Inversion Sort

You have just bought an old fashioned jukebox that can hold 10 music albums. Albums are maintained as a sequence, each album represented by a unique lowercase letter between “a” and “j”, inclusive. The jukebox allows you to select a subsequence of contiguous albums and a mechanical arm inverts that part of the sequence. For instance, if the current sequence is “abcdefghij” and you select the subsequence “bcd”, the result of the inversion would be “adcbefghij”. Soon you notice that it is possible to get the albums into any desired order using simply inversions. However, you are interested in doing so with the minimum number of operations. Given the current order and a desired order of the 10 music albums, find the minimum number of inversion operations needed to obtain the desired order.

Input

The input contains several test cases, each one described in a single line. The line contains two strings C and D separated by a single space, representing the current and desired orders of the music albums, respectively. Each of the strings has exactly 10 characters and contains the characters of “abcdefghij” in some order. The last line of the input contains two asterisks (“*”) separated by a single space and should not be processed as a test case.

Output

For each test case output a single line with an integer representing the minimum number of inversions needed to transform the current order given by C, into the desired order given by D.

Example

Input:

```
abcdefghij adcbefghij
abcdefghij abcdefghij
bcdaefghji beagfcdhji
* *
```

Output:

```
1
0
2
```