

You are planning to take some rest and to go out on vacation, but you really don't know which cities you should visit. So, you ask your parents for help. Your mother says "*My son, you MUST visit Paris, Madrid, Lisboa and London. But it's only fun in this order.*" Then your father says: "*Son, if you're planning to travel, go first to Paris, then to Lisboa, then to London and then, at last, go to Madrid. I know what I'm talking about.*"

Now you're a bit confused, as you didn't expect this situation. You're afraid that you'll hurt your mother if you follow your father's suggestion. But you're also afraid to hurt your father if you follow your mother's suggestion. But it can get worse, because you can hurt both of them if you simply ignore their suggestions!

Thus, you decide that you'll try to follow their suggestions in the better way that you can. So, you realize that the "Paris-Lisboa-London" order is the one which better satisfies both your mother and your father. Afterwards you can say that you could not visit Madrid, even though you would've liked it very much.

If your father had suggested the "London-Paris-Lisboa-Madrid" order, then you would have two orders, "Paris-Lisboa" and "Paris-Madrid", that would better satisfy both of your parents' suggestions. In this case, you could only visit 2 cities.

You want to avoid problems like this one in the future. And what if their travel suggestions were bigger? Probably you would not find the better way very easy. So, you decided to write a program to help you in this task. You'll represent each city by one character, using uppercase letters, lowercase letters, digits and the space. Thus, you can have at most 63 different cities to visit. But it's possible that you'll visit some city more than once.

If you represent Paris with 'a', Madrid with 'b', Lisboa with 'c' and London with 'd', then your mother's suggestion would be 'abcd' and your father's suggestion would be 'acdb' (or 'dacb', in the second example).

The program will read two travel sequences and it must answer how many cities you can travel to such that you'll satisfy both of your parents and its maximum.

## Input

The input will consist on an arbitrary number of city sequence pairs. The end of input occurs when the first sequence starts with an '#' character (without the quotes). Your program should not process this case. Each travel sequence will be on a line alone and will be formed by legal characters (as defined above). All travel sequences will appear in a single line and will have at most 100 cities.

## Output

For each sequence pair, you must print the following message in a line alone:

Case #*d*: you can visit at most *K* cities.

Where *d* stands for the test case number (starting from 1) and *K* is the maximum number of cities you can visit such that you'll satisfy both your father's suggestion and your mother's suggestion.

## Sample Input

```
abcd
acdb
abcd
dacb
#
```

## Sample Output

```
Case #1: you can visit at most 3 cities.
Case #2: you can visit at most 2 cities.
```