# Project2报告

**12113024**

**曹始文**

**计算机科学与工程**

### Part I 思路分析

此次project的任务有以下几点：

1. 求解一个混合运算式子的结果（带括号）
2. 给定变量的值，求代数式的结果
3. 运算中加入数学函数
4. 实现无限精度

以下为各个问题求解的基础思路：

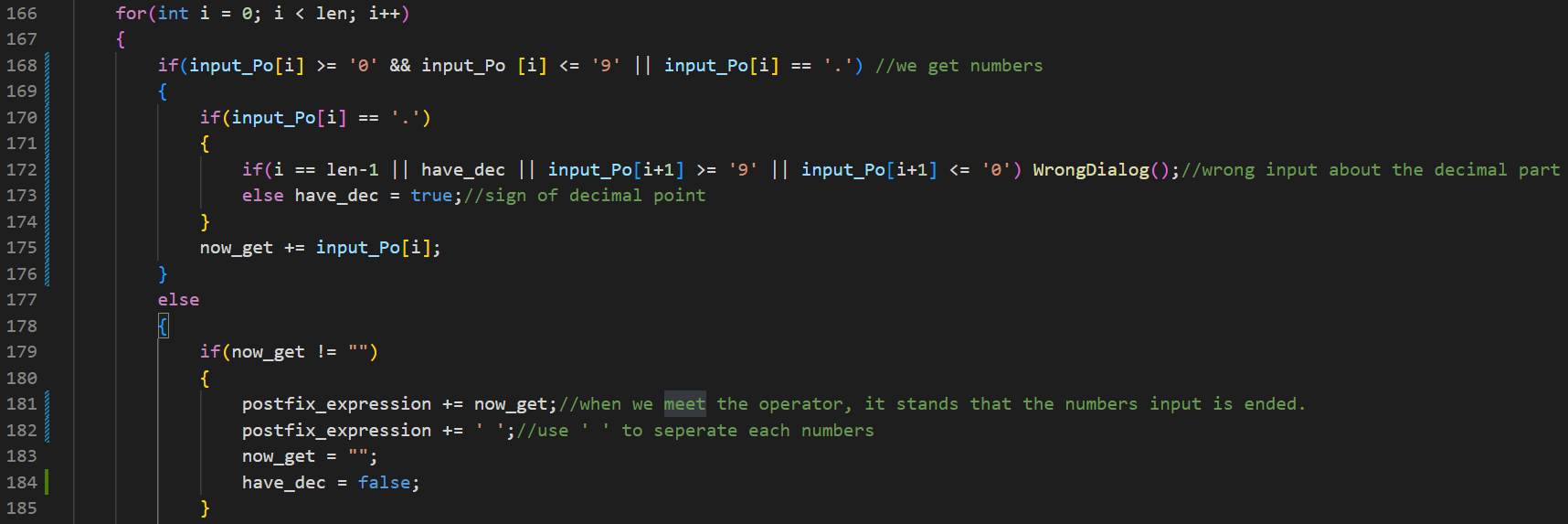
1. 求解混合运算，我们可以将中缀表达式转化成后缀表达式，通过模拟栈的操作来对数据进行计算
2. 求解代数式，我们可以先将每个变量的名称和数值存起来，再将表达式中的变量替换成其对应的数值大小，这样就可以转化成第一个问题进行求解。
3. 数学函数需要单独进行计算，我实现了两个较为简单的函数：pow(x,y) 和 abs(x)
   1. pow(x,y)实现方法：我使用快速幂去解决这个问题，由于幂函数变化很快，所以能处理的y不会特别大，否则会导致数值过大存储不下，使用快速幂能高效的得到正确答案。
   2. abs(x)实现方法：如果是负数直接删去负号
   3. 我的程序并未针对函数套函数这种形式进行求解（例如：abs(pow(x,y))）
4. 对于无限精度的实现，我们把每个数都分到整数和小数两个部分进行储存
   1. 小数乘法的实现：我们可以先把所有的数放到一起，然后对数进行乘法，乘完之后再找到结果的小数部分储存到小数数组，整数部分放到整数数组。
   2. 小数加减的实现：对于小数的加减法，我们将小数和整数两个部分分开算

