

y2018-3-3. Алгоритмы на строках

A. Substrings

2 seconds, 256 megabytes

For a given string s and m requests to check, if substrings $[a..b]$ and $[c..d]$ are equal.

Input

The first line contains string s ($1 \leq |s| \leq 10^5$).

The second line contains a single integer m — the number of requests ($0 \leq m \leq 10^5$).

Each of the next m lines contains four integers — a, b, c, d ($1 \leq a \leq b \leq |s|, 1 \leq c \leq d \leq |s|$).

Output

For each request output "Yes", if corresponding strings are equal, and "No" — otherwise.

input
trololo 3 1 7 1 7 3 5 5 7 1 1 1 5
output
Yes Yes No

B. Prefix Function

2 seconds, 256 megabytes

Compute prefix function for given string s .

Input

Single line contains a string s containing only letters ($1 \leq |s| \leq 10^6$).

Output

Output values of the prefix function for string s for indices $1, 2, \dots, |s|$.

input
aaaAAA
output
0 1 2 0 0 0

C. Z-function

2 seconds, 256 megabytes

Compute Z-function for a given string s .

Input

Single line contains string s of only Latin letters ($1 \leq |s| \leq 10^6$).

Output

Output values of the Z-function for string s for indices $2, 3, \dots, |s|$.

input
aaaAAA
output
2 1 0 0 0

input
abacaba
output
0 1 0 3 0 1

D. Fast substring search

2 seconds, 256 megabytes

For two strings s and t find all the occurrences of the string s in the string t .

Input

First line contains string s , second — t ($1 \leq |s|, |t| \leq 10^6$). Strings consist of lowercase english letters.

Output

On the first line output number of occurrences of string s in string t . On the second line output all indices where string s occurs in the string t in ascending order, indices begin from 1.

input
aba abaCaba
output
2 1 5

E. Period

2 seconds, 256 megabytes

For given string s find it's period, i.e. minimum string t , such that s can be represented as one or more concatenations of string t .

Input

Single line consists of string s , containing only letters ($1 \leq |s| \leq 10^6$).

Output

Output length of the string t .

input
abcbabcabc
output
3

input
abacaba
output
7

F. Substrings-3

2 seconds, 256 megabytes

You are given K strings of small english letters. Find their longest common substring.

Input

First line has a single integer K ($1 \leq K \leq 10$).

Next K lines hold K given strings (each string is at least 1 and at most 10 000 characters long).

Output

Output longest common substring

input
3 abacaba mycabarchive acabistrue
output
cab

G. Multiple search

3 seconds, 1024 megabytes

You are given an array of strings s_i and a string t . For each line s_i determine whether it appears in t as a substring.

Input

The first line of the input file contains an integer n — the number of elements in s ($1 \leq n \leq 10^6$). The next n lines contain one of s_i each. The sum of the lengths of all lines from s does not exceed 10^6 . The last line of the input file contains t ($1 \leq t \leq 10^6$). All strings consist of lowercase latin letters.

Output

For each line s_i print «YES» if it occurs in t and «NO» otherwise. Lines are numbered in the order they appear in the input file.

input
3 abc abcdr abcde xabcdef
output
YES NO YES

H. Multiple search 2

3 seconds, 1024 megabytes

You are given an array of strings s_i and a string t . For each line s_i determine how many times it occurs in t as a substring.

Input

The first line of the input file contains an integer n — the number of elements in s ($1 \leq n \leq 10^6$). The next n lines contain one of s_i each. The sum of the lengths of all lines from s does not exceed 10^6 . The last line of the input file contains t ($1 \leq t \leq 10^6$). All strings consist of lowercase latin letters.

Output

For each line of s_i print one number: how many times it occurs in t . Lines are numbered in the order they appear in the input file.

input
3 abc abcdr abcde xabcdef
output
1 0 1

Statement
is not
available
on
English
language

I. Множественный поиск 3

3 секунды, 1024 мегабайта

Дан массив строк s_i и строка t . Требуется для каждой строки s_i найти самое левое и самое правое вхождение в t как подстроки.

Входные данные

Первая строка входного файла содержит целое число n — число элементов в s ($1 \leq n \leq 10^6$). Следующие n строк содержат по одной строке s_i . Сумма длин всех строк из s не превосходит 10^6 . Последняя строка входного файла содержит t ($1 \leq t \leq 10^6$). Все строки состоят из строчных латинских букв.

Выходные данные

Для каждой строки s_i выведите два числа: индексы самой левой и самой правой позиции, в которых она встречается в t . Если строка не встречается в t ни разу, выведите -1 -1. Строки нумеруются в порядке появления во входном файле. Позиции нумеруются с 0.

входные данные
3 ab bcd abde abcdab
выходные данные
0 4 1 1 -1 -1

J. Suffix array

2 seconds, 512 megabytes

Build a suffix array for a given string s , for each two adjacent suffixes find the length of longest common prefix.

Input

First line holds a single string s ($1 \leq |s| \leq 400\,000$). String consists of small english letters.

Output

In first line output $|s|$ distinct integers — numbers of first symbols of s suffixes in a way, that according suffixes will be lexicographically sorted in ascending order. In second line output $|s| - 1$ integers — lengths of longest common prefixes.

input
ababb
output
1 3 5 2 4 2 0 1 1

Statement
is not
available
on
English
language

K. Количество подстрок

2 секунды, 512 мегабайт

Вычислите количество различных подстрок строки s .

Входные данные

Единственная строка входного файла содержит строку s ($1 \leq |s| \leq 400\,000$). Строка состоит из строчных латинских букв.

Выходные данные

Выведите одно число — ответ на задачу.

входные данные
ababb
выходные данные
11

L. Cyclic shifts

2 seconds, 512 megabytes

k-th cyclic shift of a string *S* is a string made by moving *k* first symbols of *S* to its end.

Consider all cyclic shifts of string *S* and sort them in ascending order.

Calculate *i*-th element of that order.

For example, for string `abacabac` 4 cyclic shifts exist: 0-th (`abacabac`), first (`bacabaca`), second (`acabacab`) and third (`cabacaba`). Sorted ascendingly results in array: `abacabac`, `acabacab`, `bacabaca`, `cabacaba`.

Input

First line of input has string *S*, at most 100 000 characters long, chars have ASCII-code from 32 to 126. Second line has a single integer *k* ($1 \leq k \leq 100\,000$).

Output

Output *k*-th cyclic shift of string *S*, our IMPOSSIBLE if it doesn't exist.

input
abacabac 4
output
cabacaba

input
abacabac 5
output
IMPOSSIBLE

M. Longest common substring

2 seconds, 512 megabytes

Find the longest common substring of two given strings *s* and *t*.

Input

First line of the input has single string *s*, second — *t* ($1 \leq |s|, |t| \leq 100\,000$). Strings are made of small latin letters.

Output

Output single line — the longest common substring of strings *s* and *t*. Output lexicographically minimal one, in case of multiple possible answers.

input
bababb zabacabba
output
aba