**江西财经大学虚拟现实（VR）现代产业学院**

**2021-2022学期C++程序设计课程大作业报告**

**———————————————————————————**

**专业：\_软件工程（VR方向）\_ 班级：\_校内四班\_ 授课教师：\_张翰进\_**

**大作业名称：\_\_\_\_\_\_\_\_\_\_\_\_\_\_基于C++语言的高性能计算器实现\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**姓名：\_\_\_\_\_\_刘继鹏\_\_\_\_\_ 学号：\_\_\_\_\_5220833\_\_\_\_\_\_**

**评分：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**———————————————————————————**

1. **大作业题目**

使用C++语言设计一个高性能计算器，具有以下功能：

（1）基本功能：

（i）在控制台下输入一个表达式，例如，输入 3+5\*4，可以按输入的表达式中的运算符优先级进行处理，得到23。

（ii）在控制台上输入一个表达式，例如，输入1+（3+5）\*4，可以按输入的表达式中的括号和运算符串的优先级进行处理，得到33。

（iii）设计的计算器可实现实数（含整数）的加、减、乘、除、乘方和开方运算。

（2）拓展功能：

（i）设计的计算器可实现复数的加、减、乘、除运算；

（ii）设计的计算器可实现无穷精度的整数的加、减、乘、除、求模运算（提示，使用结构体）。

1. **算法实现说明**

**（1）高性能计算器基本架构思路：**

使用一个C++泛型类作为整个计算器进行运算的框架，该框架用于处理用户的输入（由于考虑到有三种数据类型的输入，所以在数字和运算符中必须使用空格进行分隔），对输入的运算符号、数字进行区分，并且使用两个栈（符号栈和数字栈）和运算符优先级状态进行检测和运算。

用户只需要实现该计算器框架的数据类，并重载对应的方法和运算符，即可实现对应的运算。其中需要重载的方法有GetNumber（用于从自定义对象转到string类型以便输出），SetNumber（用于从string类型转到自定义对象，以便于计算），IsNumber（用于判断用户输入的当前字符串是否属于运算数字），以及各种运算符号的重载。

在最终main函数执行的过程中，对用户输入的数据进行判断，然后分别new不同的泛型类，即可实现不同的运算。

**（2）普通数字（实数）的加、减、乘、除、乘方、开方运算算法思路：**

在普通数字的运算中，只需要使用小数long double（或者double）数据类型即可，因为long double变量类型在c++中已经有相关的运算符，即可不用重载。

在乘方和开方运算中，则需要使用c++的cmath函数库中的pow方法，其中乘方使用符号^代替，开方使用#代替（2^3=8；4#2=2）。

**（3）复数加、减、乘、除运算算法思路：**

在复数的运算过程中，需要自己定义一个c++复数类（Complex类），然后分别重载加、减、乘、除的运算符号，并且做两个数字按照复数运算的规律（复数运算规律本质还是实部和虚部的普通数字运算，故此处省略）做对应的运算即可。

其中，在复数的输入中需要严格按照(实部+虚部i)的格式进行输入，这将会在SetNumber方法中对字符串进行拆分，从而分别赋值实部和虚部。

**（4）高精度（无穷精度）整数加、减、乘、除、求模运算算法思路：**

高精度运算和复数运算类似，同样需要定义一个c++高精度类（AccNumber类），其中对string字符串类型进行包装，并且使用一个bool的isMinus值来判断当前数是否为负数，然后分别重载运算符方法即可。以下将讲解高精度算法的基本思路。

（a）高精度加法：使用普通数字加法的模拟运算，首先将两个操作数字字符串反转，然后进行补零操作，将两个操作数字的字符数对齐，便于下面进行模拟运算。对每个位数进行相加操作，然后相除算出进位数，取余算出当前位置的真实数字即可。

（b）高精度减法：在减法运算中，首先需要保证减数大于被减数（4-2=2，其中4>2），否则将计算出真实的负数，无法使用isMinus进行表示。然后再进行普通数字减法的模拟运算，与加法模拟运算类似，通过反转，补零，模拟进位（此处进位实则为减，即借位）操作进行运算。

（c）高精度乘法：使用两个for循环分别去遍历两个操作数，对每个操作数进行普通数字乘法运算和进位即可，这里只需要注意乘法后保存到最终结果变量（res变量）的位置即可。

