

数据结构实验报告

课程名称：数据结构	班级：计科 151	
姓名：刘禾子	学号：1500170082	指导教师：刘长云
实验序号：四	实验成绩：	
一、实验名称 数组和广义表		
二、实验目的及要求 1、熟悉广义表的层次结构，自顶向下识别并建立一个广义表的操作，熟悉其遍历逻辑。 2、熟悉广义表的存储特性		
三、实验环境 Visual C++		
四、实验内容 简单 LISP 算术表达式计算器 【问题描述】 设计一个简单的 LISP 算术表达式计算器。 简单 LISP 算术表达式（以下简称表达式）定义如下： (1) 一个 0..9 的整数；或者 (2) (运算符 表达式 表达式) 例如，6, (+45), (+(+25)8) 都是表达式，其值分别为 6, 9 和 15。 【基本要求】 实现 LISP 加法表达式的求值。 【测试数据】 6, (+45), (+(+25)8), (+2 (+58)), (+(+(+12) (+34)) (+(+56) (+78)))		
五、算法描述及实验步骤 算法描述： 利用栈依次从里向外计算括号内数的思想设计成一个简单的 LISP 算术表达式计算器。该程序首先建立一个栈，通过 PUSH 函数和 POP 函数的入栈，出栈功能实现栈中元素“先进后出，后进先出”的特性。然后动态建立一大大小为 K 的整型数组，把开辟的数组指针赋给 a，从而实现栈函数的构造过程。然后再建立栈的对象，再建立一个字符型数组 ch，然后向该数组输入表达式，最后进入表达式判断阶段。若 LISP 表达式为一操作数，则直接输出；若该表达式为运算符，则经过主函数的扫描计算，最终输出结果。 实验步骤： 首先要建立一个栈 V1，将所求表达式赋值给字符变量 ch。接下来就是从头扫描字符，这也是一个循环的过程。若 ch[i] 不是 ‘(’，则直接入栈，若是则跳向下下个字符，；再判断是不是 ‘(’，若不是，则将 ch[i] 入栈，跳向下一个字符		

再判断是不是 ‘(’，若还不是则入栈。这其中若是，若是则都要进行从头扫描。接下来进行一个总的判断 “!= ‘(’ && != ‘)’ ?”， 若成立，跳向下一字符，若不成立跳过本步；接下来判断是否为 ‘)’，若是则把入栈的两个元素出栈后相加再入栈，跳向下一字符。最后对此循环作一个判断，若 ch[i] 是 0 则接着进行循环，若不是则跳出循环输出栈顶元素，即为所求。

六、调试过程及实验结果

调试过程：

本程序在调试的过程中遇到很多问题，对于判断的冗余进行修剪之后简介了不少，再而就是调试过程中判断输入字符数组的时候 while(cin>>ch) 陷入空操作，还没对字符串读取就跳入了程序末尾，修改为 cin>>ch 之后程序恢复正常。

程序清单：

```
#include<iostream>
using namespace std;
int y = 0;
class stack
{
public:
    stack(int k = 100) {
        a = new int[k];
        big = k - 1;
        top = -1;
    };
    bool empty() {
        return top == -1;
    }
    bool full() {
        return top == big;
    }
    //压栈
    void push(int k) {
        if (!full())
        {
            a[++top] = k;
        }
        else
        {
            cout << "wrong push" << endl;
            return;
        }
    }
    //出栈
```

```

int pop() {
    if (!empty())
        return a[top--];
    else
        return 0; //使其参与运算
}

private:
    int big;
    int top;
    int *a;
};

void main() {
    stack v1;
    char ch[100];
    while (1)
    {
        cout << "简单的LISP算术表达式计算器，实现LISP加法表达式求值" << endl << endl;
        cout << "===正确格式的测试数据: 6, (+45), (+(+25) 8), (+2(+58)), " << endl;
        cout << " (+(+(+12) (+34)) (+(+56) (+78))) ===" << endl;
        cout << "结果: 6, 9, 15, 15, 36" << endl;
        cout << "===== " << endl;
        cout << "请输入表达式数据: " << endl;
        cin >> ch;
        int i = 0;
        while (ch[i] != 0)
        {
            if (ch[i] == '(')
            {
                i = i + 2;
                if (ch[i] == '(')
                    continue;
                v1.push(ch[i] - 48); //以0为基准
                i++;
                if (ch[i] == '(')
                    continue;
                v1.push(ch[i] - 48);
            }
            else if (ch[i] != ')')
                v1.push(ch[i] - 48);
            if (ch[i] != ')')
            {
                i++;
            }
            if (ch[i] == ')')
            {
                v1.push(v1.pop() + v1.pop());
            }
        }
    }
}

```

```

        i++;
    }
}

cout << "结果:    " << endl;
cout << v1.pop() << endl;

}
}

```

实验结果:

```

E:\Software\Visual Studio 2010\项目组\简单的LISP运算器\Debug\简单
简单的LISP算术表达式计算器，实现LISP加法表达式求值
===正确格式的测试数据: 6, (+45), (+(+25) 8), (+2(+58)),
(+((+12) (+34)) (+(+56) (+78)))===
结果: 6, 9, 15, 15, 36
=====
请输入表达式数据:
(+45)
结果:
9

```

```

E:\Software\Visual Studio 2010\项目组\简单的LISP运算器\Debug\简单
简单的LISP算术表达式计算器，实现LISP加法表达式求值
===正确格式的测试数据: 6, (+45), (+(+25) 8), (+2(+58)),
(+((+12) (+34)) (+(+56) (+78)))===
结果: 6, 9, 15, 15, 36
=====
请输入表达式数据:
(+(+25) 8)
结果:
15

```

```

简单的LISP算术表达式计算器，实现LISP加法表达式求值
请输入表达式数据:
(+2(+58))
结果:
15

```

```

简单的LISP算术表达式计算器，实现LISP加法表达式求值
请输入表达式数据:
(+((+12) (+34)) (+(+56) (+78)))
结果:
36

```

七、总结

该程序思想是只利用一个栈完成加法运算，后期可拓展为利用两个栈，一个存放操作数，一个存放运算符进行四则运算。在对栈的操作上还有欠缺，后期查阅资料得以弥补。

