数据结构实践课报告

题目：稀疏矩阵运算器

班级：17052317 姓名：於文卓 学号17061833 完成日期：2018年11月19日

1. **需求分析**
2. 问题描述

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。

1. 基本要求

实现LISP加法表达式的求值

1. 测试数据：

　6，(+45)，(+(+25)8)，(+2(+58))，(+(+(+12)(+34))(+(+56)(+78)))

1. **概要设计**
2. 数据结构（栈） 和 操作

|  |
| --- |
| typedef struct  {  SElemType \*base;  SElemType \*top;  int stacksize;  } SqStack; |

1. 基本操作

|  |
| --- |
| Status InitStack(SqStack \*s) //初始化栈  Status ClearStack(SqStack \*s) //清楚栈  Status StackEmpty(SqStack \*s) //是否是空栈  Status Destroy(SqStack \*s) //毁掉一个栈  Status GetTop(SqStack \*s,SElemType \*e) //拿到顶点元素  Status Push(SqStack\*s,SElemType e) //进栈  Status Pop(SqStack\*s,SElemType\*e) //退栈  Status StackTraverse(SqStack\*s, Status(\*visit)(SElemType)) //遍历  Status visit(SElemType c) //访问 |

1. **本程序包含3个模块：**

(1)主程序模块：

int main(){

输入矩阵；

选择操作；

处理命令；

输出结果；

return 0;

}

(2)储存模块——栈；

（3）运算单元模块——实现运算操作抽象数据类型；

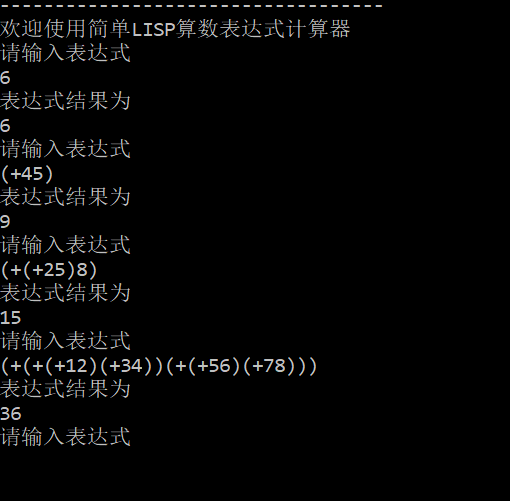
各模块之间的调用关系如下：

主程序模块 栈 运算单元模块

1. **运算单元模块部分实现代码**

|  |
| --- |
| **int f(char a[])**  **{**  **stack<int>int\_s1;//建立两个栈 一个数字栈**  **stack<char>char\_s2;//一个操作符栈**  **int num;**  **int i = 0;**  **char operate;**  **int num1;**  **int num2;**  **int res;**  **while(a[i] != '\0')**  **{**  **if (isdigit(a[i]))**  **{**  **num = (a[i] - '0');**  **int\_s1.push(num);**  **}**  **else**  **{**  **switch(a[i])**  **{**  **case '(': char\_s2.push(a[i]); break;**  **case '+': char\_s2.push(a[i]); break;**  **case ')':**  **operate = char\_s2.top();**  **char\_s2.pop();**  **char\_s2.pop();//退两次操作符栈栈**  **num1 = int\_s1.top();**  **int\_s1.pop();**  **num2 = int\_s1.top();**  **int\_s1.pop();//拿出两个数字**  **if (operate == '+')//做加法**  **{**  **res = num1 + num2;**  **int\_s1.push(res);//把结果压入栈**  **}**  **break;**  **}**  **}**  **i++;**  **}**  **res = int\_s1.top();**  **return res;**  **}** |

1. **算法分析和代码调试**
2. **当拿到的是‘）’时，注意字符栈要退栈两次**
3. **本程序用c++实现，调用其stack库**
4. **用户手册**
5. **本程序运行环境为win10操作系统，执行文件为LISP.exe**
6. **用户界面和测试数据如下**



1. **附件源代码**
2. /\*
3. \* @Author: TD21forever
4. \* @Date: 2018-11-19 20:30:55
5. \* @Last Modified by: TD21forever
6. \* @Last Modified time: 2018-11-26 23:25:33
7. \*/
8. #include<stdio.h>
9. #include<stdlib.h>
10. #include<iostream>
11. #include<stack>
12. using namespace std;
13. int f(char a[])
14. {
15. stack<int>int\_s1;//建立两个栈 一个数字栈
16. stack<char>char\_s2;//一个操作符栈
17. int num;
18. int i = 0;
19. char operate;
20. int num1;
21. int num2;
22. int res;
23. while(a[i] != '\0')
24. {
25. if (isdigit(a[i]))
26. {
27. num = (a[i] - '0');
28. int\_s1.push(num);
29. }
30. else
31. {
32. switch(a[i])
33. {
34. case '(': char\_s2.push(a[i]); break;
35. case '+': char\_s2.push(a[i]); break;
36. case ')':
37. operate = char\_s2.top();
38. char\_s2.pop();
39. char\_s2.pop();//退两次操作符栈栈
40. num1 = int\_s1.top();
41. int\_s1.pop();
42. num2 = int\_s1.top();
43. int\_s1.pop();//拿出两个数字
44. if (operate == '+')//做加法
45. {
46. res = num1 + num2;
47. int\_s1.push(res);//把结果压入栈
48. }
49. break;
50. }
51. }
52. i++;
53. }
54. res = int\_s1.top();
55. return res;
56. }
57. int main(int argc, char const \*argv[])
58. {
59. printf("-----------------------------------\n");
60. printf("欢迎使用简单LISP算数表达式计算器\n");
61. while(1)
62. {
63. char a[20];
64. printf("请输入表达式\n");
65. gets(a);
66. int res = f(a);
67. printf("表达式结果为\n");
68. printf("%d\n", res);
69. }
70. return 0;
71. }