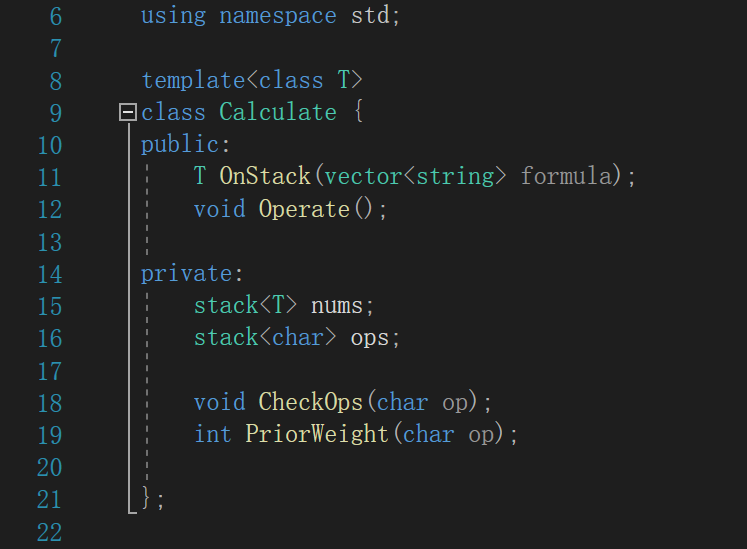
（d）高精度除法：对于除法运算，需要先判断除数和被除数的大小关系，因为在本次算法实现中，不涉及到小数问题，所以若除数小于被除数，则结果直接为0即可（2/3=0）。然后再模拟高精度的减法运算即可，其中首先将被除数补零，然后遍历除数字符的个数，每次遍历时，都将其减去被除数，每减去一次，则统计减去的次数，该次数就为结果中当前遍历位的数。如此反复直到循环完毕，即可算出高精度除法运算的结果。