# Part II 各个问题的具体实现

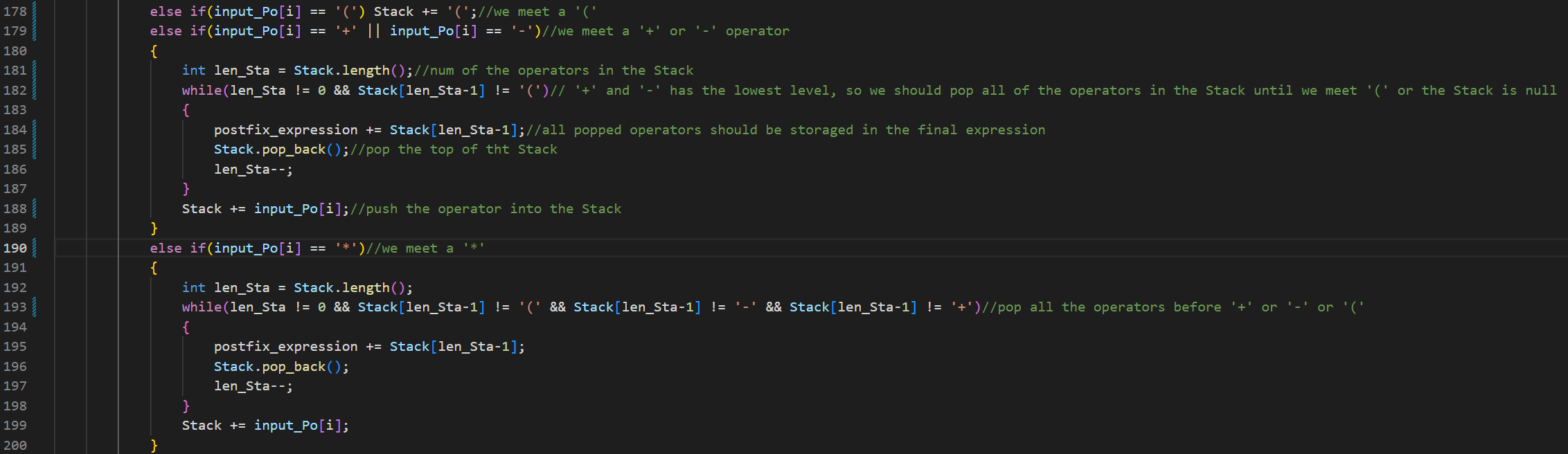
1. 混合运算求解（后缀表达式的转换和求解过程参考了<https://blog.csdn.net/amf12345/article/details/96868387> 这篇博客）
   1. 中缀表达式转换成后缀表达式：

