5 Forgotten Key

John is in a tough spot—he's written an encryption algorithm but has forgotten the key! Fortunately, he remembers the encrypted message.

Your task is to help John reverse-engineer his own algorithm and recover the key. John's encryption algorithm assigns unique identifiers to each letter based on their position in the message.

The key will be represented as a single string, where each character's identifier corresponds to the index of its first occurrence. If a letter appears multiple times, the key will reference the next unused occurrence of that letter.

5.1 Input

- The input provided will be the encrypted message and its decrypted version.
- Make sure to handle capital letters as distinct from their lowercase counterparts, ignore spaces, and account for any numbers in the input.

5.2 Output

- The output will be a string of numbers where each number represents a letter based on the position in the message

5.3 Sample Input/Output

SAMPLE INPUT 1	**SAMPLE OUTPUT 1**
test etst	1023

SAMPLE INPUT 2
Where Ever Could John Have Lost His Password dhehireveWrCssolJPoHEuaveosntsLaHword
SAMPLE OUTPUT 2
131216273467089242810121429151851119202123311725322230263334353 6

SAMPLE INPUT 3
this is a Test to Show Cas3s Tha1 ma7 not be considered ismbhiosnonesetdoios3etaS1se7TcawdsrattThaCh
SAMPLE OUTPUT 3
232632141252915368933039303835192040106132521422873418164337412411312214271723