,

Problem A. Restore the string

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

Jonathan tried to write Rabin-Karp algorithm. He had a string S and he used the following formula to calculate hash of string:

$$\sum_{i=0}^{|S|-1} (S_i - 97) \cdot 2^i$$

Thus, he has written hashes of all prefixes of string S. Unfortunately, Jonathan forgot his string S. So, he asked your help in restoring this string.

Input

The first line of the input contains the only integer N - the length of string S ($1 \leqslant N \leqslant 50$).

The second line contains N integers p_i - hashes of all prefixes of string S.

It is guaranteed that each hash does not exceed $2 \cdot 10^{18}$.

Output

Print the string S.

Examples

standard input	standard output
5 7 15 59 147 371	hello
5 22 50 118 206 254	world

Problem B. Forgotten password

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

Jonathan forgot his password to Q of his website accounts, the only thing he remember is several last letters of his password for each account. Also Jonathan has a list of passwords that he use to authorize to different websites. He wants to know how many passwords should he bruteforce to authorize each website. Please, help him find this numbers.

Input

The first line of the input contains two integers N and Q - number of passwords in Jonathan's list and the number of websites to authorize.

Each of the next N lines contains one string s_i - password in the list.

Each of the next Q lines contains one string t_i - last symbols of his real password to the i_{th} account.

It is guaranteed that $\sum_{i=1}^{n} |s_i| \leqslant 2 \cdot 10^5$ and $\sum_{i=1}^{q} |t_i| \leqslant 2 \cdot 10^5$, e.g. both sum of lengths of all passwords and sum of lengths of all last-symbols-strings does not exceed $2 \cdot 10^5$.

It is also guaranteed that all strings consist of only lowercase latin letters.

Output

Please, output the number of suitable passwords to each of the Jonathan's website accounts.

Examples

standard input	standard output
4 2 september october november december ber mber	4 3
4 4 stay hungry stay foolish stay tay ay y	2 2 2 3

Problem C. Pen-pineapple-apple-pen

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

Jonathan has recently seen 'PPAP' music video by Pikotaro. He has two strings and he decided to concatenate them. But the way he wants to do that is unusual. If the second string starts with the same symbols as the first string ends, then this substring is common for both strings. Jonathan wants to avoid duplication of common substring in resulting string. If there are multiple appropriate concatenations, Jonathan chooses one with the longest common part. If there is no common part, he just concatenate them in a usual way. Please, help Jonathan obtain new string.

Input

The only line of the input contains two strings S and T ($1 \leqslant |S|, |T| \leqslant 1000$).

Output

Print the resulting string - concatenation of strings S and T.

Examples

standard input	standard output
pen pineapple	penpineapple
pineapple apple	pineapple
pineapple lemon	pineapplemon

Problem D. Mortal Kombat

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

Jonathan likes playing videogames. One of his favorite games is Mortal Kombat. In this game in addition to usual hits, any character can make a super hit if player does special sequence of clicks. Also, combo hits can be performed. Combo is an uninterrupted sequence of super hits. So, you know the sequence of clicks that must be made in order to complete super hit playing for the character Jonathan has chosen and know the sequence of clicks that he made during last round. Please, help Jonathan count the longest combo that he performed.

Input

The first line of input contains string S - sequence of clicks for super hit.

Next line contains string T - sequence of clicks that Jonathan made in last round.

It is guaranteed, that $1 \leqslant |S| \leqslant |T| \leqslant 10^5$.

It is also guaranteed, that strings consist of only lowercase latin letters.

Output

Print the longest combo for the Jonathan in the last round.

Examples

standard input	standard output
a aabaaa	3
aba abababa	1

Problem E. String manipulation

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

Have you ever tried to pronounce a word without vowels? Perhaps you also tried to pronounce it in the reversed order. Zhandos tries to do both actions at once on the same word. The input is given a string s of even length, which can contain letters and numbers. Remove all vowels from the first half of the word and reverse the second half of the word. Output the received string (the received string must begin with a capital letter).

Input

The line contains a non-empty string s of even length.

Output

Output the string after manipulations.

Examples

standard input	standard output
0VxrJXx7BM	0VxrJMB7xX
j1g9U1umMm	J1g9mMmu1
xDwSbKNPhk	XDwSbkhPNK
Z0UMi9Uees	Z0MseeU9

Note

In the English alphabet, the letter Y can be both a vowel and a consonant letter. In this task, you can only consider it as a vowel.

Problem F. Dominating Patterns

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

The archaeologists are going to decipher a very mysterious "language". Now, they know many language patterns; each pattern can be treated as a string on English letters (only lower case). As a sub string, these patterns may appear more than one times in a large text string (also only lower case English letters). What matters most is that which patterns are the dominating patterns. Dominating pattern is the pattern whose appearing times is not less than other patterns. It is your job to find the dominating pattern(s) and their appearing times.

Input

The entire input contains multi cases. The first line of each case is an integer, which is the number of patterns N, 1 <= N <= 150. Each of the following N lines contains one pattern, whose length is in range [1, 70]. The rest of the case is one line contains a large string as the text to lookup, whose length is up to 106 . At the end of the input file, number '0' indicates the end of input file.

Output

For each of the input cases, output the appearing times of the dominating pattern(s). If there are more than one dominating pattern, output them in separate lines; and keep their input order to the output.