转化过程为：

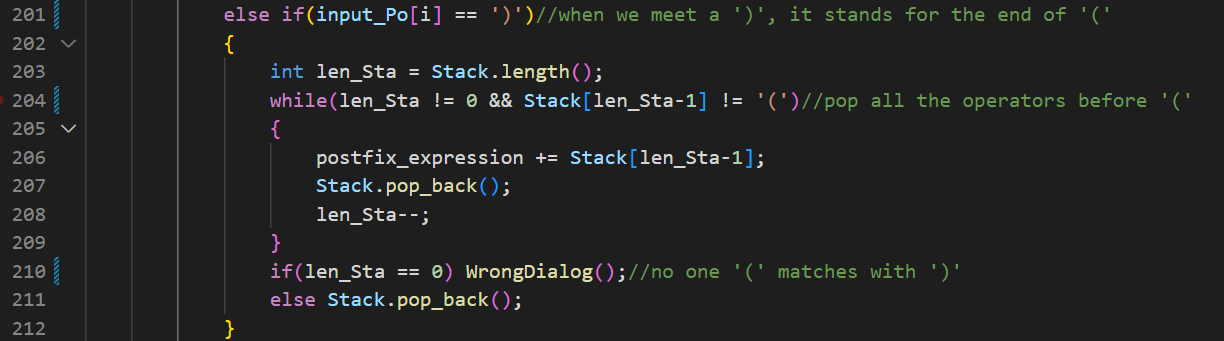
如果当前位置为数字，直接储存到结果中：



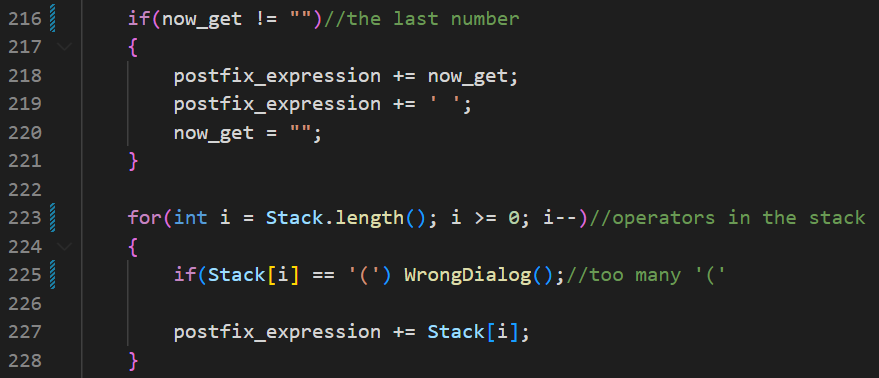
如果是运算符，我们需要从栈顶开始检查，将优先级高于当前运算符的字符全都储存到结果中，直到找到更低优先级的元素（或栈空），然后插入当前运算符（左括号也插入）：