（e）高精度取余（求模）：求模的本质实则为求除法后的余数，故只需要结合上述的高精度除法、乘法和减法即可计算完成。

1. **源代码及注释**

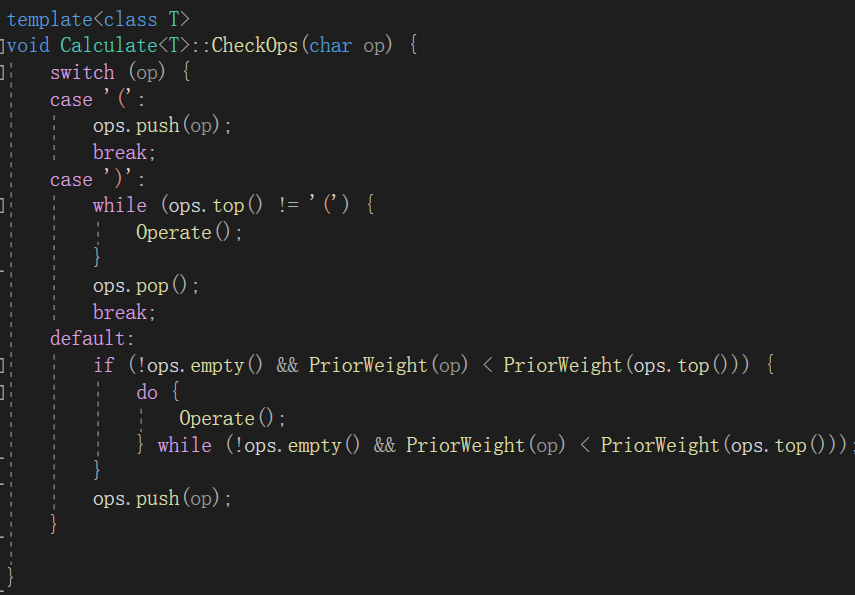
**（1）高性能计算器基本架构实现：**

（1.1）Calculate.h文件：

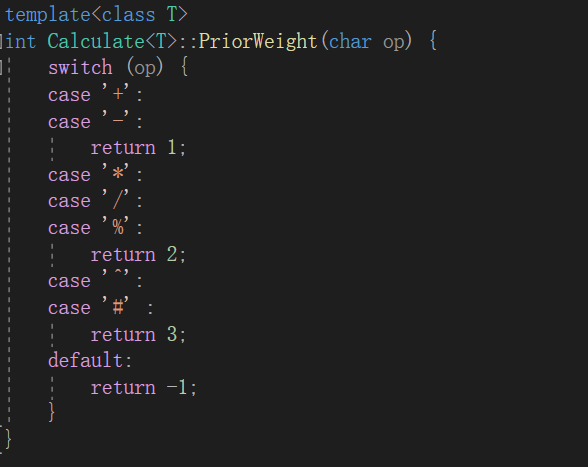


（1.2）Calculate.cpp文件：

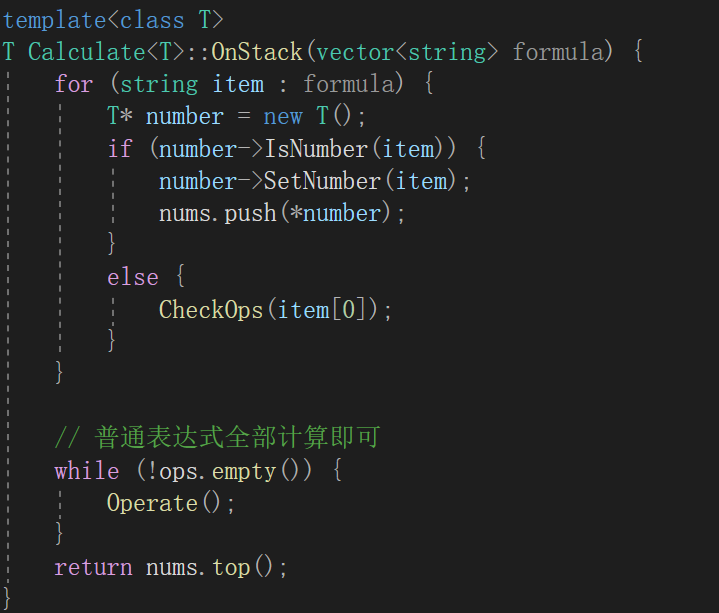
