Problem A. 109670. Hash of digits

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

You have N strings s^i consisting of digits and hashes h^i of these strings, where hash of string s^i equal to h^i. But one day You noticed that hashes and strings were mixed. You need to determine which of the numbers are hashes and which are strings consisting of digits. Also map each string to its hash. Hash should be calculated by formula below. All operations must be performed modulo $1e9 + 7$:

$$h = (\sum_{i=0}^{|s|-1} (s_i - 47) \times 11^i) \mod (10^9 + 7)$$

Input

Given the number N - number of strings ($1 \leq N \leq 10^4$). Following $2 \cdot N$ lines consist from strings and hashes. The length of each string does not exceed 100. It is guaranteed that for each string there is a corresponding hash.

Output

In each line print string with corresponding hash like in example below in order their input.

Examples

standard input	standard output
3 334368200 111 100000 266 123456789 93085	Hash of string "111" is 266 Hash of string "123456789" is 334368200 Hash of string "93085" is 100000

3
13903
338308044731467504464331 3
9839
813695185
425675346
3461762860035486
	Hash of string "3383080447314675044643313" is 425675346
	Hash of string "9839" is 13903
	Hash of string "3461762860035486" is 813695185

Note

In first example, hash of string "111" was calculated as follows:

$$h = \Big(\Big(\big((49 - 47) \times 11^0\big) \mod (1e9 + 7) + \big((49 - 47) \times 11^1\big) \mod (1e9 + 7)\Big) \mod (1e9 + 7) + \\ + \big((49 - 47) \times 11^2\big) \mod (1e9 + 7)\Big) \mod (1e9 + 7) = 2 + 22 + 242 = 266$$

Note, that the resulting number should be positive.

Problem B. Plagiarism

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

Alimzhan wanted to check his students for cheating and he added the parasite word to his exam. He believes that students cheat if the parasite word occurs in the same place in two different results of two students. Alimzhan heard that two of his most beloved students were copied from each other. Help Alimzhan know how many times the parasite word is repeated in the same position in his beloved students' results.

Input

The first line contains string s_1 ($1 \leq |s_1| \leq 100000$) - result of first student.

The second line contains string s_2 ($1 \leq |s_1| \leq 100000$) - result of second student.

The third line contains string t ($1 \leq |t| \leq 100000$) - parasite word.

Output

Output number of positions where parasite word meets in both students' results in increasing order.

Examples

standard input	standard output
kbtuutbkkbtu utbkkbtukbtu kbtu	1
aaaaa aaaaa a	5
abracadabra abacabaabac ab	2

Problem C. 109920. 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`

Note

Solve this task using hash algorithm.

Problem D. 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 of lowercase Latin letters. There is a large text they want to analyze. What matters most is that which patterns are the *dominating patterns*. *Frequency* of the pattern is the number of its appearances in the text as a substring. Pattern is dominating if the frequency the pattern is is not less than the frequency of any other pattern. It is your job to find all dominating patterns.

Input

The input consists of multiple test cases.

The first line of each test case is an integer N $(1 \leq N \leq 150)$ - the number of patterns. Each of the following N lines contain single string of length in range $[1, 70]$ - patterns. The next line contains a single string of length in range $[1, 10^5]$ - text.

At the end of the input file there is a 0 indicating the end of an input file.

Output

For each test case, output the dominating patterns. If there are several dominating patterns, output them in order they appear in the input.

Example

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

Note

In the first test case, the frequency of pattern *aba* is 2, and of pattern *bac* is 1.

In the second test case, the frequency of patterns *alpha* and *haha* is 2, and all other patterns has frequency not exceeding 1.

Problem E. 108995. 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`

Note

Use larger integer data structures (like “long long” In C++).

Problem F. Distinct substrings

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

Given string s. How many different substrings it contains?

Input

Line contains string s ($0 \leqslant s.length \leqslant 2000$).

Output

Output single integer - output to the problem.

Examples

standard input	standard output
aab	5
abcde	15

Note

You can solve this problem using hashes. Naive $O(N^3)$ solutions must fail.

Problem G. Substrings

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	512 megabytes

You have string S and q queries. Each query given a pair of integers L and R; the answer for a query is number of times substring from L to R appears in S as substring.

Input

In first line given string S ($1 \leq |s| \leq 2000$).

In second line given integer q ($1 \leq q \leq 5 * 10^4$).

Then next q lines contains queries, each query is two integer L and R ($1 \leq L \leq R \leq |s|$), substring you must to check.

Output

Print q lines with one single integer, answer for each query.

Example

standard input	standard output
abracadabra	5
5	2
1 1	2
1 2	1
3 4	2
1 5	
1 4	

Problem H. LCS

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

You have K strings, find the longest common substring of all K strings.

Input

You are given K ($1 \leq K \leq 10$) strings s ($1 \leq |s| \leq 5000$), consisting of small Latin letters.

Output

Print one string, the longest common substring of all K strings. It is guaranteed that it is always exist.

Example

standard input	standard output
3 abacaba mycabarchive acabistrue	cab