

UVA673 平衡的括号

题目链接

<https://www.luogu.com.cn/problem/UVA673>

题目描述

输入一个包含 “()” 和 “[]” 的括号序列，判断是否合法。 具体规则：

空串合法；
如果A和B合法，那么AB合法；
如果A合法(A)和[A]都合法

输入样例

```
3
([])
(([])))
([() [] (]) ()
```

输出样例

```
Yes
No
Yes
```

解析

本题是一道很经典的用栈来解决的题目

每次入栈的时候判断栈的顶部是不是这个符号的另一半，如果是就pop并且这个符号也不入栈，如果不是就入栈。最后判断栈是否为空，如果为空，说明正确，否则错误。

最坑的是空字符也合法，并且这个空字符可能出现在一个字符的中间这样的话也是合法的，比如([])

所以读入要注意，不能直接cin，这样一个单独的空行会被忽略掉导致答案错误。

编码

```
#include <bits/stdc++.h>

using namespace std;

int num;

string p;
```

```

stack<char> s;

//根据输入的括号，返回对应的前括号
char trans(char a) {
    if (a == ')') {
        return '(';
    }
    if (a == ']') {
        return '[';
    }
    if (a == '}') {
        return '{';
    }
    return '\0';
}

int main() {
    cin >> num;
    //为了在后面使用getline，在这里必须处理回车符号
    getchar();
    for (int i = 0; i < num; ++i) {
        //清空数据
        while (!s.empty()) {
            s.pop();
        }
        getline(cin, p);

        for (int j = 0; j < p.size(); ++j) {
            //栈空间为空，直接将字符串存入
            if (s.empty()) {
                s.push(p[j]);
                continue;
            }
            //匹配上了
            if (trans(p[j]) == s.top()) {
                s.pop();
            }
            //没有匹配成功，将数据存入
            else {
                s.push(p[j]);
            }
        }
        //判断栈中是否有残留的括号
        if (s.empty()) {
            cout << "Yes" << endl;
        } else {
            cout << "No" << endl;
        }
    }
}

```

```
}  
return 0;  
}
```

逻辑航线培优教育，信息学奥赛培训专家。

扫码添加作者获取更多内容。

