

实验一 表达式运算器设计与实现

1. 问题的描述

(1) 设计目标

本实验旨在使用栈结构，设计和实现一个稀疏表达式运算器。该运算器可以完成的功能包括 1.波兰式计算；2.逆波兰式计算；3.中缀式计算。

(2) 输入数据

输入数据可以分为三部分。第一是用户通过菜单选择操作类型。第二是用户使用运算器功能时，比如求和、求导等输入的索引值与相关参数（比如求导次数）。第三是使用计算器时输入的表达式操作数和运算符。

(3) 输出结果

输出结果是计算得到的值。

2. 算法的描述

(1) 数据结构的描述

✓ 逻辑结构和存储结构

逻辑结构：使用链栈结构存储运算符和操作数，便于实现后进先出的操作。

存储结构：

结构体 SLNode 定义了链栈的节点，包含字符数据和指向下一个节点的指针。

LinkStack 为指向 SLNode 的指针类型，代表整个链栈。 主要变量和数组：

OPND：操作数栈，用于存储表达式中的数字。

OPTR：运算符栈，用于存储表达式中的运算符。

(2) 程序结构的描述

✓ 函数原型、功能和接口的描述

void showmenu(): 显示菜单，无返回值。 -void push(LinkStack &LS, char e): 将元素 e 压入栈 LS，无返回值。 void pop(LinkStack &LS, char &e): 从栈 LS 中弹出元素到 e，无返回值。 bool isOperator(char c): 判断字符 c 是否为运算符，返回布尔值。 int operate(int a, char op, int b): 执行运算符 op 在操作数 a 和 b 上的运算，返回结果。 int PolishCalcu(): 计算波兰式，返回计算结果。 int InversePolishCalcu(): 计算逆波兰式，返回计算结果。 char getTop(LinkStack LS): 获取栈顶元素，返回栈顶字符。 char precede(char op1, char op2): 比较运算符优先级，返回比较结果。 int EvaluateExpression(): 计算中缀表达式，返回计算结果。

3. 调试分析

(1) 测试数据及方法

测试数据的选择：选择了包含不同运算符和操作数的简单和复杂表达式进行测试。 程序调试中遇到的问题及解决方法：

问题：在处理多位数时，程序无法正确识别。 解决方法：修改程序逻辑，将多位数作为一个整体处理，而不是逐位处理。

问题：运算符优先级处理不当。 解决方法：完善 precede 函数，确保所有运算符的优先级关系正确。

（2）遇到的问题及解决方法

①在输入多位数时，发现运算会报错，经过分析，发现 `getchar()` 这个函数在接收变量时，只会一个一个字符的接收，因此我以为输入的多位数实际上被程序认为是输入了多个一位数。因此，我采用空格判定一位数和多位数，最终解决了这个问题，但是也对输入的格式有了要求。

②在写语句 `while ((c = getchar()) != '#' || getTop(OPTR) != '#')` 时，`(c = getchar()) != '#'` 省略了那个括号，写成了 `c = getchar() != '#'` 导致程序一直输出难以理解的结果，通过断点调试，发现程序执行路径与逻辑上的路径不一致，最终通过查资料，明白了问题所在，这是调试的伟大胜利！

③程序正常运行，有时会因为上一步的输入而导致程序发生难以预料的崩溃，通过网上搜索解决方法，写入 `cin.ignore();` 语句来清空程序的缓存区指令。

④最开始写中缀表达式函数时，没有对分支语句中栈顶元素优先级大于新读取的元素进行循环操作，导致优先级低的元素没有重复和栈顶元素进行比较，产生逻辑上的错误，通过测试代码，最终加入 `is_end` 参数来控制这个循环。

4.算法的时空分析

（1）时间复杂度

时间复杂度：对于波兰式和逆波兰式计算，时间复杂度为 $O(n)$ ，其中 n 是表达式中元素的数量。对于中缀表达式计算，时间复杂度也是 $O(n)$ ，但由于需要处理运算符优先级，实际运行时间可能更长。