如果遇到左括号，则放到栈中，当我们读到右括号时，就将操作符从栈顶开始储存到结果中，直到发现第一个左括号（左括号不存入结果）：

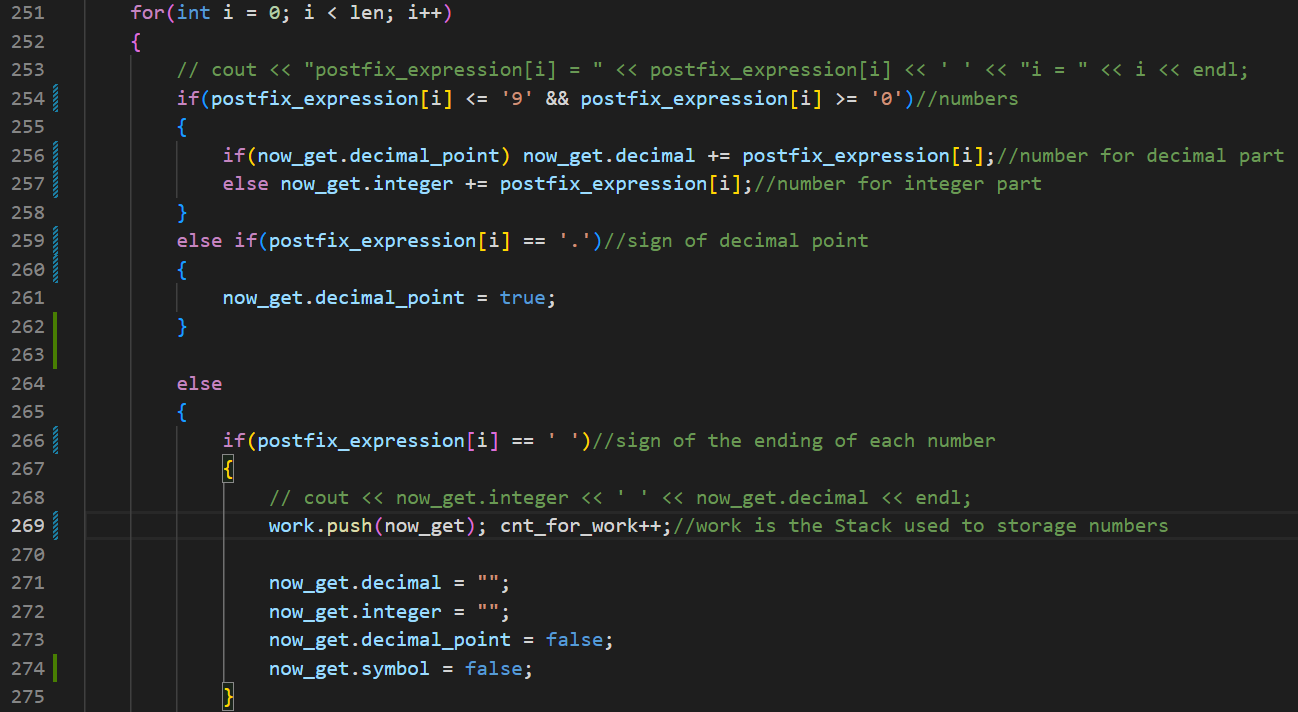


当读入结束时，栈中的所有操作符都要储存到结果中：



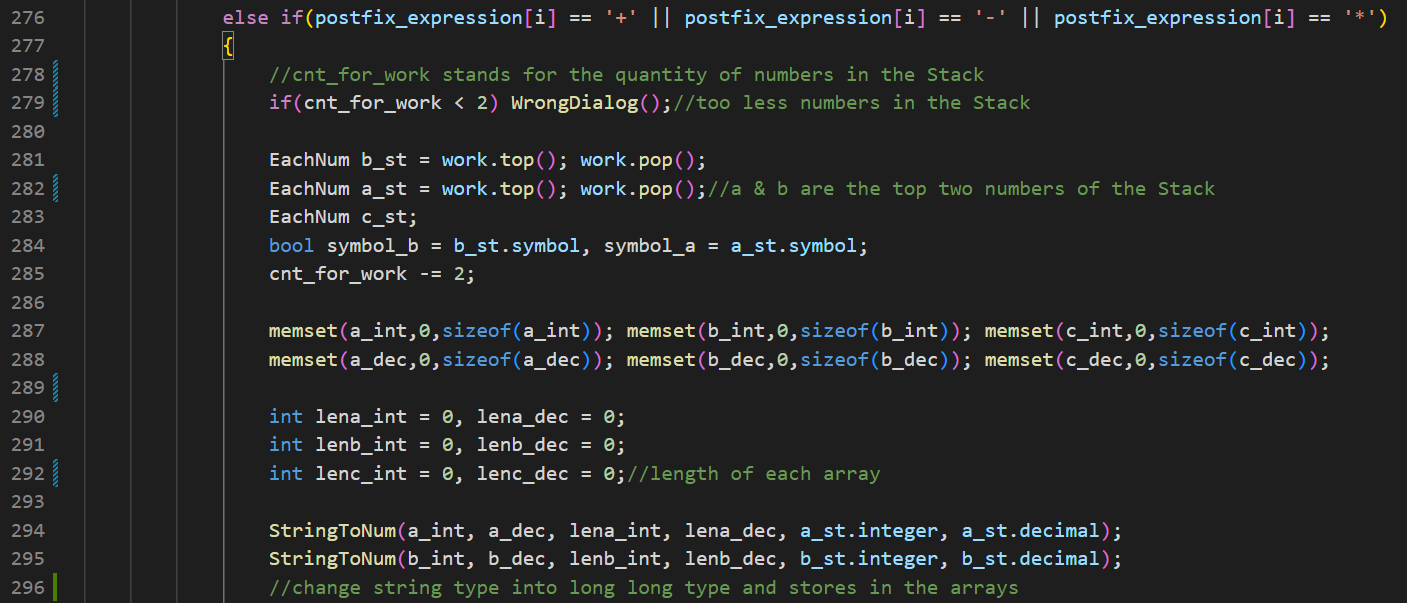
* 1. 后缀表达式的运算：

