

Задача А. 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 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 английского алфавита.

Формат выходных данных

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

Примеры

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

Задача В. 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 строке задана непустая строка, состоящая из маленьких букв английского алфавита. Длина строки не превышает одного миллиона символов.

Формат выходных данных

В первой строке выведите генератор данной строки.

Примеры

genstr.in	genstr.out
ababab	ab
abcabca	abc
abcd	abcd

Задача С. Палиндромы

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

Строка называется палиндромом, если она одинаково читается как слева направо, так и справа налево. Например, abba — палиндром, а omax — нет. Для строки S будем обозначать $S[i..j]$ её подстроку длины $j - i + 1$ с i -й по j -ю позицию включительно (позиции нумеруются с единицы).

Для заданной строки S длины N требуется найти количество таких пар (i, j) , что $1 \leq i < j \leq n$ и подстрока $S[i..j]$ является палиндромом.

Формат входных данных

Ввод содержит одну строку S длины N , состоящую из маленьких английских букв ($1 \leq N \leq 100\,000$).

Формат выходных данных

Выведите искомое количество пар.

Примеры

palindrome.in	palindrome.out
aaa	3
abba	2
omax	0

Задача D. Разбиение на палиндромы

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

Дана строка, состоящая из символов с ASCII-кодами от 32 до 126, включительно. Можно ли разбить эту строку на палиндромы чётной длины?

Формат входных данных

Первая строка ввода содержит число n — количество тестов ($1 \leq n \leq 100$). Следующие n строк содержат сами тесты. Длина каждой строки не превышает 10^6 символов. Общий объём ввода не превышает трёх мегабайт.

Формат выходных данных

Выведите ответ на каждый тест на отдельной строке. Выводите «YES», если строку можно разбить на палиндромы чётной длины, и «NO» в противном случае.

Пример

palsplit.in	palsplit.out
3	NO
madam	NO
aA	YES
aabb	

Задача E. Период строки

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

Строка S имеет период T , если

$$\exists n > 0 : S = T^n = \underbrace{TT \dots T}_n.$$

Вам дана строка S . Ваша задача — найти минимальную по длине T такую, что $S = T^n$ для некоторого $n \in \mathbb{N}$.

Формат входных данных

Строка S длиной от 1 до 10^6 символов.

Формат выходных данных

Единственное число — длина T .

Пример

period.in	period.out
abaabaabaabaaba	3

Задача F. 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 ($1 \leq n \leq 1\,000\,000$). 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 маленьких букв английского алфавита, для которой префикс-функция от позиций $1, 2, \dots, n$ 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 Г. 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 записана строка S ($1 \leq |S| \leq 1\,000\,000$). Строка содержит лишь символы с ASCII-кодами от 33 до 126.

Формат выходных данных

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

Примеры

retro.in	retro.out
z	z
aabaabaabaabaabaaba	aabaabaabaaba

Задача Н. 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 даны две строки T и S ($1 \leq |T| \leq 1\,000\,000$, $1 \leq |S| \leq 1\,000\,000$). Строки состоят только из маленьких букв «a»–«z» английского алфавита.

Формат выходных данных

Выведите через пробел все начала вхождений строки S в строку T в порядке возрастания. Вхождение начинается в позиции k , если $T_k = S_1, T_{k+1} = S_2, \dots, T_{k+|S|-1} = S_{|S|}$. Если вхождений нет, выведите единственное слово «none» вместо списка вхождений.

Примеры

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

Задача I. Подстрока

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

Дана строка. Найдите её подстроку, которая встречается в этой строке как можно больше раз.

Формат входных данных

В первой строке записана непустая строка S длиной не более 100 000 символов, состоящая из маленьких букв английского алфавита.

Формат выходных данных

Выведите непустую подстроку P данной строки S , встречающуюся в S как можно больше раз. Если оптимальных ответов несколько, выведите любой.

Примеры

substr.in	substr.out
a	a
abc	ab

Задача J. Минимальный суффикс

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

Как известно, k -м суффиксом строки $S = s_1s_2\dots s_n$ называется строка $S_k = s_k s_{k+1} s_{k+2} \dots s_n$.

Например, для строки $S = \text{abaca}$ суффиксы будут такие: $S_1 = \text{abaca}$, $S_2 = \text{baca}$, $S_3 = \text{aca}$, $S_4 = \text{ca}$, $S_5 = \text{a}$, а все последующие суффиксы пусты.

В этой задаче требуется найти лексикографически минимальный непустой суффикс заданной строки S .

Формат входных данных

В первой строке записана строка S , состоящая только из маленьких букв английского алфавита. Длина этой строки — от 1 до 100 000 букв, включительно.

Формат выходных данных

В первой строке выведите суффикс S_k , являющийся лексикографически минимальным непустым суффиксом строки S .

Примеры

suffix.in	suffix.out
abcde	abcde
abaca	a
bcdeabc	abc

Задача K. 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:

$$S_0 = \text{«aabaf»}$$

$$S_1 = \text{«abaf»}$$

$$S_2 = \text{«baf»}$$

$$S_3 = \text{«af»}$$

$$S_4 = \text{«f»}$$

$$S_5 = S_6 = S_7 = \dots = \text{«»}$$

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 задана строка S , состоящая из не менее чем одной и не более чем 100 000 маленьких английских букв «a–z».

Формат выходных данных

Выведите одно число — количество суффиксов строки S , имеющих такое же циклическое расширение, как и она сама.

Примеры

suffixes.in	suffixes.out
aa	2
ab	1
qqqq	4
xyzzуху	1

Задача L. 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 маленьких букв английского алфавита, для которой z-функция от позиций $1, 2, \dots, n$ 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