Big Dragons

Again our country is being attacked by the Enemy country. The soldiers are coming in a line. There are different kinds of soldiers. Now they have Big Dragons. But again our spy informed us how Big Dragons look like. Now you have to find how many Big Dragons they can have so that we can prepare ourselves.

██████ ██████ █████ █████ ███ ██████ ████ ██████ ██████████ ██ ████ ███ ██████ █████ █████ ██████ █████ ████ ███ ██████ █████ ███ ████ ██ █████ ████ ████ ██ █████ ███ ██ ██ ██████ ████ ████ ███ ██████ ███ ██████ █████ █████ █████ ██████ ██ ██████ █████ ████ ████ ███ ████ ███

Input

The first line of input file contains the number of test cases T (T ≤25). Each test case is composed of two lines. The first line describes the line of soldiers using a string (a sequence of nonempty alphabetical character). Now a Big Dragon can contain more than one character (that's why it is big). There will be no space between two consecutive characters. The second line contains another string describing a Big Dragon. Now you have to find the maximum possible number of Big Dragons in the line of soldiers, that is, number of all occurrence of the 2nd string in the previous string. The maximum number of characters will not be greater than 100. The problem is case sensitive. Two Big Dragons may overlap.

██████ ██████ █████ █████ █████ ██████ █████ ███ ███ ,████ ██████ T(T≤25) ████ ██████████ █████ ███ ██ █ ₂ ██ ████ █████ █████ █████ █████ ██████ ██████ ,████ ███████ (████ ████████ ████████████ ████████████ █████████) ██████ █████ █████ ██ ████████ ██████ ███ ████ ██ ██████ ██████ ████████████ ███ ████ (███ ████ ██ ██████) ███ ██████ ██████ █████ ██████ ██ ██████ ███ ████ ███ ████ ███ ,███ ████ ████ ██████ ‑█ ████ ██████████ ██████ ███ ████ ███ ████ ███ █████ ████ 100 ██ ████ ███ ██ ██████████ ███ █████████ █████ ██████ █████████ ████ ████

Output

For each case, print a line like "Case X: N", without the quotes, where X is the number of test case starting from 1 and N is the number of Big Dragons. Check sample input and output sections for more details.

██████████ ███ ██ ████ ████ ████ "Case X: N" ██████ ████ ███ ,████ ████ ,██████ X ██ ₁ ████ █████ ████ █████ ███ ██ ██████ ,██ N ██ ██ ██████ ██ ████

Example

Input:
5
ssddDddssddDSdsdsd
ddD
adnalkdjfkasdjkaa
a
aaknklopmnhlakdjsfksdakldjfalkdfla
la
AAAAAA
AA
AbAbAbAb
bAb

Output:
Case 1: 2
Case 2: 5
Case 3: 2
Case 4: 5
Case 5: 3

For case 5: we start matching from the beginning. At first 'bAb' matched with 'AbAbAbAb'. Next match will be 'AbAbAbAb'. Third match is ''AbAbAbAb'