

  (/amphiSession/instructions/4323?popup=true) This Section has 1 page.

Problem

Program

PROBLEM 6

Alphabet Soup

Develop a program that, given a “bowl of alphabet soup” along with one or more “messages” (i.e., character strings), determines, for each message, whether or not the bowl contains all the letters necessary to form the message.

Regarding the case (lower or upper) of letters, your program should treat a as being indistinguishable from A, b as being indistinguishable from B, etc. Hence, an occurrence of, say, F in a message is “covered by” an occurrence of f in the soup.

Note that the number of occurrences of each letter matters: if, for example, there are three

occurrences of e/E in the soup but four (or more) occurrences of e/E in a message, that message cannot be formed from the letters in the soup.

As for non-letters appearing in a message (e.g., spaces, digits, punctuation symbols, etc.), they should be ignored in determining whether the message can be formed in the soup.

Input Format:

The first line contains a string of letters, in no particular order and (typically) containing duplicates. This line represents the collection of letters in the bowl of soup. You may assume that this string has length between one and 60, inclusive.

On the next line is a single positive integer 'm' indicating how many messages are to follow. Each of the next m lines contains a message, which is simply a string of characters, possibly including non-letters (such as punctuation marks).

You may assume that each message has length between one and 60, too.

Output Format:

Print each given message and report (using either "YES" or "NO") whether or not it can be formed using the letters in the soup. (See sample output below for exact formatting.)

Sample Input :

lhLoe

3

HELLO

eel

eL,\$H.

Sample Output:

HELLO : YES

eel : NO

eL,\$H. : YES

[Sections](#)[Finish](#)