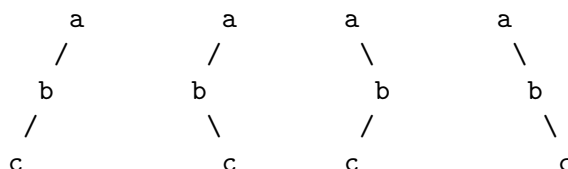


## 2584 Pre-Post-erous!

We are all familiar with pre-order, in-order and post-order traversals of binary trees. A common problem in data structure classes is to find the pre-order traversal of a binary tree when given the in-order and post-order traversals. Alternatively, you can find the post-order traversal when given the in-order and pre-order. However, in general you cannot determine the in-order traversal of a tree when given its pre-order and post-order traversals. Consider the four binary trees below:



All of these trees have the same pre-order and post-order traversals. This phenomenon is not restricted to binary trees, but holds for general  $m$ -ary trees as well.

### Input

Input will consist of multiple problem instances. Each instance will consist of a line of the form

$m s_1 s_2$

indicating that the trees are  $m$ -ary trees,  $s_1$  is the pre-order traversal and  $s_2$  is the post-order traversal.

All traversal strings will consist of lowercase alphabetic characters. For all input instances,  $1 \leq m \leq 20$  and the length of  $s_1$  and  $s_2$  will be between 1 and 26 inclusive. If the length of  $s_1$  is  $k$  (which is the same as the length of  $s_2$ , of course), the first  $k$  letters of the alphabet will be used in the strings. An input line of '0' will terminate the input.

### Output

For each problem instance, you should output one line containing the number of possible trees which would result in the pre-order and post-order traversals for the instance. All output values will be within the range of a 32-bit signed integer. For each problem instance, you are guaranteed that there is at least one tree with the given pre-order and post-order traversals.

### Sample Input

```
2 abc cba
2 abc bca
10 abc bca
13 abejkcfghid jkebfghicda
0
```

### Sample Output

```
4
1
45
207352860
```