**Question Title**

3. Consider a string containing lowercase characters from the Portuguese alphabet. You can perform two types of operations on this string:  
1. **Concat** a lowercase character from the Portuguese alphabet at the end of the string.  
2. **Remove** the last character from the string. If the string is empty, it will remain empty.  
  
Given an integer ***k***and two strings ***s***and ***t***, determine if you can convert ***s***to ***t***using exactly ***k***operations described above on ***s***. If possible, the program prints '*yes*', otherwise it prints '*no*'.  
  
For example, string s = [a, b, c] and string t = [d, e, f]. The number of moves k = 6. To convert s to t, we first remove all characters using 3 moves. Then we concatenate each of the characters of t in the order. In the sixth move, you will have the expected string s. If more movements are available than necessary, they can be eliminated by performing multiple removals on an empty string. If there are fewer movements, it would not be possible to create the new string.  
  
Develop a **C/C++** or **Java** language program that implements and uses the **ConcatRemove (s, t, k)** function. It should return the results '*yes*' or '*no*'.  
The function has the following parameters:  
**s**: *starting string*  
**t**: *desired string*  
**k**: *an integer that represents the number of operations*  
  
**Input format**  
*The first line contains the string s, the starting string.*  
*The second line contains the string t, the desired string.*  
*The third line contains an integer k, the number of operations.*  
  
**Constrains**  
*a) 1 <= | s | <= 100*  
*b) 1 <= | t | <= 100*  
*c) 1 <= k <= 100*  
*d) s and t consists of lowercase letters of the Portuguese alphabet, ascii [a-z]*  
  
**Output format**  
Print '*yes*' if you can get the string t by executing exactly k operations on the string s, and print '*no*' otherwise.  
  
**Example 1**  
bla bla bla bla  
blablabcde  
8  
yes  
  
*Explanation*  
It took 5 operations to reduce the string "blablablabla" to "blablab". Then, 3 concatenation operations (c, d, e) were needed to obtain the string "blablabcde".  
In this way, as it was possible to convert s to t using exactly k = 8 operations, the program printed "yes".  
  
**Example 2**  
tab  
tab  
7  
yes  
  
*Explanation*  
It took 4 operations to reduce the string s (remember that a removal operation on an empty string, keeps the string empty). Then, 3 concatenation operations were performed (a, b, a). Since it was possible to convert s to t using exactly k = 7 operations, the prog

**Question Title**

4. Write an Unit Test using C/C++ or Java for the function ***ConcatRemove(s,t,k)***