（2）空间复杂度

空间复杂度：主要取决于输入表达式的长度，空间复杂度为 $O(n)$ 。

5.测试结果及分析

（1）波兰式

`+ 2 * 3 - 5 1`，期望结果：14

`++ 2 * 3 - 7 4 / 8 4`，期望结果：13

```

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
****0. 退出此系统****
****1. 波兰式计算****
****2. 逆波兰式计算****
****3. 中缀表达式计算***
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
请输入数字，代表你要执行的功能
1
请一次性输入前缀表达式，每次输入都摁一个空格确认你的输入（以#结束）：
数字之间没有空格就代表是同一个数，数字之间有空格就代表是不同的数
+ 2 * 3 - 5 1 #
结果是14

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
****0. 退出此系统****
****1. 波兰式计算****
****2. 逆波兰式计算****
****3. 中缀表达式计算***
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
请输入数字，代表你要执行的功能
1
请一次性输入前缀表达式，每次输入都摁一个空格确认你的输入（以#结束）：
数字之间没有空格就代表是同一个数，数字之间有空格就代表是不同的数
+ + 2 * 3 - 7 4 / 8 4 #
结果是13

```

(2) 逆波兰式

2 3 5 1 - * +, 期望结果: 14

9 3 1 - 3 * + 10 2 / +, 期望结果: 20

```

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
****0. 退出此系统****
****1. 波兰式计算****
****2. 逆波兰式计算****
****3. 中缀表达式计算***
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
请输入数字，代表你要执行的功能
2
请一次性输入后缀表达式，每次输入都摁一个空格确认你的输入（以#结束）：
数字之间没有空格就代表是同一个数，数字之间有空格就代表是不同的数
2 3 5 1 - * + #
结果是14

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
****0. 退出此系统****
****1. 波兰式计算****
****2. 逆波兰式计算****
****3. 中缀表达式计算***
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
请输入数字，代表你要执行的功能
2
请一次性输入后缀表达式，每次输入都摁一个空格确认你的输入（以#结束）：
数字之间没有空格就代表是同一个数，数字之间有空格就代表是不同的数
9 3 1 - 3 * + 10 2 / + #
结果是20

```

(3) 中缀表达式

4 + 2 * 3 - 10 / 5, 期望结果: 8

$(4+2) * 3 - 10 / 5$ ，期望结果：16

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
****0. 退出此系统****
****1. 波兰式计算****
****2. 逆波兰式计算****
****3. 中缀表达式计算***
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
请输入数字，代表你要执行的功能
3
请一次性输入中缀表达式，每次输入都回车或空格确认你的输入（以#结束）：
注意：
这里一定要两个##结尾！！！少摁了#就再摁一次#！！！
4 + 2 * 3 - 10 / 5 ##
结果是8

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
****0. 退出此系统****
****1. 波兰式计算****
****2. 逆波兰式计算****
****3. 中缀表达式计算***
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
请输入数字，代表你要执行的功能
3
请一次性输入中缀表达式，每次输入都回车或空格确认你的输入（以#结束）：
注意：
这里一定要两个##结尾！！！少摁了#就再摁一次#！！！
( 4 + 2 ) * 3 - 10 / 5 ##
结果是16
```

（4）健壮性检验

当输入表达式有非法字符时，如： $+ 2 A$ 、 $2 3 A -$ 、 $4 + a$ 时，程序会提示你输入了非法字符，结果作废，请重新输入正确的表达式。

```

请输入数字，代表你要执行的功能
1
请一次性输入前缀表达式，每次输入都摁一个空格确认你的输入（以#结束）：
数字之间没有空格就代表是同一个数，数字之间有空格就代表是不同的数
+ 2 A #
输入非法字符，结果作废，请重新输入
结果是2

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
****0. 退出此系统*****
****1. 波兰式计算*****
****2. 逆波兰式计算*****
****3. 中缀表达式计算***
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
请输入数字，代表你要执行的功能
2
请一次性输入后缀表达式，每次输入都摁一个空格确认你的输入（以#结束）：
数字之间没有空格就代表是同一个数，数字之间有空格就代表是不同的数
2 3 A - #
输入非法字符，结果作废，请重新输入
结果是-1

