двухкратная подстрока

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

Дана строка S длины n и число k . Найдите в строке S такую подстроку длины k , которая встречается в ней по крайней мере два раза, или выясните, что такой подстроки нет.

Формат входных данных

В первой строке входного файла задана строка S ; её длина n — от 1 до 100 000 символов, включительно. Во второй строке задано целое число k ($1 \leq k \leq n$). Строка состоит только из маленьких букв латинского алфавита.

Формат выходных данных

Если подстроки длины k , встречающейся хотя бы два раза, не существует, выведите слово «NONE» в первой строке выходного файла. В противном случае выведите любую из таких подстрок в первой строке выходного файла.

Примеры

double.in	double.out
ast 1	NONE
blinkingblueblogger 2	in
aaaaaab 5	aaaaa

Задача F. Ретрострока

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

Строкой S называется последовательность символов $S_1S_2\dots S_n$, где $|S| = n$ — это длина строки S .

Для любого k ($1 \leq k \leq |S|$) k -м префиксом строки S называется строка $S_1S_2\dots S_k$ длины k . Если $k < |S|$, то префикс называется собственным.

Аналогично для любого k ($1 \leq k \leq |S|$) k -м суффиксом строки S называется строка $S_{|S|-k+1}S_{|S|-k+2}\dots S_{|S|}$ длины k . Если $k < |S|$, то суффикс также называется собственным.

Назовём числом повторяемости строки S количество её различных собственных суффиксов, каждый из которых совпадает с префиксом той же длины, что и этот суффикс.

Назовём строку ретрострокой, если её число повторяемости строго больше чисел повторяемости всех её собственных префиксов.

Дана строка S . Нужно найти её префикс максимальной длины (не обязательно собственный), являющийся ретрострокой.

Формат входных данных

В первой строке входного файла записана строка S , $1 \leq |S| \leq 1000000$. Строка содержит лишь символы с ASCII-кодом от 33 до 126.

Формат выходных данных

В первой строке выходного файла должен быть выведен префикс S максимальной длины, являющийся ретрострокой.

Примеры

retro.in	retro.out
z	z
aabaabaabaabaabaaba	aabaabaabaaba

Задача G. Суффиксы

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

Назовём *строкой* последовательность из маленьких букв английского алфавита. Строкой, например, является пустая последовательность «», слово «aabaf» или бесконечная последовательность букв «a».

Определим *i*-й суффикс S_i строки S : это просто строка S , из которой вырезаны первые i букв. Так, для строки $S = \text{«aabaf»}$ суффиксы будут такими:

$$\begin{aligned} S_0 &= \text{«aabaf»} \\ S_1 &= \text{«abaf»} \\ S_2 &= \text{«baf»} \\ S_3 &= \text{«af»} \\ S_4 &= \text{«f»} \\ S_5 = S_6 = S_7 = \dots &= \text{«»} \end{aligned}$$

Суффиксы определены для всех $i \geq 0$.

Циклическое расширение S^* конечной строки S — это строка, полученная приписыванием её к самой себе бесконечное количество раз. Так,

$$\begin{aligned} S^* = S_0^* &= \text{«aabafaabafaa...»} \\ S_1^* &= \text{«abafabafabaf...»} \\ S_2^* &= \text{«bafbafbafbaf...»} \\ S_3^* &= \text{«afafafafafaf...»} \\ S_4^* &= \text{«ffffffffffffff...»} \\ S_5^* = S_6^* = S_7^* = \dots &= \text{«»} \end{aligned}$$

По данной строке S выясните, сколько её суффиксов S_i имеют такое же циклическое расширение, как и сама строка S , то есть количество таких i , что $S^* = S_i^*$.

Формат входных данных

В первой и единственной строке входного файла задана строка S , состоящая из не менее, чем одной, и не более, чем 100 000 маленьких латинских букв «a–z».

Формат выходных данных

Выведите в первую строку выходного файла одно число — количество суффиксов строки S , имеющих такое же циклическое расширение, как и она сама.

Примеры

<code>suffixes.in</code>	<code>suffixes.out</code>
aa	2
ab	1
qqqq	4
хyzzухy	1

Задача H. Генератор строки

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

Генератором строки назовём наименьший префикс, который нужно повторить (возможно, нецелое число раз), чтобы получить эту строку. Например, генератор строки «ababab» — это строка «ab», генератор строки «abcabca» — это строка «abc», генератором строки «abcd» является она сама.