（a）CheckOps方法实现：



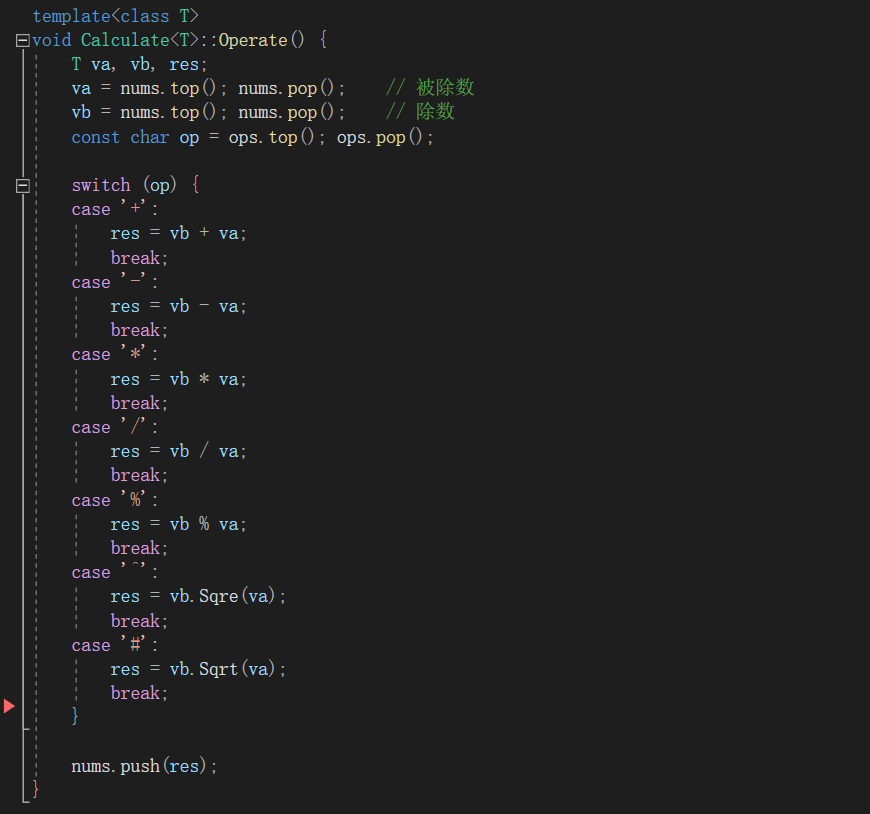
（b）PriorWeight方法实现：



（c）OnStack方法实现：



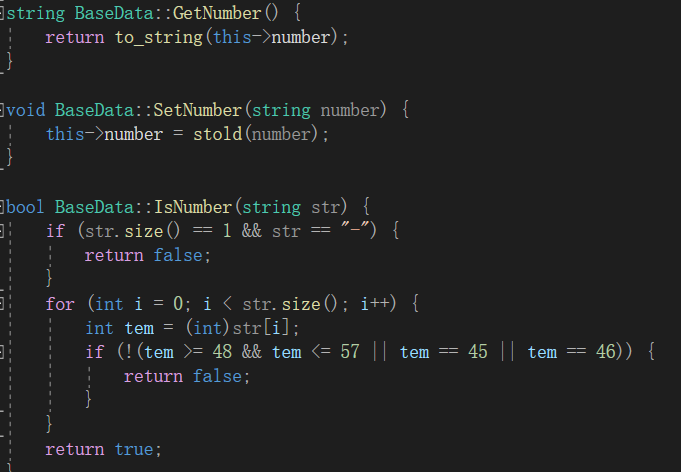
（d）Operate方法实现：



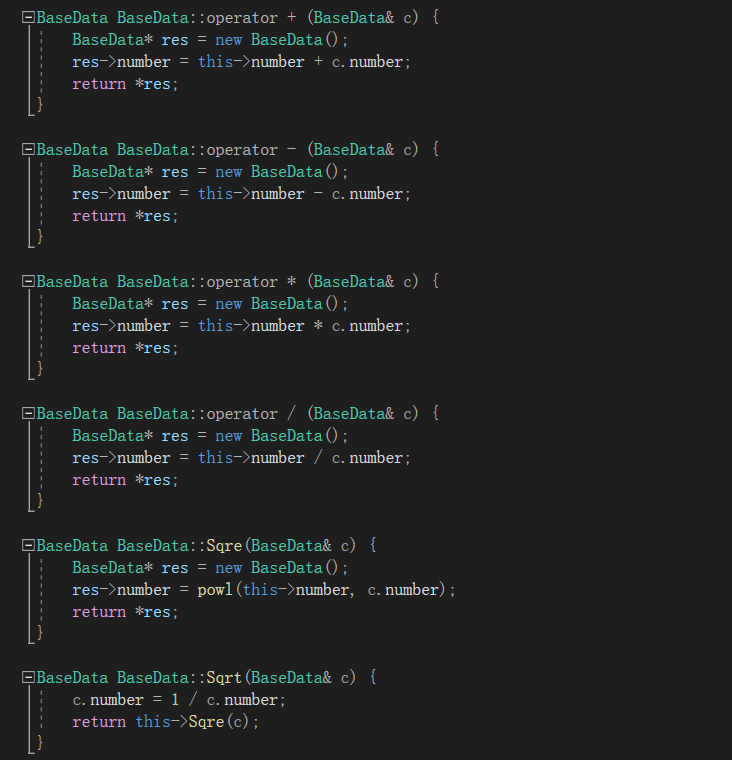
**（2）普通数字（实数）的加、减、乘、除、乘方、开方实现：**

* BaseData类字段：
  + Long double number;

