Dinesh and Parsers

locked

by pruthvishalcodi1

Editorial by PruthvishE

A regular bracket sequence is defined as follows:

- S="" is regular.
- S="<" + S1 + ">" is regular, if S1 is regular.
- S=S1 concat S2 is regular, if S1 and S2 are regular.

If S is regular bracket sequence, for any i, number of closing brackets in S[0,i] should not exceed number of opening brackets. Also, if number of opening brackets is equal to number of closing brackets in S[0,i], S[0,i] is a regular bracket sequence.

Statistics

Difficulty: Medium
Time Complexity: O(N)
Required Knowledge: stacks
Publish Date: Jun 19 2019

Set by PruthvishE

Problem Setter's code :

```
#include <string>
#include <iostream>
using namespace std;

int main()
{
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);
    string exp;
    int test;
    int sum=0;
    int d,b;
    cin >> test;
    while(test--){
        b=0;
        cin >> exp;
        const int len = exp.size();
        d=0;
        sum+=exp.size();
        for(int i=0;i<len;i++){
        if(exp.substr(i,1)==">")
            d--;
        else
            d++;
        if(d==0)
            b=i+1;
        else if(d<0)
            break;
        }
        cout << b << endl;
    }
return 0;
}
```

Tested by PruthvishE

Problem Tester's code :

```
for _ in range(int(input())):
    s = input()
    st = []
    i = 0
    ans = 0
    while i<len(s):
        if s[i] == '<':
            st.append('<')
        elif st:
            st.pop()
        else:
            break
        i += 1
        if not st:
            ans = i
    print(ans)
```