

Have you tried to modify code by changing some names of the identifiers? For example, the following codes can be changed to each other.

<pre>int i, j; i = 3; j = i + 1;</pre>	<pre>int a, i; a = 3; i = a + 1;</pre>
--	--

However, “int” cannot be changed, because it’s a keyword, not an identifier. Similarly, operators like “=” or “+” can’t be changed, either.

To simplify the I/O, we use one upper-case letter to denote one kind of token that cannot be changed, and use one lower-case letter to denote an identifier whose name can be changed. However, two different lower-case letters cannot be changed to the same letter.

For example, if we use the following table:

Token	int	,	;	=	3	+	1
letter	A	B	C	D	E	F	G

Then the first program can be written as **AiBjCiDECjDiFGC**, the second one is **AaBiCaDECiDaFGC**.

Given a snippet (a small piece of code), can you find all its occurrences (possibly overlapping) in a large program?

Input

The first line contains the number of test cases T ($T \leq 100$). Each test case contains two lines, the first line is the program to be searched in, and the second line is the snippet. Both lines will contain letters only. There will be at most 10^6 characters in either string. The total input size will be less than 10M bytes.

Output

For each test case, print the number of occurrences of the snippet in the program.

Explanation:

In the first sample, ‘ccd’ and ‘dde’ are both “changed” version of ‘aab’. ‘def’ is not counted because ‘a’ cannot be changed into both ‘d’ and ‘e’. In the second sample, ‘DEabc’ is not counted because ‘AB’ cannot be changed into ‘DE’. ‘ABcaa’ is not counted because ‘b’ and ‘c’ cannot be both changed to ‘a’.

Sample Input

```
2
ccddef
aab
ABdefDEabcABcaa
ABabc
```

Sample Output

```
Case 1: 2
Case 2: 1
```