基本方法实现：



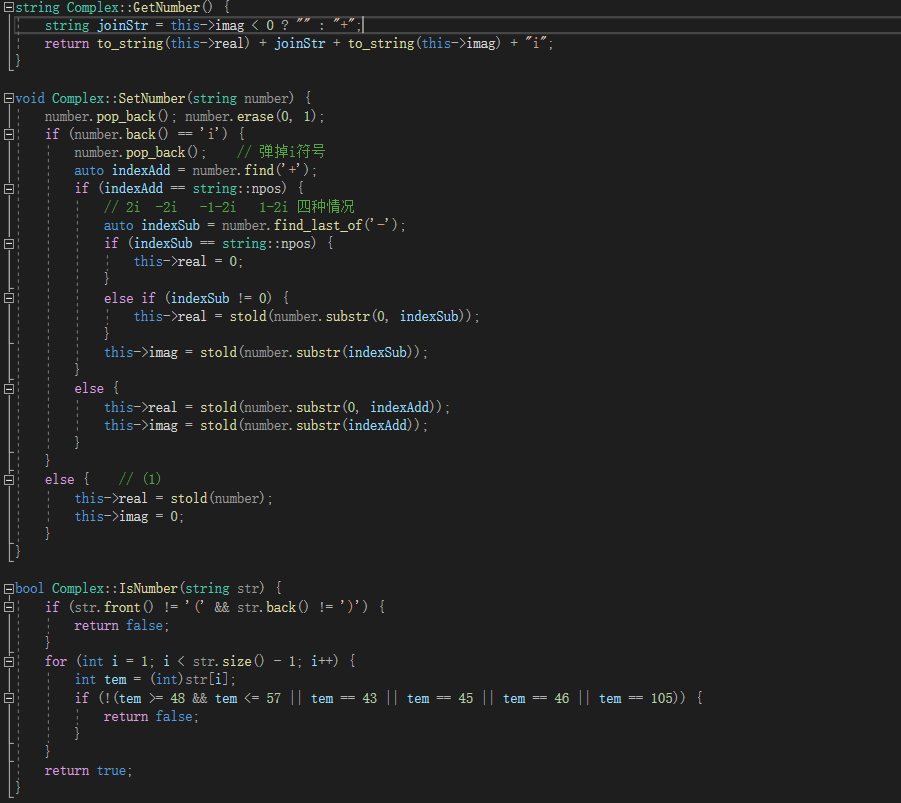
包装运算符号重载实现：



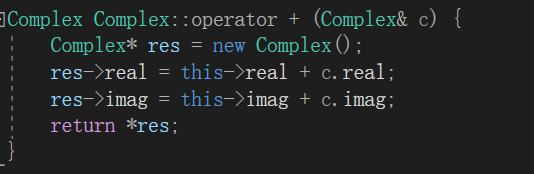
**（3）复数加、减、乘、除运算实现：**

* Complex类字段：
  + Long double real;
  + Long double imag;