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
****0. 退出此系统*****
****1. 波兰式计算*****
****2. 逆波兰式计算*****
****3. 中缀表达式计算***
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
请输入数字，代表你要执行的功能
3
请一次性输入中缀表达式，每次输入都回车或空格确认你的输入（以#结束）：
注意：
这里一定要两个##结尾!!! 少摁了#就再摁一次#!!!
4 + a ##
输入非法字符，结果作废，请重新输入
空栈，错误操作
结果是-44

```

当输入表达式操作数和运算符数目不匹配（操作数多或运算符多），如： $+ 2 2 2$ 、 $2 2 2 +$ 、 $2 + 2 2$ 、 $++ 2 2$ 、 $2 2 ++$ 、 $2 ++ 2$

我们将其分为两种情况。情况 1 是操作数数目多，则程序会提示输入的操作数过多，请重新输入，并且给出的结果是忽略程序认为多余的操作数，将其自动匹配，得到的修改后的结果，需要自行根据错误及需求来判断是否能用。如果能用则不用重新输入，如果不能，则重新输入表达式即可。

```

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
****0. 退出此系统****
****1. 波兰式计算****
****2. 逆波兰式计算****
****3. 中缀表达式计算****
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
请输入数字，代表你要执行的功能
1
请一次性输入前缀表达式，每次输入都摁一个空格确认你的输入（以#结束）：
数字之间没有空格就代表是同一个数，数字之间有空格就代表是不同的数
+ 2 2 2 #
多数字：2
操作数过多，请重新输入
结果是4

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
****0. 退出此系统****
****1. 波兰式计算****
****2. 逆波兰式计算****
****3. 中缀表达式计算****
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
请输入数字，代表你要执行的功能
2
请一次性输入后缀表达式，每次输入都摁一个空格确认你的输入（以#结束）：
数字之间没有空格就代表是同一个数，数字之间有空格就代表是不同的数
2 2 2 + #
多数字：2
操作数过多，请重新输入
结果是4

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
****0. 退出此系统****
****1. 波兰式计算****
****2. 逆波兰式计算****
****3. 中缀表达式计算****
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
请输入数字，代表你要执行的功能
3
请一次性输入中缀表达式，每次输入都回车或空格确认你的输入（以#结束）：
注意：
这里一定要两个##结尾！！！少摁了#就再摁一次#！！！
2 + 2 2 ##
多数字：2
操作数过多，请重新输入
结果是4

```

当运算符数目多时，程序会报错，提示取到了在运算符运算时取到了空栈。

```

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
****0. 退出此系统****
****1. 波兰式计算****
****2. 逆波兰式计算****
****3. 中缀表达式计算***
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
请输入数字，代表你要执行的功能
1
请一次性输入前缀表达式，每次输入都摁一个空格确认你的输入（以#结束）：
数字之间没有空格就代表是同一个数，数字之间有空格就代表是不同的数
+ + 2 2 #
空栈，错误操作
结果是8

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
****0. 退出此系统****
****1. 波兰式计算****
****2. 逆波兰式计算****
****3. 中缀表达式计算***
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
请输入数字，代表你要执行的功能
2 2 + + #
请一次性输入后缀表达式，每次输入都摁一个空格确认你的输入（以#结束）：
数字之间没有空格就代表是同一个数，数字之间有空格就代表是不同的数
空栈，错误操作
结果是4

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
****0. 退出此系统****
****1. 波兰式计算****
****2. 逆波兰式计算****
****3. 中缀表达式计算***
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
请输入数字，代表你要执行的功能
3
请一次性输入中缀表达式，每次输入都回车或空格确认你的输入（以#结束）：
注意：
这里一定要两个##结尾！！！少摁了#就再摁一次#！！！
2 + + 2 ##
空栈，错误操作
结果是-44

```

当输入表达式括号无法完成配对等非法输入，如 $(4 + 2 * 3, 4 + 2) * 3$ 时，程序会警告，输入了非法字符，比如上面的“、”并且因为程序最终无法运算前面的操作数导致程序也会认为是操作数过多，我们只需要知道这时候说明匹配等非法输入出现即可。另外，我认为应该再测试一下有左括号和右括号无法匹配的情况：比如 $(4+2)) * 3$ ，程序会提示括号匹配有误，输入有误，需要重新输入。

