ACSL

 American Computer Science League

**008 2015 - 2016**

**Contest #4**

ACSL Reg Exp
SENIOR DIVISION

PROBLEM: Regular Expressions is one of the ACSL categories that is usually only tested at the ACSL Invitational All-Star Contest. 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 are used:

|  |  |
| --- | --- |
| **˙** | Matches any single character. Ex: a.c matches aac, abc, acc, adc, a#c, a4c, …etc. |
| [ ] | Matches a single character contained within the brackets. Ex: [abc] matches a, b, or c. |
| [^] | Matches any single character not contained within the brackets. [^abc] matches any character other than a, b, or c. |
| \* | Matches the preceding character zero or more times. Ex; a\*b matches b, ab, aab, aaab, …etc. |
| {m,n} | Matches the preceding character at least m but not more than n times. Ex: a{3,5} matches aaa, aaaa, or aaaaa. |

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 at most two (2) operators.

OUPUT: For each regular expression print all the character strings that are matches to the strings on Line #1. If none match, then print NONE. **#** is used here to represent the empty string.

SAMPLE INPUT SAMPLE OUTPUT
1. **#**, aac, acc, abc, ac, abbc, abbbc, abbbbc, aabc, aabbc
2. a.c 1. aac, acc, abc
3. a[ab]c 2. aac, abc
4. a[^ab]c 3. acc
5. ab\*c 4. ac, abc, abbc, abbbc, abbbbc
6. a.b{2,4}c 5. aabbc, abbbc, abbbbc

ACSL

 American Computer Science League

**008 2015 - 2016**

**Contest #4**

 ACSL RegExp
 SENIOR DIVISION

TEST DATA

TEST INPUT TEST OUTPUT

1. **#**, xx, xyzz, xzz, xxyyzz, xyz, xxyz, xyzzz, yzz, xxxyz
2. x.y.z 1. NONE
3. [xy].z 2. xzz, xyz, yzz
4. x\*z\* 3. **#**, xx, xzz
5. y[^xy]z\* 4. yzz
6. x{1,2}.yz 5. xxyz, xxxyz