2250   Encryption Scheme

Most text encryption schemes use a secret key string to convert the plain text to the enciphered text in some way. A novel method being tested by the Australian Security Service consists of a transformation of a key string K into a target string P using block moves. Each *block move* is of the form $copy(start, length)$, where $start$ indicates a position in K and $length$ is the number of characters to be copied from K to P. Since the idea is to eventually transmit only the block moves, the principle is to use as few block moves as possible. For example if:

K: `abaabba`
P: `aaabbbabbbaaa`

Assuming that here string positions start with 1, two shortest block move sequences would be:

$copy(3, 2); copy(4, 3); copy(2, 2); copy(5, 2); copy(2, 3); copy(1, 1)$

or

$copy(7, 1); copy(3, 3); copy(5, 2); copy(4, 2); copy(5, 3); copy(3, 2)$

The actual shortest block move sequences are not unique but the minimum number is, 6 in this case. If the moves are now transmitted, then it is possible to construct the plaintext message P from the key string K.

The Australian Security Service is now automating this procedure, so given K and P they need to count the minimum number of block moves from K to P. To make things simple at the beginning, they are considering strings comprised of lowercase letters and digits. The set of characters within string P is a subset of the set of characters of the key string K.

You are to help the Australian Security Service by writing a program to get two strings K and P as above, and print the minimum number of block moves from K to P.

Input

Your code will be tested with a sequence of lines. Odd lines are to be used as the key strings K, and even lines to be used as target strings P.

The input will be terminated by a '#' by itself in the place of a K string.

Assume that each of K and P is made up of 1 to 120 characters (K is allowed to be longer than P).

Output

The output will consist solely of the minimum number of block moves for each pair, on a line by itself.

COMMENTS: The first sample is discussed above. Here follows a minimal sequence of block moves for the second sample:

$copy(4, 1); copy(4, 1); copy(4, 1); copy(3, 1);$
$copy(2, 1); copy(2, 1); copy(2, 2);$
$copy(1, 1); copy(1, 1); copy(1, 1)$

Sample Input

```
abaabba
aaabbbabbbaaa
xy0z
zzz0yyy0xxx
#
```

Sample Output

```
6
10
```