unstop | Backspace String Compare

Problem | Solution | Submissions

Problem Statement

Alice and Bob both are friends and are playing with each other. Alice receives a new typewriter and is very excited to type the words on it. He challenged Bob to write the same word as his on the typewriter. Both are kids and are making some mistakes in typing and are making use of the '#' key on a typewriter to delete the last character printed on it.

Note that after backspacing an empty text, the text will continue empty.

Input Format

- The first line contains a string 'Bob' of Bob.
- The second line contains a string 'Alice' of Alice.

Output Format

- The first line contains 'YES' if Alice is able to print the exact words as Bob , otherwise 'NO'.

Constraints

- 1 <= Bob.length, Alice.length <= 100000
- Bob and Alice only contain lowercase letters and '#' characters.

Sample Testcase 0

Testcase Input

```
ab#c
ad#c
```

Testcase Output

```
YES
```

Explanation

Here ,
The actual typed string by Bob : 'ac'
The actual typed string by Alice : 'ac'
Hence , they matched.

Sample Testcase 1

Testcase Input

```
a#c
b
```

Testcase Output

```
NO
```

Explanation

Here ,
The actual typed string by Bob : 'c'
The actual typed string by Alice : 'b'
Hence , they do not matched.

```
#include <iostream>
#include <string>

using namespace std;

bool canTypeExactWords(string bob, string alice) {
    int i = bob.length() - 1;
    int j = alice.length() - 1;

    while (i >= 0 && j >= 0) {
        int backspaces = 0;

        while (i >= 0 && (backspaces > 0 || bob[i] == '#')) {
            if (bob[i] == '#') {
                backspaces++;
            } else {
                backspaces--;
            }
            i--;
        }

        backspaces = 0;

        while (j >= 0 && (backspaces > 0 || alice[j] == '#')) {
            if (alice[j] == '#') {
                backspaces++;
            } else {
                backspaces--;
            }
            j--;
        }

        if (i >= 0 && j >= 0 && bob[i] != alice[j]) {
```

```
        if (i >= 0 && j >= 0 && bob[i] != alice[j]) {
            return false;
        }

        if ((i >= 0) != (j >= 0)) {
            return false;
        }

        i--;
        j--;
    }

    while (i >= 0 && (bob[i] == '#' || bob[i] == ' ')) {
        i--;
    }

    while (j >= 0 && (alice[j] == '#' || alice[j] == ' ')) {
        j--;
    }

    return (i < 0) && (j < 0);
}

int main() {
    string bob, alice;
    cin >> bob >> alice;

    if (canTypeExactWords(bob, alice)) {
        cout << "YES" << endl;
    } else {
        cout << "NO" << endl;
    }

    return 0;
}
```

This C++ code checks if Alice can type the exact same words as Bob by simulating their typing process, taking into account backspaces ('#'). It iterates through the strings from right to left, considering backspaces and comparing characters. If they match, it proceeds; otherwise, it returns "NO." If the loop completes successfully, it means Alice can type the same words, and it returns "YES."