23rd Jan 2023

First:-

Geeks And The String : : Medium

Our geek loves to play with strings, Currently, he is trying to reduce the size of a string by recursively removing all the consecutive duplicate pairs. In other words, He can apply the below operations any number of times.

- Remove all the consecutive duplicate pairs and concatenate the remaining string to replace the original string.

Your task is to find the string with minimum length after applying the above operations.

Note: If the string length become zero after applying operations, return "-1" as a string.

Example 1:

```
Input:
aaabbaaccd
Output:
ad
Explanation:
Remove (aa)abbaaccd =>abbaaccd
Remove a(bb)aaccd => aaaccd
Remove (aa)accd => accd
Remove a(cc)d => ad
```

Example 2:

```
Input:
aaaa
Output:
```

-: DONE FOR THE DAY :-

```
Empty String
Explanation:
Remove (aa)aa => aa
Again removing pair of duplicates then (aa)
will be removed and we will get 'Empty String'.
```

Your Task:

This is a **function** problem. You only need to **complete** the function **removePair()** that takes a **string** as a **parameter** and **returns** the **modified string**. Return an empty string if the whole string is deleted.

Expected Time Complexity: O(N)

Expected Auxiliary Space: O(N)

Constraints:

1 <= |str| <= 10^4

CODE SECTION:-

```cpp
class Solution
{
public:
    string removePair(string s)
    {
        stack<char> st;
        string ans = "";
        for (auto a : s)
        {
            if (!st.empty() && st.top() == a)
                st.pop();
            else
                st.push(a);
        }
        while (!st.empty())
        {
            ans = st.top() + ans;
            st.pop();
        }
        return ans == "" ? "-1" : ans;
    }
};
```

-: DONE FOR THE DAY :-