Append and Delete

You have two strings of lowercase English letters. You can perform two types of operations on the first string:

1. Append a lowercase English letter to the end of the string.
2. Delete the last character of the string. Performing this operation on an empty string results in an empty string.

Given an integer, k, and two strings, s and t, determine whether or not you can convert s to t by performing exactly k of the above operations on s. If it's possible, print `Yes`. Otherwise, print `No`.

Example. $s = [a, b, c]$
$t = [d, e, f]$
$k = 6$

To convert s to t, we first delete all of the characters in 3 moves. Next we add each of the characters of t in order. On the 6^{th} move, you will have the matching string. Return `Yes`.

If there were more moves available, they could have been eliminated by performing multiple deletions on an empty string. If there were fewer than 6 moves, we would not have succeeded in creating the new string.

Function Description

Complete the appendAndDelete function in the editor below. It should return a string, either `Yes` or `No`.

appendAndDelete has the following parameter(s):

- string s: the initial string
- string t: the desired string
- int k: the exact number of operations that must be performed

Returns

- string: either `Yes` or `No`

Input Format

The first line contains a string s, the initial string.
The second line contains a string t, the desired final string.
The third line contains an integer k, the number of operations.

Constraints

- $1 \leq |s| \leq 100$
- $1 \leq |t| \leq 100$
- $1 \leq k \leq 100$
- s and t consist of lowercase English letters, $\textbf{ascii}[a\text{-}z]$.

Sample Input 0

```
hackerhappy
hackerrank
```

9

Sample Output 0

Yes

Explanation 0

We perform **5** delete operations to reduce string ***s*** to `hacker`. Next, we perform **4** append operations (i.e., `r`, a, n, and k), to get `hackerrank`. Because we were able to convert ***s*** to ***t*** by performing exactly $k = 9$ operations, we return `Yes`.

Sample Input 1

aba
aba
7

Sample Output 1

Yes

Explanation 1

We perform **4** delete operations to reduce string ***s*** to the empty string. Recall that though the string will be empty after **3** deletions, we can still perform a delete operation on an empty string to get the empty string. Next, we perform **3** append operations (i.e., a, b, and a). Because we were able to convert ***s*** to ***t*** by performing exactly $k = 7$ operations, we return `Yes`.

Sample Input 2

ashley
ash
2

Sample Output 2

No

Explanation 2

To convert `ashley` to `ash` a minimum of **3** steps are needed. Hence we print `No` as answer.

Change Theme Language Python 3

```
#!/bin/python3

import math
import os
import random
import re
import sys

#
# Complete the 'appendAndDelete' function below.
#
# The function is expected to return a STRING.
# The function accepts following parameters:
```

```
#  1. STRING s
#  2. STRING t
#  3. INTEGER k
#

def appendAndDelete(s, t, k):
    # Write your code here
    common_length = 0
    min_len = min(len(s), len(t))

    # Find the length of the common prefix
    for i in range(min_len):
        if s[i] == t[i]:
            common_length += 1
        else:
            break

    total_operations = (len(s) - common_length) + (len(t) - common_length)

    # Case 1: We can perform the operations directly
    if k >= total_operations and (k - total_operations) % 2 == 0:
```

EMACS Line: 56 Col: 1

Upload Code as File | Test against custom input | Run Code | Submit Code

You have earned 20.00 points!

You are now 1174.8 points away from the 6th star for your problem solving badge.

13% 1025.2/2200

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

Test case 0

Test case 1

Test case 2

Test case 3

Test case 4

Test case 5

Test case 6

Compiler Message

```
Success
```

Input (stdin) Download

```
hackerhappy
hackerrank
9
```

Expected Output Download

```
Yes
```