Problem A. Two Strings

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

You are given two strings a and b. Find shortest string which being repeated infinitely contains the both strings. I.e. find such shortest s that infinite string $ss\ldots s\ldots$ contains a and contains b.

Input

The first line contains number of test cases (between 1 and 100 inclusive). Each testcase consists of two lines: string a and string b. Each of them is not longer than 10000. Strings contain characters with ASCII codes from 33 to 126 inclusive.

Output

For each test case print two lines: length of the answer and answer itself. Separate answers for test cases with empty line.

Example

standard input	standard output
2 abcabc bcabca toolkit kitten	3 abc 10 toolkitten

Problem B. Epigraph

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

Young writer has just finished the text of his first novel. As an epigraph he has chosen the sequence obtained as a concatenation of all distinct substrings of the novel name.

Substring of a given string is a sequence of consecutive characters. You are asked to find the length of the resulting epigraph.

Input

The only line contains the name of the novel consisting of lowercase and uppercase latin letters. Its length is at least 1 and doesn't exceed 8000 characters.

Output

Print one integer — the length of the epigraph.

Examples

standard input	standard output
BOBR	19
ABACABA	73

Note

In the first sample, epigraph is obtained using substrings "B", "O", "R", "BO", "OB", "BR", "BOB", "OBR", and "BOBR". Total length is 19.

Problem C. Inaccurate Search

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

Given text $t = t_1\ t_2\ \ldots\ t_{|t|}$ and pattern $p = p_1\ p2\ \ldots\ p_{|p|}$. Also given the parameter k.

For position d if $t[d\ \ldots\ d + |p| - 1] = p$ then there is an exact matching from the position d.

For position d if in substring $t[d\ \ldots\ d + |p| - 1]$ there is a window (segment) of f consecutive characters which can be changed to make the substring be equal to p and $f \leq k$ then it is inaccurate matching. Obviously, an exact matching is special case of an inaccurate matching.

For example, if $k = 3$, t="`abacabadabacaba`" and p="`baxayad`", there is one inaccurate matchings from the position 2 (indices are 1-based).

Your task is to find all inaccurate matchings of p for given text t.

Input

The first line contains the text t. The second line contains the pattern p. Both of them contain only Latin letters. Their lengths are between 1 and 10^6, inclusive. The third line contains k ($0 \leq k \leq 10^6$).

Output

Print the number of inaccurate matchings and the corresponding positions in increasing order.

Examples

standard input	standard output
abacaaa aaa 1	4 1 3 4 5
ababa aaaaa 3	1 1

Problem D. Olympiad string

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes

On one very famous olympiad programming site you can enter some additional information about your team. You can enter the full name of your team, the name of your university, city and so on. Entering this information is not very easy, because the authors of the site have added some JavaScript to the corresponding web-page, so during entering the information you are limited with next two operations:

1. Enter one symbol to the beginning of the line, or
2. Copy any part of already typed string to the beginning of the line.

First you wanted to find such algorithm to type the name of your team that will take the minimal time. But then you realized that this algorithm will probably be not so good. Really, copying a part of a string takes more time than just typing a character, but when you are typing a character you can do a mistake (you can press a wrong key), but when you are copying a part of a string, if the algorithm is correct, you won't do any mistake. So you decided not to think about time, but find such algorithm to enter the name of your team, which will require the minimal possible number of operations. Write a program, which will find such a way for any string.

Input

Input will consist of several tests (at most 50). For each test one string of lowercase and uppercase latin letters will be given — the string, for which you have to find the optimal algorithm. The length of each string will not exceed 4 000. The sum of lengths over all tests doesn't exceed 20000.

Output

For each test, write to output the algorithm, which will require the minimal number of operations. Start describing each algorithm with string "Typing this string will require K operations" (without quotes), where K is the number of operations in your algorithm. Then output one line with text "These operations are the following:" without quotes. Then output K lines describing the operations. For each operation, output it type (`1` for typing a character, or `2` for copying a string). If the type is `1`, output then one space and then the character itself. If the type is `2`, output then two numbers — the positions of the first and the last characters of copied part in the string before performing this operation.

If several solutions exist, output any.

Example

standard input	standard output
dabfabc a	Typing this string will require 6 operations These operations are the following: 1 c 1 b 1 a 1 f 2 2 3 1 d Typing this string will require 0 operations These operations are the following: Typing this string will require 1 operations These operations are the following: 1 a

Problem E. Fibonacci Strings

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

Memory limit is 1024 megabytes.

The sequence $f_0\ f_1\ \ldots$ is Fibonacci Strings if:

- $f_0 =$ “”
- $f_1 =$ “a”
- $f_2 =$ “b”
- $f_3 =$ “ab”
- $f_4 =$ “bab”
- ...
- $f_i = f_{i-2} + f_{i-1}$

Given string s containing only letters ‘a’ and ‘b’ and nonnegative integer k.

Find the number of occurrences string s in the string f_k modulo $1\,000\,000\,009$ $(10^9 + 9)$.

Input

The first line contains non-empty s, the length of s is not greater than 10^3. It contains only ‘a’ and ‘b’.

The second line contains integer k $(0 \le k \le 10^4)$.

Output

Print the number of occurrences modulo $1\,000\,000\,009$ $(10^9 + 9)$.

Examples

standard input	standard output
b 4	2
ab 5	2
a 7	5
abab 1009	294248075

Problem F. Censored

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

Count the number of strings of length m over alphabet of size n, and do not contain any "forbidden" word from a specified set of words as a substring.

Input

The first line contains integers n, m, p ($1 \le n, m \le 100$, $0 \le p \le 10$) — the size of the alphabet, the length of desired strings, and the number of forbidden words.

The next line contains n characters with ASCII code strictly greater than 32 that denote the letters of the alphabet.

Next p lines contain forbidden words. Each of the words consists exclusively of the alphabet letters, and the length of each word does not exceed $\min(m, 10)$.

Output

Print a single integer — the answer to the problem.

Example

standard input	standard output
2 3 1 ab bb	5

Problem G. Keep It Counted

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
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	
atari	

Problem H. Suffix Automaton

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

You are given a string. Construct its suffix automaton.

Input

A string of 1 to 10^5 lowercase English letters.

Output

On the first line print n and m — the number of states and arcs of the automaton.

On the next m lines print descriptions of arcs. Each arc should be described by the start state, target state, and the corresponding letter.

On the next line print the number of terminal states, followed by the line with their numbers. The initial state should have number 1.

Example

standard input	standard output
ababb	7 9 1 2 a 1 7 b 2 3 b 3 4 a 3 6 b 4 5 b 5 6 b 7 4 a 7 6 b 3 6 7 1

Problem I. Nenokku

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

There is an initially empty text string. You should process the following queries:

1. `? <word>` (<word> consists of at most 50 English letters) — determine if <word> is a substring of the text.
2. `A <text>` (<text> consists of at most 10^5 English letters) — append <text> to the text string.

It is guaranteed that the total number of characters added to the text will not exceed 10^5. The total size of all queries does not exceed 16MB.

Input

Queries, one per line.

Output

For each query of the first type print "`YES`" if <word> is a substring of the text, and "`NO`" otherwise.

Example

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