Problem A. Suffix Array

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Build the suffix array for the given string s constisting of lowercase Latin letters.

Input

The only string of input contains a string s ($1 \le |s| \le 400\,000$). All charachers are lowercase Latin letters.

Output

Output integers $p_1, p_2, \ldots, p_{|s|}$ separated by spaces, such that the suffix of s starting at position p_i is lexicographically smaller than the suffix starting at p_{i+1} for all valid i.

Example

standard input	standard output
ababb	1 3 5 2 4

Problem B. K-th substring

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Given a string s and an integer k, find the k-th substring of s in lexicographic order. Formally, let $\{t_1, \ldots, t_m\}$ be the set of all substrings of s (there are no repetitions in this set). If $t_1 < t_2 < \ldots < t_m$, print t_k.

Input

The first string of input contains a single string s consisting of lowercase Latin letters ($|s| \le 10^5$).

The second string of input contains a single integer k ($1 \le k \le 10^{18}$).

Output

Output the k-th substring of s in lexicographic order. If s doesn't have k distinct substrings, output its lexicographically largest substring.

Example

standard input	standard output
abacaba 10	acab

Problem C. Refrain

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Consider a sequence of length n consisting of integer numbers from 1 to m.

A contiguous subsequence of this sequence is called a refrain if the product of its length and its number of occurences in the initial sequence is maximum possible. Find the refrain of the given sequence.

Input

The first line of input contains two integers, n and m, separated by spaces ($1 \le n \le 150\,000$, $1 \le m \le 10$).

The second line contains n integers varying from 1 to m.

Output

First, print the product of the refrain's length and the number of occurences.

Second, print the length of the refrain.

Third, print the refrain itself.

Example

standard input	standard output
9 3 1 2 1 2 1 3 1 2 1	9 9 1 2 1 2 1 3 1 2 1

Problem D. Mystery Key

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Hercule Poirot is solving a mystery. He knows it must be deduced from the string s. Denote $f(w)$ as the length of the longest proper suffix of w (i.e. not coinciding with the entire w) which is also a prefix of w. For example, $f(\texttt{abc}) = 0$, $f(\texttt{abab}) = 2$, $f(\texttt{aaa}) = 2$.

The key to the mystery is equal to the maximum value of $|t| + f(t)^2$ among all t which are substring of s. Help Hercule solve thid riddle!

Input

The only line of input contains a stirng s of lowercase Latin letters ($1 \le |s| \le 500\,000$).

Output

Print the maximum value of $|t| + f(t)^2$ among all substrings of s.

Example

standard input	standard output
ababaab	14

Problem E. Keep It Counted

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Vova likes algorithms that deal with strings. He usually prepares programming contests devoted to string-based algorithms. One day, he prepared such a contest and showed it to Pasha.

Pasha, in turn, does not like string-based algorithms, but he likes hash functions. He solved the most difficult problem of the contest with a help of a hash function.

Vova is furious about that. He wants to create a test that will make Pasha's solution get "Time limit exceeded" verdict. He knows that the main idea of this solution and wants to create a string S that has a vast number of distinct substrings in any of its prefixes.

Now Vova needs a program that will count the number of distinct substrings for each prefix of the given string S. You are to write such program.

Input

The only line of input contains a nonempty string S consisting of N ($1 \le N \le 2 \cdot 10^5$) lowercase English letters.

Output

Output N lines. On i-th line, output the number of distinct substrings of i-th prefix of string S.

Examples

standard input	standard output
aabab	1 2 5 8 11
atari	1 3 5 9 14

Problem F. Common Substrings

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

You are given a collection of k strings of lowercase Latin letters. The sum of lengths of these strings is n.

For an integer i denote ℓ_i as the length of the longest string which is a substring of at least i strings of the initital collection. Calculate ℓ_i for every i from 2 to k.

Input

The first line of input contains a single integer k ($1 \le k \le 200\,000$) — the number of strings.

The following k lines contain the strings of the collection. It is guaranteed that the sum of lengths of all strings (n) doesn't exceed 200 000.

Output

Print $\ell_2, \ell_3, \ldots, \ell_k$ on separate lines.

Example

standard input	standard output
6 matter animate pattern thermal domain teammate	5 3 2 2 1

Problem G. Nenokku

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Given a string. Initially string is empty.

Your task is to process queries of two types:

1. append some string to the end of our string or
2. check if the given word occurs in our string as a substring.

Input

Input consists of queries. Each line of the input is one query:

1. `A <string>` (append a string `<string>` consisting of no more than 10^5 lower- and uppercase English letters, to the end of the string);
2. `? <string>` (check if `<string>` consists of no more than 50 lower- and uppercase English letters exists in our string).

Length of the big string never exceeds 10^5, summary length of all queries does not exceed 15 mebibytes plus 12140 bytes.

Output

For each '`?`' request print "`YES`", if the given string occurs in the big string as a substring, and "`NO`" otherwise. The queries are **case-insensitive**.

Example

standard input	standard output
? love ? is A Loveis ? love ? WHO A Whoareyou ? is	NO NO YES NO YES

Problem H. Substrings

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You are given k strings consisting of lowercase Latin letters. You are to find their longest common substring.

Input

The first line of input contains an integer k ($1 \leqslant k \leqslant 10$).

The following k lines contain the given strings (their lengths are from 1 to 10 000).

Output

Print the longest common substring of all these strings.

Example

standard input	standard output
3 abacaba mycabarchive acabistrue	cab

Problem I. Suffix Tree

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1.5 seconds
Memory limit:	256 megabytes

Given a string s, build its suffix tree.

Input

The only line of input contains the string s ($1 \le |s| \le 10^5$) consisting of lowercase Latin letters and terminating by a dollar sign ('$').

Output

To output the tree, number all vertices 0 through $n-1$ in order of DFS traversal, selecting lexicographically smaller characters to descend to first. Use ASCII-codes to compare characters.

The first line of output should contain an integer n corresponding to the number of vertices in the tree. The following $n-1$ lines should contain the description of vertices 1, 2, ..., $n-1$.

The description of vertex v consists of three integers: p, lf, rg, where p ($0 \le p < n, p \ne v$) is the number of v's parent vertex. The edge leading from p to v should contain the substring $s[lf \dots rg-1]$ ($0 \le lf < rg \le |s|$).

Examples

standard input	standard output
aaa$	7 0 3 4 0 2 3 2 3 4 2 2 3 4 3 4 4 2 4
b$	3 0 1 2 0 0 2
ababa$	10 0 5 6 0 4 5 2 5 6 2 3 5 4 5 6 4 3 6 0 3 5 7 5 6 7 3 6

Problem J. 3-Substrings

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

You are given a string s of length n. For each $k \in \{1, 2, \ldots, \lfloor \frac{n}{3} \rfloor\}$ you should find the number of different substrings of s of length exactly k such that each of them has at least **three** pairwise non-overlapping occurences in s.

Input

The only line of input contains the string s ($3 \le |s| \le 100\,000$) consisting of lowercase Latin letters.

Output

Output $\lfloor \frac{n}{3} \rfloor$ numbers: the answer for every k.

Examples

standard input	standard output
abracadabra	1 0 0
abacabaabacabaabacaba	3 4 4 4 3 2 1
aaaa	1