Problem: Append and Delete

You have a string, , of lowercase English alphabetic letters. You can perform two types of operations on :

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

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

Input Format

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

Constraints

-
-
-
- and consist of lowercase English alphabetic letters.

Output Format

Print `Yes` if you can obtain string by performing exactly operations on ; otherwise, print `No`.

Sample Input 0

```
hackerhappy
hackerrank
9
```

Sample Output 0

```
Yes
```

Explanation 0

We perform delete operations to reduce string to `hacker`. Next, we perform append operations (i.e., `r`, `a`, `n`, and `k`), to get `hackerrank`. Because we were able to convert to by performing exactly operations, we print `Yes`.

Sample Input 1

```
aba
aba
```

```
7
```

Sample Output 1

```
Yes
```

Explanation 1

We perform delete operations to reduce string to the empty string (recall that, though the string will be empty after deletions, we can still perform a delete operation on an empty string to get the empty string). Next, we perform append operations (i.e., `a`, `b`, and `a`). Because we were able to convert to by performing exactly operations, we print `Yes`.

Solution

```
int main()
   {
      string one, two;
      int operations, commonLength=0;

      cin>>one >>two >>operations;
      int length1=one.size();
      int length2=two.size();


      for(int i=0; i<one.size(); i++)
         {
            if(one[i]==two[i])
               { commonLength+=1;
               }
            else
            { break; }
         }

      if( length1+length2 - (2*commonLength) > operations )
         { cout<<"No"; }
      else if((length1+length2 - (2*commonLength))%2 == operations%2)
         {cout<<"Yes";}
```

```
    else if(length1+length2+1<=operations)
        {cout<<"Yes";}
    else
        {cout<<"No";}

return 0;
}
```

- Anshul Aggarwal