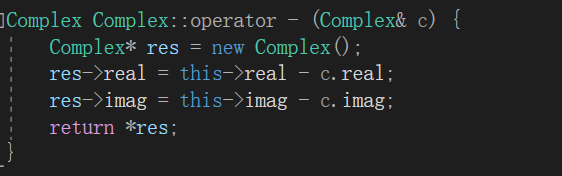
基本方法实现：



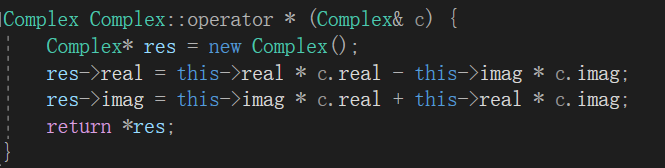
加法运算符重载：



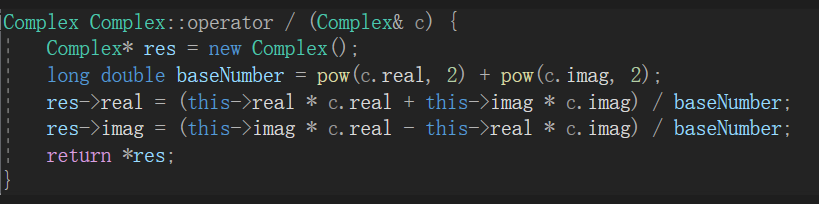
减法运算符重载：



乘法运算符重载：



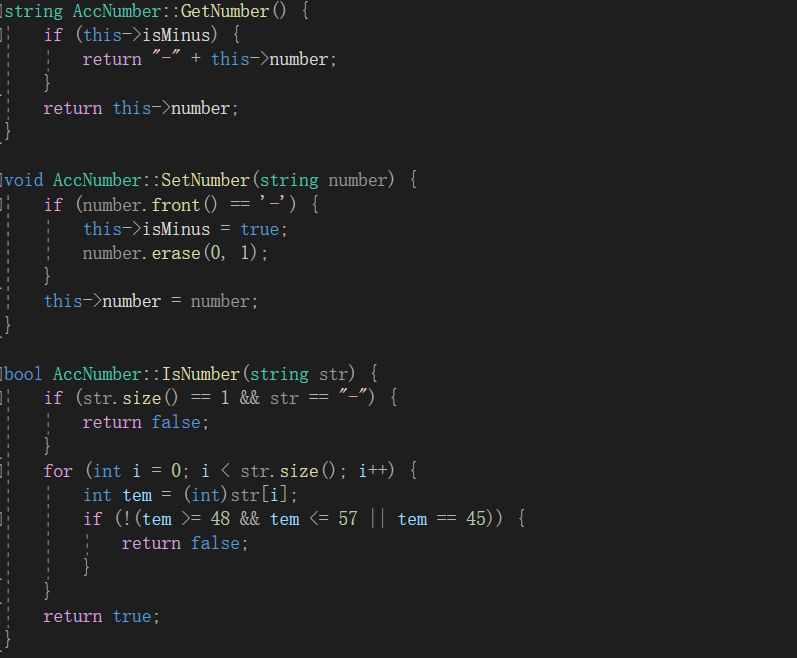
除法运算符重载：



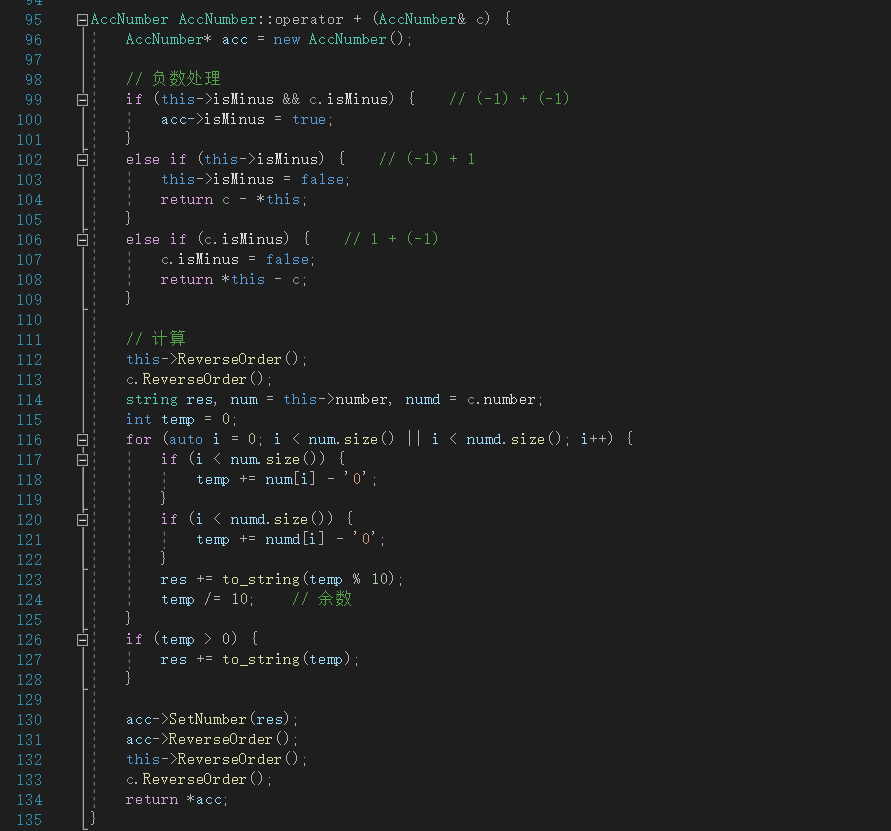
**（4）高精度（无穷精度）整数加、减、乘、除、求模运算实现：**

* AccNumber类字段：
  + String number;