Example

standard input	standard output
2 aba bab abababababac 6 beta alpha haha delta dede tata dedeltalphahahahototatalpha 0	4 aba 2 alpha haha

Note

You should solve this problem using Rabin-Karp algorithm

Problem G. Uragirimono no Requiem

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

You are given string $s(1 \leq |s| \leq 5 * 10^5)$ that consists of english lowercase letters, and a dictionary of size $m(1 \leq m \leq 5 * 10^5)$. Determine if it's possible to split into two non-empty strings s1 and s2, such that s1 + s2 = s and both s1 and s2 exist in dictionary.

Input

First line contains string s. Second line contains integer m - number of words in dictionary. The next m lines contain lowercase english letters - words in dictionary. It's guranteed that sum of sizes of words in dictionary $\leq 5 * 10^5$.

Output

Output 'YES' if answer exists and 'NO' otherwise.

Examples

standard input	standard output
goldenwind 2 golden wind	YES
goldenwind 2 goldenw wind	NO
jojoreference 5 jojo reference lol kek d	YES

Problem H. Modified "Towns"game

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

One day Olzhas was bored and he wanted to play the game "towns"with friends. But the game has been slightly modified. In this game, each participant in turn calls another real city of any country, the name of which begins with the maximum possible length of suffix, which ends with the name of the city of the previous participant. It was Olzhas' turn and he should choose the name of the city. Help him with the choice of the name of the city.

Input

Given string P ($1 \leq |P| \leq 400$) name of the city of previous participant. In next line given N ($1 \leq N \leq 10^3$) - numbers of city names which Olzhas know. Next N lines represent name a_i ($1 \leq |a_i| \leq 400$) of the cities. Each name of the city start with Upper case and other characters in the name are lowercase.

Output

First line should be M number of possible names of the cities which could say Olzhas. Next M lines should consist from this possible names of the cities in the order of their input.

Examples

standard input	standard output
Kokshetau 5 Astana Tauemel Tainan Almaty Budapest	1 Tauemel
Almaty 3 Yacuiba Yurga Moscow	2 Yacuiba Yurga

Problem I. Phone Operators

Input file:	`standard input`
Output file:	`standard output`
Time limit:	0.5 seconds
Memory limit:	256 megabytes

You are given codes of mobile operators. You need to determine whether the phone number belongs to some mobile operator. The phone number belongs to the mobile operator if the first digits of the phone number coincide with the code of the mobile operator.

Input

You are given N and $M(1 \leq N, M \leq 3 \times 10^4)$ - number of mobile operators and number of phone numbers. Next N lines consist from c_i codes of mobile operator. Next M lines consist from phone numbers, where p_i is i^{th} phone number. Each phone number and code of operator consist only of digits. Length of each phone number and code of operator not exceed 1000 and consist from at least one digit.

Output

For each phone number if it's belongs to some operator print 'YES' , otherwise 'NO'.

Example

standard input	standard output
3 5 747 8702 9 747697289 870 722678 87021234567 9	YES NO NO YES NO

Problem J. Cyclic Shift

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

Tamerlan once wrote on a piece of paper a line consisting of large and small Latin letters, and then went to help Askar. When he returned, he found that his friend Alikhan had written another line of the same length under his line. Alikhan claims that he got his line by cyclic shift of Tamerlan's line a few steps to the right (cyclic shift of qwerty line for 2 positions to the right will give ertyqw line). However, Alikhan is known for the fact that he can accidentally make a mistake in a large number of calculations, so Tamerlan is at a loss – whether to believe Alikhan? Help him! From given line, print the minimum possible shift size or -1 if Alikhan is wrong.

Input

The first two lines of input data contain Tamerlan and Alikhan lines, respectively. The line lengths are the same, do not exceed 10000 and are not equal to 0.

Output

Print the single number - the answer to the problem question.

Examples

standard input	standard output
zabcd abcdz	4
abcde decba	-1

Problem K. Rhyme time

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

Jonathan likes poetry and he decided to write a poem. Unfortunately, inspiration has left him and he cannot determine does two words rhyme or not. Therefore he asked your help.

Note, that two words rhymes if and only if their emphasised vowels coincide and parts of words after emphasised vowels coincide too.

Input

Two lines of the input contain two words s and t ($1 \leqslant |s|, |t| \leqslant 10^5$).

It is guaranteed, that there is only one emphasised vowel in each word and it is always uppercase, while other letters are lowercase.

Output

Help Jonathan and print YES, if words rhyme, otherwise print NO.

Examples

standard input	standard output
potAto tomAto	YES
alOha alohA	NO

Problem L. Big (and not baby) tape

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	256 megabytes

You have a long tape which is divided into square cells by vertical lines. There is a lowercase latin letter in each cell. You also have n tapes of the same format but of the smaller length. Finally, you have device that can make an infinite number of copies of each tape, except the longest. Your task is to cover the longest tape with smaller tapes such that a) each cell is covered by at least one tape; b) only cells with the same letter can lie on them; c) there is no cell of smaller tape that lie outside the longest tape.

Input

The first line of the input contains the string s - the longest tape ($1 \leqslant |s| \leqslant 10^5$).

The second line contains the only integer n - number of smaller tapes ($1 \leqslant n \leqslant 500$).

Each of the next n lines consists of string t_i - the i_{th} smaller tape ($1 \leqslant |t_i| \leqslant |s|$).

Output

If you can cover big tape in accordance with the rules, print YES, otherwise print NO.

Examples

standard input	standard output
bigandnotbabytape 5 big and not baby tape	YES
abaaba 2 ab ba	YES
abaaba 1 baab	NO