

Periodic permutation is a simple encryption technique which involves choosing a period, k , and a permutation of the first k numbers. To encrypt a message, split the message into groups of k characters (padding if necessary) and apply the given permutation. Decryption involves taking groups of k characters and performing the inverse permutation. Thus for $k = 4$, a permutation could be 2431. This would encrypt 'Mary' to 'yMra' and 'Maryan' to 'yMra?a?n'. Once one knows the permutation, one can apply its inverse to other encrypted messages (cyphertext) to recover the original text (plaintext).

Write a program that will read (*plaintext*, *cyphertext1*, *cyphertext2*) triples, and for each (*plaintext*, *cyphertext1*) pair determine whether or not a periodic permutation encryption method has been used. If it has, determine the value of k and the permutation function and apply the reverse permutation to *cyphertext2* to recover the corresponding plaintext.

Input

Input will consist of a series of (*plaintext*, *cyphertext1*, *cyphertext2*) triples. Lines will be no more than 80 characters long. The first two strings (of length n) represent the first n characters of the *plaintext* and *cyphertext*. There is no implication that n is a multiple of k . The file will be terminated by a line consisting of a single '#'.

Output

Output will consist of a series of lines, one for each triple in the input. If a permutation cycle has been found, apply the inverse permutation to *cyphertext2*, padding it if necessary with '?'. If no periodic permutation can be found (with period less than or equal to the length of the *plaintext* and *cyphertext1* strings) that transforms the *plaintext* into the *cyphertext*, then print *cyphertext2* unchanged. If more than one periodic permutation could have mapped the plain text to the *cyphertext1*, then apply the periodic permutation that has the smallest value for k . There will never be more than one shortest permutation function that matches the data.

Sample Input

```
Mary had a little lamb!!
aMyrh daa l tilt ealbm!!
hTsii s aetts
Foobar
blargg
No cycle
abc
bca
abcd
#
```

Sample Output

```
This is a test
No cycle
cab?d?
```