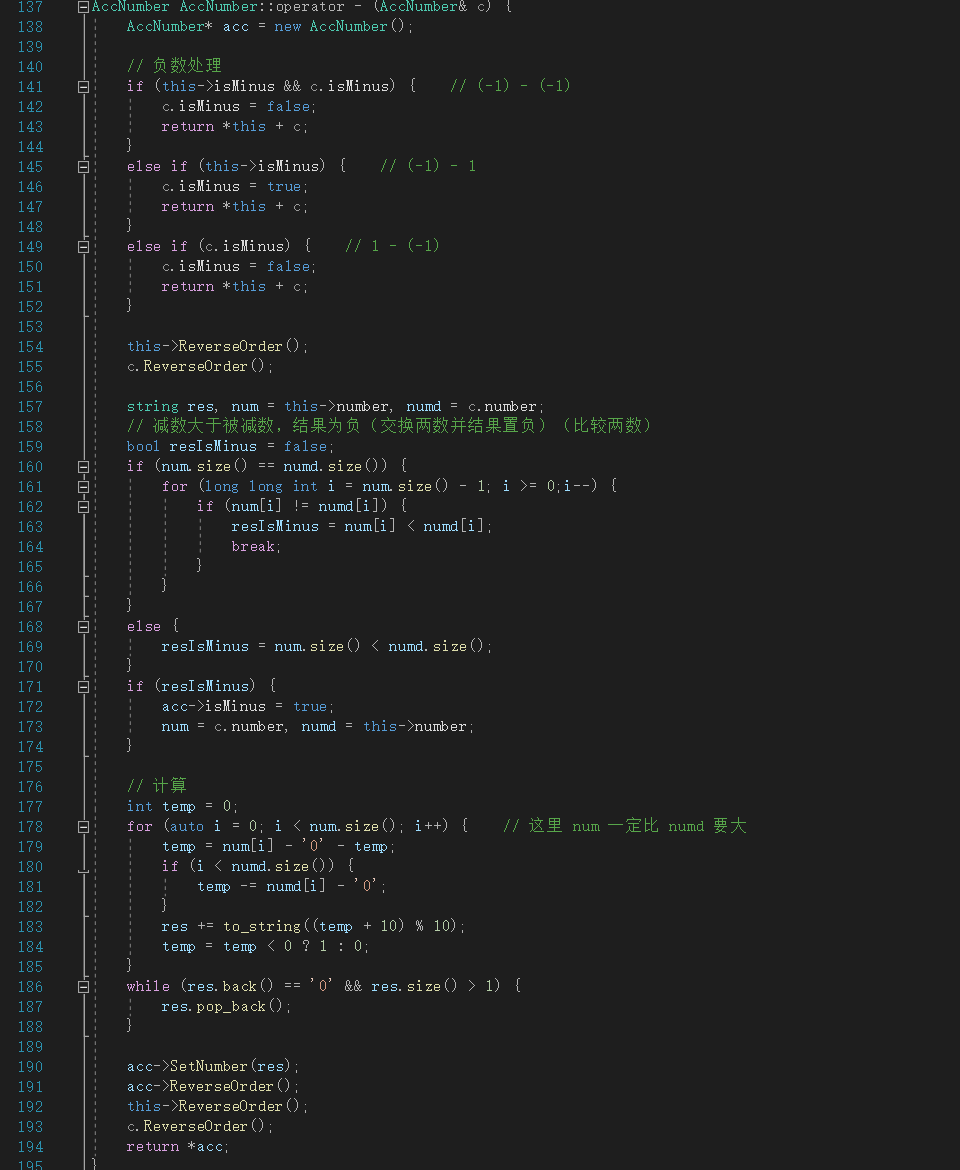
基本方法实现：



加法运算符重载：



减法运算符重载：



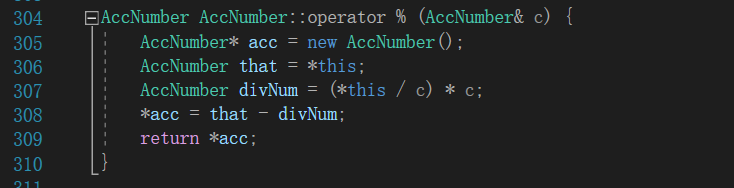
乘法运算符重载：



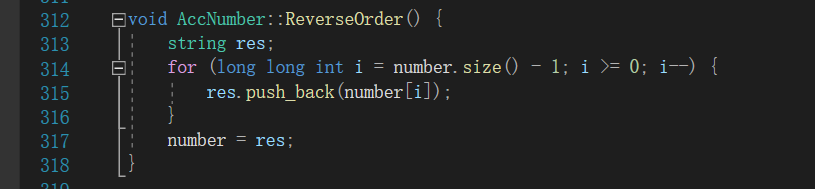
除法运算符重载：



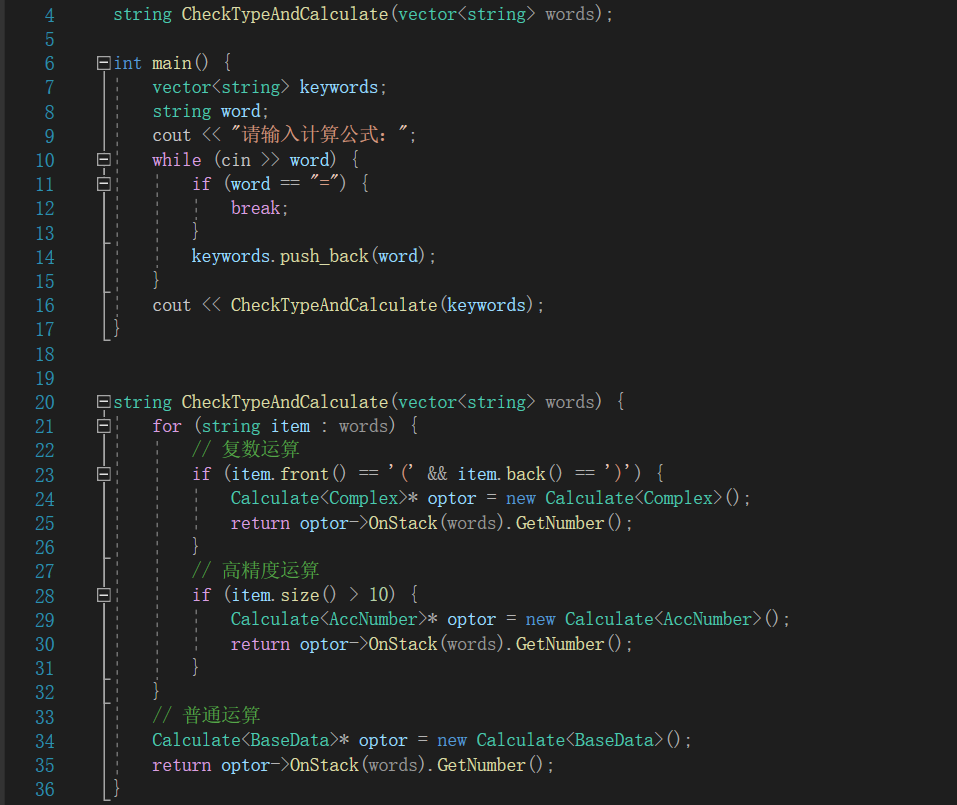
求模运算符重载：



私有字符串反转方法：

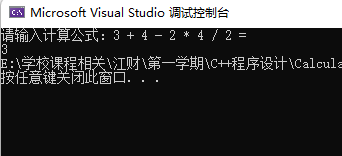


**（5）main函数判断用户输入实现：**

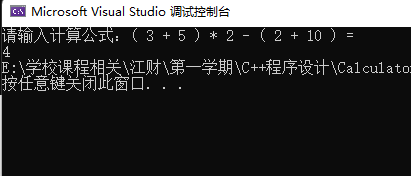


1. **运行结果截图与说明**