如果当前得到的是数，则放入栈中：

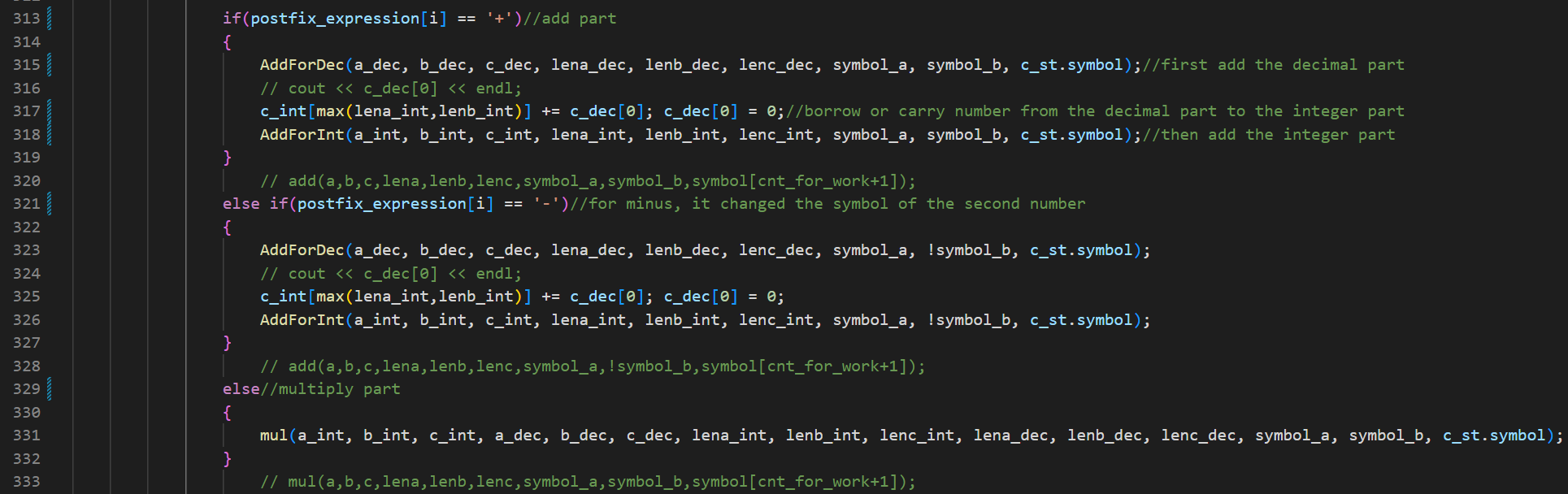


如果当前遇到的是运算符，则将栈顶的两个数据弹出进行运算：

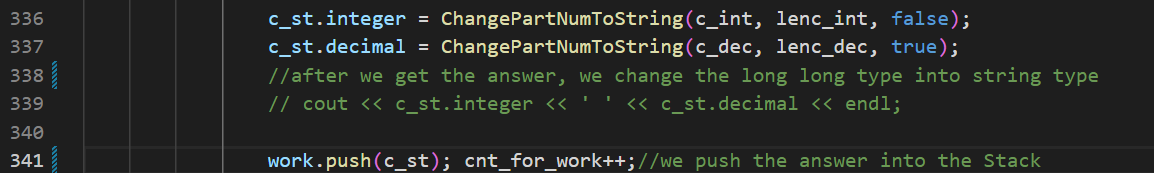
得到两个数



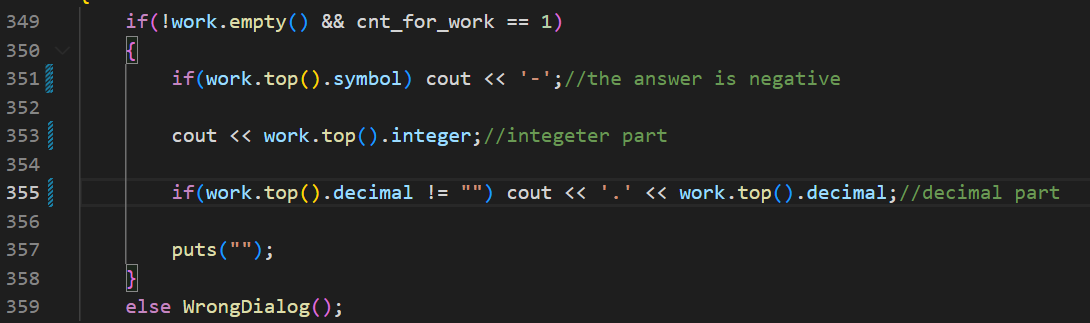
运算两个数



