P1241 括号序列

题目描述

定义如下规则序列(字符串)：

1．空序列是规则序列；

2．如果S是规则序列，那么(S)和[S]也是规则序列；

3．如果A和B都是规则序列，那么AB也是规则序列。

例如，下面的字符串都是规则序列：

()，[]，(())，([])，()[]，()[()]

而以下几个则不是：

(，[，]，)(，())，([()

现在，给你一些由‘(’，‘)’，‘[’，‘]’构成的序列，你要做的，是补全该括号序列，即扫描一遍原序列，对每一个右括号，找到在它左边最靠近它的左括号匹配，如果没有就放弃。在以这种方式把原序列匹配完成后，把剩下的未匹配的括号补全。

输入格式

输入文件仅一行，全部由‘(’，‘)’，‘[’，‘]’组成，没有其他字符，长度不超过100。

输出格式

输出文件也仅有一行，全部由‘(’，‘)’，‘[’，‘]’组成，没有其他字符，把你补全后的规则序列输出即可。

输入样例

([()

输出样例

()[]()

解析

本题最难的点是对于“匹配”一词的理解，我们举几个例子，来加深理解。

1、([对应的结果为：()[]

2、([) 对应的结果为：()[]()

3、[([]) 对应的结果为：[]([])

总之，我们应当将输入中已经成对的括号，原封不动的保留下来，对于没有匹配成对的，我们需要将其进行补全。此外，成对的括号中间也必须是成对的括号，不能为单一括号，例如情况2。弄懂之后，我们直接模拟。

编码

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


using namespace std;

int book[105]; // 标记

int main() {
    //用于遍历两种括号的索引
    int i, j;
    //读入的括号串
    string s;
    cin >> s;
    for (i = 0; i < s.length(); i++) {
        if (s[i] == ')') { // 找到了右括号
          for (j = i - 1; j >= 0; j--) {
                //找到了没被匹配过的左括号，则匹配成功
              if (s[j] == '(' && book[j] == 0) {
                    //将两个位置都标记成为已使用
                  book[i] = book[j] = 1;
                    break;
                }
                    //两个小括号中间不能存在单独的方括号
              else if (s[j] == '[' && book[j] == 0) {
                    break;
                }
            }
            // 找不到左括号，不做任何操作
        }
            // 下面同理
        else if (s[i] == ']') {
            for (j = i - 1; j >= 0; j--) {
                if (s[j] == '[' and book[j] == 0) {
                    book[i] = book[j] = 1;
```

```
                    break;
                } else if (s[j] == '(' and book[j] == 0) break;
            }
        }
    }
    for (i = 0; i < s.length(); i++) {
        if (book[i] == 0) {
            //没有匹配，则自动补全
          if (s[i] == '(' || s[i] == ')') {
                cout << "()";
            } else {
                cout << "[]";
            }
        }
            //匹配成功则直接输出原字符串
       else {
            cout << s[i];
        }
    }
    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。