**COUNT SUBSEQUENCES**

A common problem is to determine whether a string $S$ contains a string $T$ as a subsequence. That is, is it possible to remove some of the characters of $S$ and end up with $T$? For example, if $S=$ABCABC and $T=$AC, then the answer is “yes”. Here we consider a related question: in how many distinct ways is $T$ a subsequence of $S$? Using $S=$ABCABC and $T=$AC again, the answer should be 3: _ABCABC_, _ABCABC_, and _ABCABC_.

**input**
The first line of input contains an integer $C$ (≤ 1000), where $C$ is the number of test cases. The next $C$ lines each contain 2 space-separated strings, $S$ $T$. You can assume that $S$ and $T$ consist of lower case characters (a–z) and that $1 \leq \text{len}(T) \leq \text{len}(S) \leq 33$.

**output**
The output $C$ integers on separate lines, each indicating the number of distinct ways in which $T$ can be a subsequence of $S$.

<table>
<thead>
<tr>
<th>Sample Input</th>
<th>Sample Output</th>
</tr>
</thead>
</table>
| 4
abcabc ac
babgbag bag
aaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaa
abc cba | 3
5
184756
0 |

Note: The output in the third case can be explained as “20 choose 10”=$C(20,10)=184756$. 