

1354: 括弧匹配检验

题目描述

假设表达式中允许包含两种括号：圆括号和方括号，其嵌套的顺序随意，如（[]（））或[（[] []）]等为正确的匹配，[（]）或（[]（）或（（）））均为错误的匹配。

现在的问题是，要求检验一个给定表达式中的括弧是否正确匹配？

输入一个只包含圆括号和方括号的字符串，判断字符串中的括号是否匹配，匹配就输出“OK”，不匹配就输出“Wrong”。输入一个字符串：[（[] []）]，输出：OK。

输入格式

输入仅一行字符（字符个数小于255）。

输出格式

匹配就输出“OK”，不匹配就输出“Wrong”。

输入样例

[（]）

输出样例

Wrong

解析

也是一道非常简单的匹配校验，我们只需要判断栈顶的数据是否能与右括号匹配即可，如果匹配不上，则存入栈中，最后我们只需要检查栈中是否有无数据即可。

编码

```
#include <bits/stdc++.h>
using namespace std;
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 '[';
    }
    return '\0';
}

int main() {
    getline(cin, p);
    for (int j = 0; j < p.size(); ++j) {
        //排除异常数据
        if (p[j] != '(' && p[j] != ')' && p[j] != '[' && p[j] != ']')
            continue;
    }
    //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 << "OK" << endl;
} else {
    cout << "Wrong" << 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。

