ACSL Reg Exp
JUNIOR DIVISION

PROBLEM: Regular Expressions and FSA is one of the ACSL categories that is not tested in the Junior Division. However it is being used here as a programming problem to show how a long string or multiple strings can be written in condensed format. The concept of regular expressions was first formalized by Stephen Kleene in the 1950s. The following operations will be used:

? The question mark indicates there is zero or one of the preceding element. For example, colou?r matches both "color" and "colour".

* The asterisk indicates there is zero or more of the preceding element. For example, ab*c matches "ac", "abc", "abbc", "abbbc", and so on.

+ The plus sign indicates there is one or more of the preceding element. For example, ab+c matches "abc", "abbc", "abbbc", and so on, but not "ac".

INPUT: There will be 6 lines of input. The first line will contain 10 character strings. The last 5 lines will contain a valid regular expression string. Each regular expression will have no more than 3 characters and no more than two operators.

OUPUT: For each regular expression print all the character strings that are matches. If none match, then print NONE. # is used here to represent the empty string.

SAMPLE INPUT   SAMPLE OUTPUT

1. #, a, aa, aaa, ab, abb, aabb, aaab, abbb, b

2. a?b   1. ab, b

3. a*b   2. ab, aaab, b

4. a+b   3. ab, aaab

5. a*b+   4. ab, abb, aabb, aaab, b, abbb

6. a?b+   5. ab, abb, abbb, b