

11739 Giving Candies

Two siblings are trying to figure out a way of sharing a packet of candy. They would like to divide it as equally as possible. The problem is that they are all attached into one continuous strip. Sharing it would involve breaking the strip into smaller pieces containing one or more individual candies. The candies themselves are not all the same, thus in a totally fair distribution, each sibling must get an equal number of the same type of candy.

But, being small children, they also have some additional (and more annoying) demands. Each sibling must get exactly one whole segment broken off from the main strip, and that one segment must contain all the candies that form his share. The child will not accept more than one segment for his or her share. What is more, the order of the candies from left to right in one child's segment must be exactly the same as that in other child's (and rotating the strip is not allowed), otherwise they have difficulty verifying that they have got a fair distribution. So both segments must be exactly the same. To make matters worse, there are certain types of candies that neither sibling wants. Those candies cannot be in the segment handed to the children, because they consider any segment containing those candies to be ruined.

Your job is, given a large strip of candy and a list of 'bad' candies, identify the largest contiguous segment that occurs at least twice (the two occurrences cannot overlap, obviously) and does not have any bad candies in it. Any left over candy is discarded. If there are no such segments in the strip, the whole strip is discarded and the children get nothing.



Input

The first line of input contains a single integer N ($N \leq 50$). N cases follow. Each case consists of exactly two lines. The first line is a non-empty string ($1 \leq \text{length} \leq 1000$) describing one large strip of candy. Each candy is given a unique character that is either a lower case or upper case English letter ('a' to 'z' or 'A' to 'Z', both ranges inclusive). The second line is a string of characters corresponding to 'bad' candies. Like the first line, this string consists of upper or lower case English letters. But in some cases there may not be any bad candies (i.e. there are no candy types that the children don't like), and the second line will be an empty line instead. Furthermore, each candy can occur only once in the list of bad candies (no duplicates).

Each candy corresponds to exactly one unique letter, so upper case and lower case forms of the same character denote different candies (i.e. 'a' and 'A' are not the same candy).

Output

Your program should print exactly N lines of output, one for each input case.

Each line should be of the form 'Case #X: Y', where X is the case number and Y is the size of the largest segment that at least two non-overlapping occurrences and has no bad candies in it. If there is no such segment, Y is zero.

Sample Input

ABADZEDGBADEZ

ZEG

CaNdY

baaaa

a

aaaab

b

MnMnMn

Aa

aAaA

fghFGH

Sample Output

Case #1: 3

Case #2: 0

Case #3: 0

Case #4: 2

Case #5: 2

Case #6: 2