По данной строке найдите её генератор.

Формат входных данных

В первой строчке входного файла задана непустая строка, состоящая из строчных букв латинского алфавита. Длина строки не превышает одного миллиона символов.

Формат выходных данных

В первой строчке выходного файла выведите генератор данной строки.

Примеры

<code>genstr.in</code>	<code>genstr.out</code>
ababab	ab
abcabca	abc
abcd	abcd

Задача I. От префикс-функции к z-функции

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

Префикс-функция $p(i)$ для строки $s = s_1s_2\dots s_n$ определяется от позиции i ($1 \leq i \leq n$) в строке так: $p(i)$ — это максимальная длина собственного префикса строки $s_1s_2\dots s_i$, равного её собственному суффиксу. Напомним, что *собственный префикс* строки $s = s_1s_2\dots s_n$ — это строка $s_1s_2\dots s_r$ для некоторого $r < n$. Аналогично, *собственный суффикс* строки $s = s_1s_2\dots s_n$ — это строка $s_ls_2\dots s_n$ для некоторого $l > 1$.

Z-функция $z(i)$ для строки $s = s_1s_2\dots s_n$ определяется от позиции i ($1 \leq i \leq n$) в строке так: $z(1) = 0$, а для $i > 1$ $z(i)$ — это максимальное число такое, что строки $s_1s_2\dots s_{z(i)}$ и $s_is_{i+1}\dots s_{i+z(i)-1}$ совпадают.

Даны длина строки n и значения префикс-функции $p(1), p(2), \dots, p(n)$ для этой строки. Найдите для этой строки значения z-функции $z(1), z(2), \dots, z(n)$.

Формат входных данных

В первой строчке задано целое число n ($1 \leq n \leq 1000000$). Во второй строчке заданы n чисел через пробел — значения префикс-функции $p(1), p(2), \dots, p(n)$. Гарантируется, что существует строка длины n , состоящая из строчных букв латинского алфавита, для которой префикс-функция от позиций 1, 2, ..., n принимает данные значения.

Формат выходных данных

В первой строчке выведите n чисел через пробел — значения z-функции для строки, имеющей данную префикс-функцию.

Примеры

ptoz.in	ptoz.out
6 0 0 1 2 3 4	0 0 4 0 2 0
7 0 0 0 1 2 3 4	0 0 0 4 0 0 1
4 0 0 0 0	0 0 0 0

Задача J. От z-функции к префикс-функции

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

Z-функция $z(i)$ для строки $s = s_1s_2\dots s_n$ определяется от позиции i ($1 \leq i \leq n$) в строке так: $z(1) = 0$, а для $i > 1$ $z(i)$ — это максимальное число такое, что строки $s_1s_2\dots s_{z(i)}$ и $s_is_{i+1}\dots s_{i+z(i)-1}$ совпадают.

Префикс-функция $p(i)$ для строки $s = s_1s_2\dots s_n$ определяется от позиции i ($1 \leq i \leq n$) в строке так: $p(i)$ — это максимальная длина собственного префикса строки $s_1s_2\dots s_i$, равного её собственному суффиксу. Напомним, что *собственный префикс* строки $s = s_1s_2\dots s_n$ — это строка $s_1s_2\dots s_r$ для некоторого $r < n$. Аналогично, *собственный суффикс* строки $s = s_1s_2\dots s_n$ — это строка $s_ls_2\dots s_n$ для некоторого $l > 1$.

Даны длина строки n и значения z-функции $z(1), z(2), \dots, z(n)$ для этой строки. Найдите для этой строки значения префикс-функции $p(1), p(2), \dots, p(n)$.

Формат входных данных

В первой строчке задано целое число n ($1 \leq n \leq 1000000$). Во второй строчке заданы n чисел через пробел — значения z-функции $z(1), z(2), \dots, z(n)$. Гарантируется, что существует строка длины n , состоящая из строчных букв латинского алфавита, для которой z-функция от позиций 1, 2, ..., n принимает данные значения.

Формат выходных данных

В первой строчке выведите n чисел через пробел — значения префикс-функции для строки, имеющей данную z-функцию.

Примеры

ztop.in	ztop.out
6 0 0 4 0 2 0	0 0 1 2 3 4
7 0 0 0 4 0 0 1	0 0 0 1 2 3 4
4 0 0 0 0	0 0 0 0