

# 作业: 计算器的实现

钟琦

混合 2103 3210103612

2023 年 11 月 1 日

## 1 问题描述

本作业要求设计一个四则混合运算器, 要求忽略用户输入的字符串中除了小数点, +, -, \*, / , (, ) 以及空格以外的字符, 并将剩下的字符识别成浮点数和运算符。若字符串能组成合法的中缀表达式, 输出运算结果, 否则, 输出 “Error”。另外, 对于包含多行表达式的测试文件 input.txt, 本程序操作顺序为每读取一行得到一个结果, 再读取下一行, 直到操作完成。

## 2 设计思路

由于用户输入的中缀表达式不易被程序直接运行计算, 因此本程序依次进行**筛选合法的中缀表达式、形成中缀表达式、将中缀表达式转化为后缀表达式、计算后缀表达式**四个步骤。

### 2.1 筛选合法的中缀表达式

考虑到生成中缀表达式、后缀表达式并得到结果的计算量较大, 因此本程序首先筛选出合法的中缀表达式, 对不合法的中缀表达式直接报错输出 “Error”, 不再进入将中缀表达式转化到后缀表达式的循环中, 节省了算力。

对于筛选合法的中缀表达式, 本程序自主设计 isValid() 函数, 通过排除法筛选出合法的中缀表达式。其中, 排除的情况有: 两不连续数字之间无运算符, 运算符 (括号除外)、小数点早于第一个数字出现, 数字或 “)” 没有出现在表达式末, 运算符 (括号除外) 连续出现, 小数点在同一数字中出现两次及以上, 括号没有配对。

需要注意的是，本程序设定空格若出现在两个数字字符之间，则代表两个数字，即“1 2”不代表“12”，而表示“1”和“2”两个数字，中间没有运算符出现的情况下，为不合法的中缀表达式。

## 2.2 形成中缀表达式

此步骤主要是考虑到输入的中缀表达式中数字可能需要多个字符表示，例如“1.2f3g45”代表“1.2345”，需要将 char 类型转换成 double 类型。但若直接转换成 double 类型，则无法和运算符“+”“-”“\*”“/”等一起存储到队列 mid\_exp 中，所以采用了 struct 数据类型，包含数字 num 和符号 symbol 两种数据，其中代表数字的类型用 symbol=“\$”来标识。

## 2.3 将中缀表达式转化为后缀表达式

这一步骤用于形成方便计算的后缀表达式，这种形式的表达式可以单用数字和运算符的顺序给出唯一的结果，单从出现顺序表示计算顺序，而不用考虑运算符本身或括号的优先级。此步骤中遍历步骤 2 中形成的中缀表达式，用栈 op 暂时存储一些优先级较低的运算符，队列 exp 存储后缀表达式，以生成后缀表达式的规则进行 push 和 pop 操作，最终得到合法的后缀表达式。

## 2.4 计算后缀表达式

将中缀表达式转化为后缀表达式，可以大大降低计算的复杂程度。只要在遍历后缀表达式 exp，用栈 number 暂时存储数字，当读取到运算符时，用该运算符运算 number 中最后压入的两个数（同时 pop 掉这两个数），并将得到的数字再次压入栈，最终 number 中仅剩一个数字，即为结果。

# 3 测试说明

结合 input.txt 测试文件及测试结果：

(1) - (4) 为作业要求文档中提示 2 给出的一些输出对应，(5) 代表单个数字无运算符的运行结果，(6) (7) 代表加减乘除都存在的混合运算，并以运算符左右是否为在空格作为对照，(8) 代表多层括号的复合运算，(9) - (13) 为一些普遍性的四则运算，(14) 为以零为除数的情况，将产生警告

“Can not be divided by zero!”, (15) 为无法组成有效（但合法）的表达式，(16)-(18) 为相邻数字间没有运算符的情况，(19) - (21) 分别为运算符在开头、连续、在结尾的情况，(22) - (26) 分别为小数点在开头、后面没有紧跟数字（忽略无效字符）、一个数中出现两次小数点的情况，(27) 为括号没配对的情况。

经验证，下列所有情况均与预设相同，即，(1)、(2)、(5) - (13) 均输出正确的表达式结果，(3)、(4)、(16) - (27) 因无法形成合法的中缀表达式而输出 “Error”，(14) 因除以 0 输出警告 “Can not be divided by zero!”，(15) 因无法识别有效表达式而输出 “No Expressions Identified!”。

$$(1) \quad 1.2sdf3!!3 - 0.23 \longrightarrow 1.003$$

$$(2) \quad 1t.y21 * (3RR - 4T + 1s.s00) \longrightarrow 0$$

$$(3) \quad 1((2 \longrightarrow Error$$

$$(4) \quad 1.22.3 + 12 \longrightarrow Error$$

$$(5) \quad 1.23sf4g6 \longrightarrow 1.2346$$

$$(6) \quad 1.1 + 2.23 * (3.345 * 4.4567 / 5.56789 - 6.6789) \longrightarrow -7.82327$$

$$(7) \quad 1.1 + 2.23 * (3.345 * 4.4567 / 5.56789 - 6.6789) \longrightarrow -7.82327$$

$$(8) \quad 2f * g(1.2p \backslash n + (3 + (4 * 8 - 2) * 5 =)) \longrightarrow 308.4$$

$$(9) \quad 1.z1q1u1r1m1s * s9f - e1v0 \longrightarrow -1e - 05$$

$$(10) \quad 14 \# * 5/13 * 4/7 \longrightarrow 3.07692$$

$$(11) \quad 9 / 4 * (7 / 8 - 1 / 2) \longrightarrow 0.84375$$

$$(12) \quad (4.92 + 44/7 + 2.08 + 33/7) * (17/8 - 0.125 + 1) \longrightarrow 54$$

$$(13) \quad 0.0125 * 16/5 + 1/7 * 87.5 / (15/16) * (16/15) - 14/6 * 6 \longrightarrow 0.262222$$

$$(14) \quad 2 * 3 + 1/0 \longrightarrow Can not be divided by zero!$$

$$(15) \quad Abcd \longrightarrow No Expressions Identified!$$

$$(16) \quad 1 a2 \longrightarrow Error$$

$$(17) \quad 1. 2 \longrightarrow Error$$

$$(18) \quad 1 \ (4) \longrightarrow Error$$

$$(19) \quad fg + 1 \longrightarrow Error$$

$$(20) \quad 1 + z * 2 \longrightarrow Error$$

$$(21) \quad 1 + mn \longrightarrow Error$$

$$(22) \quad opq.1 + 2 \longrightarrow Error$$

$$(23) \quad 1. \ 2 \longrightarrow Error$$

$$(24) \quad 1. + 3 \longrightarrow Error$$

$$(25) \quad 12.g \longrightarrow Error$$

$$(26) \quad 1.2.2 \longrightarrow Error$$

$$(27) \quad 1((+2) \longrightarrow Error$$