

# 算术表达式求值

## 一、问题描述

从键盘上输入中缀算术表达式，包括括号，计算出表达式的值。

## 二、基本要求

- (1)程序能对所输入的表达式作简单的判断，如表达式有错，能给出适当的提示。
- (2)能正确计算含有：单目运算符：+，-，双目运算符：+，-，\*，/，%，^，括号（，）的表达式
- (3)能处理整数与小数运算

## 三、工具及准备工作

硬件：联想ThinkBook 16+

软件：VS 2022

## 四、分析与实现

### • 设计架构

calculator\_homework.h

```
template <class ElemType>
class Calculator
{
public:
    Calculator();
    Calculator(const char* pstrInfix);
    ~Calculator();
    const char* GetInfix();
    const char* GetPostFix();
    bool InfixToPostfix();//中缀表达式转换为后缀表达式
    bool CalcuPostfix(ElemType& evalue);//计算后缀表达式的值
    virtual Calculator& operator = (const char* pstrInfix); // 重载赋值运算符
    bool PrintPostfix(); // 定义输出后缀表达式的函数
    bool CheckInfix();//检查中缀表达式是否合法

private:
    CDynamicArray <char> m_larryInfix; //中缀表达式字符动态数组
    CDynamicArray <char> m_larryPostfix; //后缀表达式字符动态数组
    int IspPriority(char chOperator); //获取操作码栈内优先级
    int IcpPriority(char chOperator); //获取操作码栈外优先级
    bool IsOperator(char ch); //是否字符是操作码

};
```

```
#include <iostream>
#include <string>
#include "../include/calculator_homework.h"
using namespace std;

int main()
{
    int n;
    float res;
    string s;
    cout << "请输入表达式个数:";
    cin >> n;
    for (int i = 0; i < n; i++) {
        cout << "表达式:";
        cin >> s;
        calculator<float> cal(s.c_str());
        cal.InfixToPostfix();
        if (cal.CheckInfix() && cal.CalcuPostfix(res)) {
            cout << "结果: " << res << endl;
        }
        else {
            cout << "不合法" << endl;
        }
    }
    return 0;
}
```

• 算法整体流程

- 1. 从输入流读入待计算的中缀表达式
- 2. 检查表达式是否合法

- 1. 未定义符号检查
- 2. 括号完整性检查
- 3. 末尾等号检查

- 3. 将中缀表达式转换为后缀表达式

转换时栈内栈外优先级：

运算符	=	(	*, /, %	+, -	^	)
栈内优先级	0	1	5	3	7	8
栈外优先级	0	8	4	2	6	1

通过维护一个栈存储运算符，再通过上表优先级，进行出入栈操作。

- 2. 对后缀表达式进行运算

通过判断类初始化的模板参数，将字符串转换为数字入栈。

维护一个栈，存储运算数，遇到运算符就进行运算。

由于小数的取余操作不存在，所以在进行取余操作时，会发生强制类型转换，转换为 `int` 后计算。

## 2. 输出结果

# 五、测试与结论

## 1. 基础功能测试

### ◦ 测试集设计

输入：

$-5+(-3)-(-2)=$

$-4*(2+5)+9/3^2=$

$(1+2)*(3-4)/(5^2)\%6=$

$(7^2)+(8/3)-(9*4)+(10\%2)=$

$((11+12)*(13-14))/(15/2)^(16\%3)=$

$(17+18)*(19-(20/2))^(21\%5)=$

$(22*(23+24))/(25-(26\%3))+(27/2)=$

$(28-(29*30))/(31+(32/33))*(34^(35-36))=$

$(37^2)+((38+39)*(40-41))/(42/43)=$

$(44*(45+46))-(47/2)+(48^(49\%3))=$

$(50/2)^(51-(52*53))+(54\%55)=$

$(56+(57*58))/(59-(60/61))*(62^(63/64))=$

输出：

-6

-27

0

15.6667

-3.06667

315

58.4565

-0.774631

1290.17

4028.5

54

3368.47

### ◦ 测试结果检验

```
表达式: -5+(-3)-(-2)=
结果: -6
表达式: -4*(2+5)+9/3^2=
结果: -27
表达式: (1+2)*(3-4)/(5^2)%6=
结果: 0
表达式: (7^2)+(8/3)-(9*4)+(10%2)=
结果: 15.6667
表达式: ((11+12)*(13-14))/(15/2)^(16%3)=
结果: -3.06667
表达式: (17+18)*(19-(20/2))^(21%5)=
结果: 315
表达式: (22*(23+24))/(25-(26%3))+(27/2)=
结果: 58.4565
表达式: (28-(29*30))/(31+(32/33))*(34^(35-36))=
结果: -0.774631
表达式: (37^2)+((38+39)*(40-41))/(42/43)=
结果: 1290.17
表达式: (44*(45+46))-(47/2)+(48^(49%3))=
结果: 4028.5
表达式: (50/2)^(51-(52*53))+(54%55)=
结果: 54
表达式: (56+(57*58))/(59-(60/61))*(62^(63/64))=
结果: 3368.47
```

测试正确，结果无误！

## 2. 异常输入测试

### ○ 测试集设计

空表达式、括号问题、非法字符、未以括号结尾、运算符冗余、零除、超出范围的数值

**输入:**

$$(2+3))(4-1=$$
$$((2+3)(4-1)=$$
 $2+3@5=$  $2+3$  $2+*3-5=$  $2+3- =$ 

5/0=

`999999999999999999999999999999999999+1=`

### ○ 测试结果检验

[illegible]

测试无误，程序能够正确分辨出各类不合法输入！

## 六、思考与感悟

- 深入理解并实践了C++的面向对象编程和模板元编程等
- 学会了中缀转后缀和后缀表达式运算的方法
- 加深了对栈、队列的理解
- 后续可以通过添加支持的运算操作，细化不合法表达式报错提示等加以改进