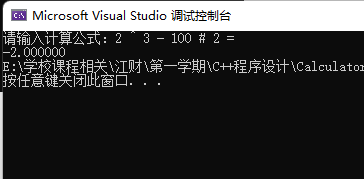
（1）普通数字加减乘除运算：



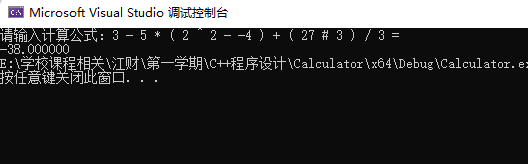
（3）普通数字带括号加减乘除运算：



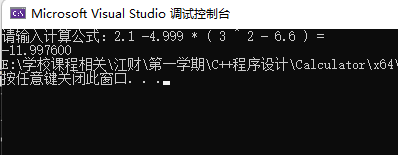
（4）普通数字乘方和开方运算：



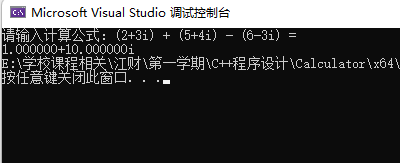
（5）普通数字运算综合实例：

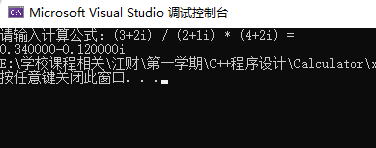


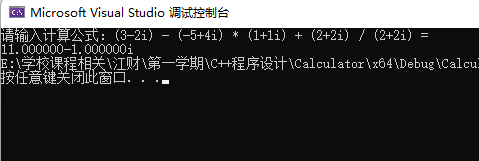
（6）普通数字运算综合实例（小数）：



（6）复数加减乘除运算：







（7）高精度数字运算：

