P19

Ancient Roman empire had a strong government system with various departments, including a secret service department. Important documents were sent between provinces and the capital in encrypted form to prevent eavesdropping. The most popular ciphers in those times were so called *substitution cipher* and *permutation cipher*.

Substitution cipher changes all occurrences of each letter to some other letter. Substitutes for all letters must be different. For some letters substitute letter may coincide with the original letter. For example, applying substitution cipher that changes all letters from 'A' to 'Y' to the next ones in the alphabet, and changes 'Z' to 'A', to the message "VICTORIOUS" one gets the message "WJDUPSJPVT".

Permutation cipher applies some permutation to the letters of the message. For example, applying the permutation $\langle 2, 1, 5, 4, 3, 7, 6, 10, 9, 8 \rangle$ to the message "VICTORIOUS" one gets the message "IVOTCIRSUO".

It was quickly noticed that being applied separately, both substitution cipher and permutation cipher were rather weak. But when being combined, they were strong enough for those times. Thus, the most important messages were first encrypted using substitution cipher, and then the result was encrypted using permutation cipher. Encrypting the message "VICTORIOUS" with the combination of the ciphers described above one gets the message "JWPUDJSTVP".

Archeologists have recently found the message engraved on a stone plate. At the first glance it seemed completely meaningless, so it was suggested that the message was encrypted with some substitution and permutation ciphers. They have conjectured the possible text of the original message that was encrypted, and now they want to check their conjecture. They need a computer program to do it, so you have to write one.

Input

Input file contains several test cases. Each of them consists of two lines. The first line contains the message engraved on the plate. Before encrypting, all spaces and punctuation marks were removed, so the encrypted message contains only capital letters of the English alphabet. The second line contains the original message that is conjectured to be encrypted in the message on the first line. It also contains only capital letters of the English alphabet.

The lengths of both lines of the input file are equal and do not exceed 100.

Output

For each test case, print one output line. Output 'YES' if the message on the first line of the input file could be the result of encrypting the message on the second line, or 'NO' in the other case.

Sample Input

```
JWPUDJSTVP
VICTORIOUS
MAMA
ROME
HAHA
HEHE
AAA
AAA
```

NEERCISTHEBEST
SECRETMESSAGES

Sample Output

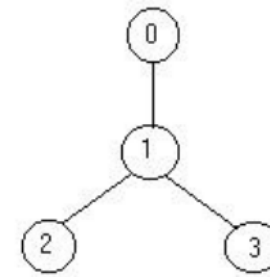
YES
NO
YES
YES
NO

P20

Bob enjoys playing computer games, especially strategic games, but sometimes he cannot find the solution fast enough and then he is very sad. Now he has the following problem. He must defend a medieval city, the roads of which form a tree. He has to put the minimum number of soldiers on the nodes so that they can observe all the edges. Can you help him?

Your program should find the minimum number of soldiers that Bob has to put for a given tree.

For example for the tree on the right the solution is one soldier (at the node 1).



Input

The input file contains several data sets in text format. Each data set represents a tree with the following description:

- the number of nodes
- the description of each node in the following format

  $node\_identifier$`:(`$number\_of\_roads$`)` $node\_identifier_1$ ... $node\_identifier_{number\_of\_roads}$

  or

  $node\_identifier$`:(0)`

The node identifiers are integer numbers between 0 and $n-1$, for n nodes ($0 < n \leq 1500$). Every edge appears only once in the input data.

Output

The output should be printed on the standard output. For each given input data set, print one integer number in a single line that gives the result (the minimum number of soldiers).

Sample Input

```
4
0:(1) 1
1:(2) 2 3
2:(0)
3:(0)
5
3:(3) 1 4 2
1:(1) 0
2:(0)
0:(0)
4:(0)
```

Sample Output

```
1
2
```