Problem E. Expression

Input file:	`expression.in`
Output file:	`expression.out`
Time limit:	10 seconds
Memory limit:	256 megabytes

In computing, regular expressions is a powerful tool for text search and string matching. In this problem a simplified version of regular expressions is used:

- An empty string "" is a regular expression, only the empty string matches it.
- A single lowercase letter "c" is a regular expression, a string consisting of a single letter c matches it.
- A dot "`.`" is a regular expression, a string consisting of any single letter matches it.
- Alternation: if α and β are regular expressions then "$(\alpha|\beta)$" is a regular expression, a string s matches it only if s matches α or s matches β.
- Concatenation: if α and β are regular expressions then "$(\alpha\beta)$" is a regular expression, a string s matches it only if $s = xy$, x matches α and y matches β.
- Kleene star: if α is regular expression then "$(\alpha*)$" is a regular expression, a string s matches it only if s is empty or $s = xy$, x is nonempty and matches α and y matches $(\alpha*)$. In other words, s consists of zero or more strings, each of them matches α.

Parentheses can be omitted, in this problem Kleene star has the highest priority, concatenation has medium priority and alternation has lowest priority. Thus "`abc*|de`" means "`(ab(c*))|(de)`".

For example, string "`abcabcab`" matches "`a(bc|a)*ab`", but string "`abcbab`" does not.

Your task is to find the shortest string that matches the given regular expression E and contains the given substring S.

Input

The first line of the input file contains the regular expression E. The second line of the input file contains the substring S ($1 \le |E|, |S| \le 10\,000$).

String S consists of lowercase English letters. Expression E consists of lowercase English letters and special characters: dots ('`.`'), parentheses ('`(`' and '`)`'), pipes ('`|`'), and asterisks ('`*`').

Output

Output the shortest possible string T that both matches E and contains S as substring. If there are no such strings, output "`NO`".

The string T should contain only lowercase English letters.

Examples

expression.in	expression.out
a.*b bab	abab
(ab)* bb	NO