**数据结构**

**——实验报告书**



姓名：樊雨晨

学号：201632110111

班级：软件工程(中外)16 班

学院：数理与信息工程学院

日期：2017年05月10日

# **实验三 括号匹配判断算法**

## **问题**

假设在表达式中（［］（））或［（［ ］［ ］）］等为正确的格式，［（ ］）或（［（ ））或 (( )])均为不正确的格式。基于栈设计一个判断括号是否正确匹配的算法。

## **实验思路**

1. 首先是while（scanf！=EOF），可以多次输入。
2. 需要把左括号储存在栈中，先进后出的顺序。
3. 检测到右括号，将栈中的左括号退出来，进行比较。

## **实验步骤**

1. 首先讲括号的字符串输入到一个字符数组中。
2. 编写函数，判断当前的一个字符（左括号）是否在原先设置好的字符数组中有相同字符。
3. 若存在相同的，则将当前字符入栈，并进行下一个字符判断。
4. 若没有（当前字符为右括号），将栈顶的左括号进行退栈，进行比较，是否匹配。
5. 若匹配，则继续进行下一个字符判断，若不匹配，则返回no。
6. 在返回值时，对字符串赋值要使用strcpy，并且一开始的函数需要编写为字符指针的形式char \*EvaluateExpression(char\* MyExpression)，返回值Back需要为static char。在主函数中，定义char \* result，result=EvaluateExpression。然后Print result。

## **代码**

#**include**<stdio.h>

#**include**<string.h>

#**include** <iostream>

#**include** <assert.h>

#**include** <vector>

#**include** <deque>

#**include** <list>

#**include** <stack>

#**include** <string>

**using** **namespace** std;

**int** **In**(**char** Test, **char**\* TestOp);

**char** LOPSET[2] = { '[' , '(' };

**char** Expression[20];

**char** \***EvaluateExpression**(**char**\* MyExpression)

{

**static** **char** Back[4];

**typedef** stack<**char**> StackChar;

StackChar OPTR;

OPTR.push('#');

**char** \*c,Dr[2],theta;

c=MyExpression;

**while**(\*c != '#' || OPTR.top() != '#')

{

Dr[0] = \*c;

Dr[1] = '\0';

**if** (In(Dr[0], LOPSET))

{

OPTR.push(Dr[0]);

c++;

}

**else** **if**(!In(Dr[0], LOPSET))

{

theta=OPTR.top();

OPTR.pop();

**if**(theta=='[')

{

**if**(Dr[0]==']'){

c++;

}

**else**{

strcpy(Back,"no");

**return** Back;

}

}

**else** **if**(theta=='('){

**if**(Dr[0]==')'){

c++;

}

**else**{

strcpy(Back,"no");

**return** Back;

}

}

}

**else**

{

strcpy(Back,"no");

**return** Back;

}

}

strcpy(Back,"yes");

**return** Back;

}

**int** **In**(**char** Test, **char**\* TestOp) {

**bool** Find = false;

**for** (**int** i = 0; i< 2; i++) {

**if** (Test == TestOp[i]) Find = true;

}

**return** Find;

}

**int** **main**()

{

**int** n;

**char** \*result;

**while**(scanf("%s", Expression)!=EOF)

{

n=strlen(Expression);

Expression[n]='#';

result=EvaluateExpression(Expression);

printf("%s\n", result);

};

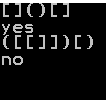
**return** 0;

}

## **测试**

一开始if条件中没有注意到写成了=，造成了一系列错误，应该为==。

## **结果**



## **总结**

1. 要注意字符串复制strcpy
2. If条件的符号
3. 返回类型为字符串的方法

有四种：<http://www.cnblogs.com/qingergege/p/6496683.html>