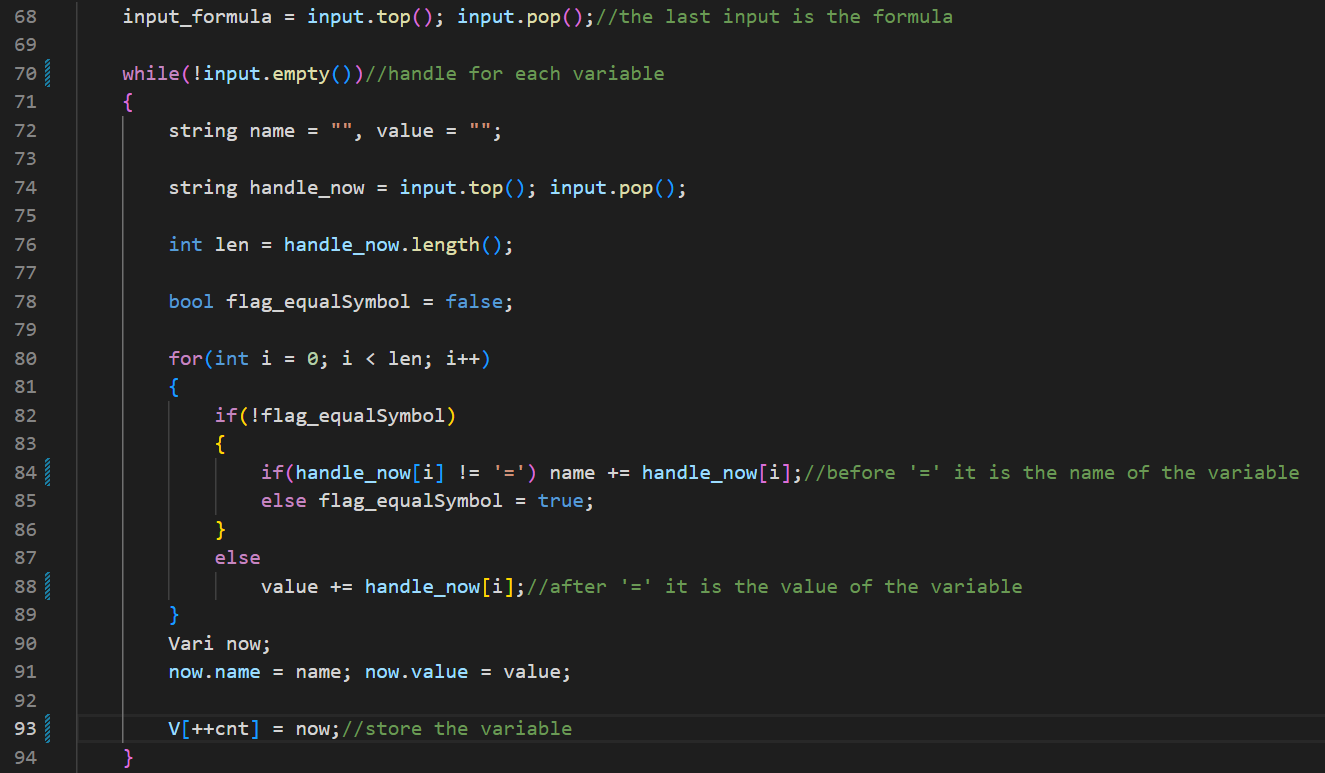
将结果放入栈中



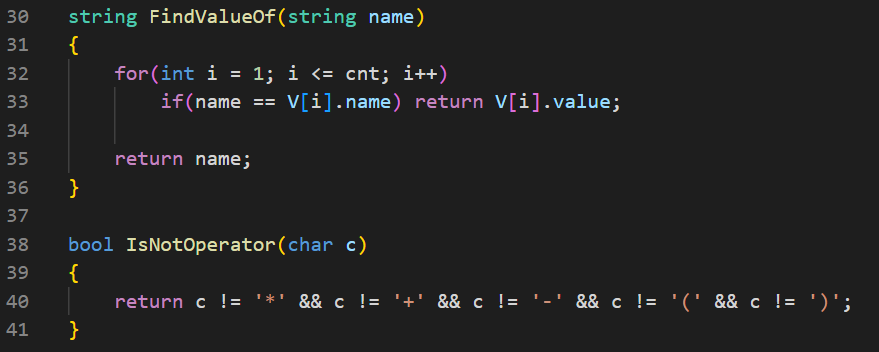
* 1. 最终结果即为栈中最后剩下的最后一个数：

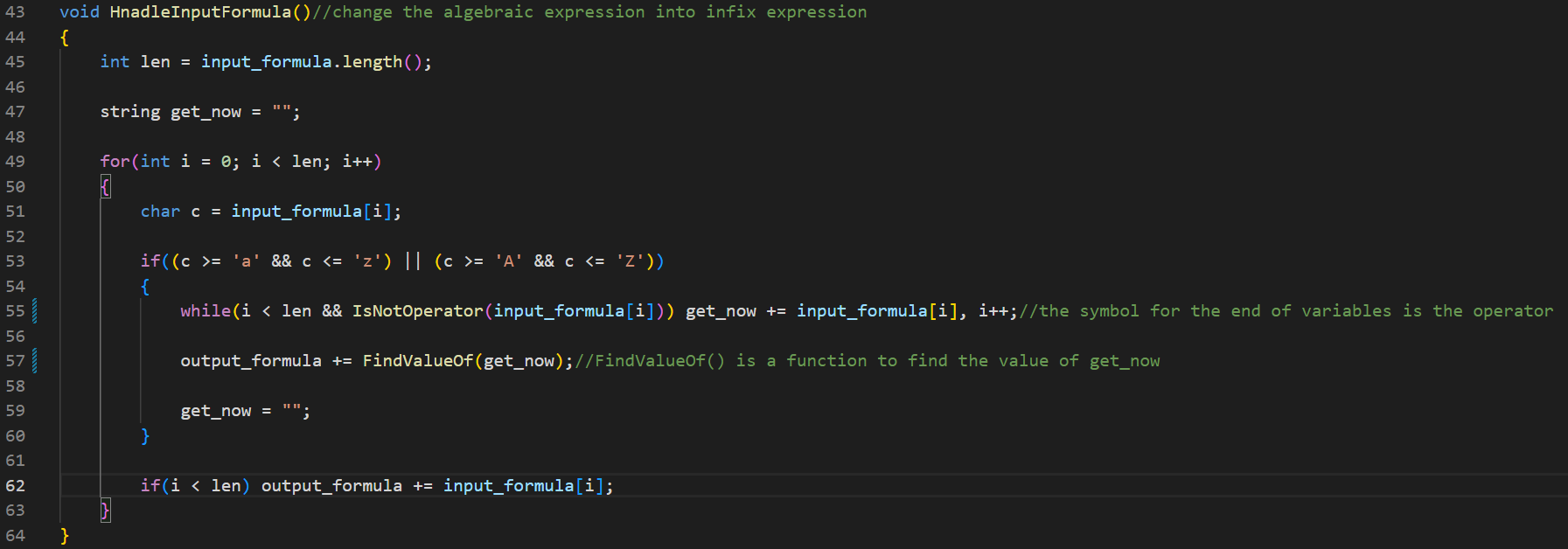


1. 代数式的求解：
   1. 首先将读入的变量的名称和值储存起来：



* 1. 将读入的式子中的变量转化成其对应值





* 1. 转化后的式子即可作为第一问需要的混合运算进行求解



1. 数学函数的处理
   1. 读入式子之后，我们提前对式子进行预处理，将其中的数学函数转化成具体结果（具体函数在Cpp\src\PolynomialWork.cpp 文件中的 PreHandleFormula函数中



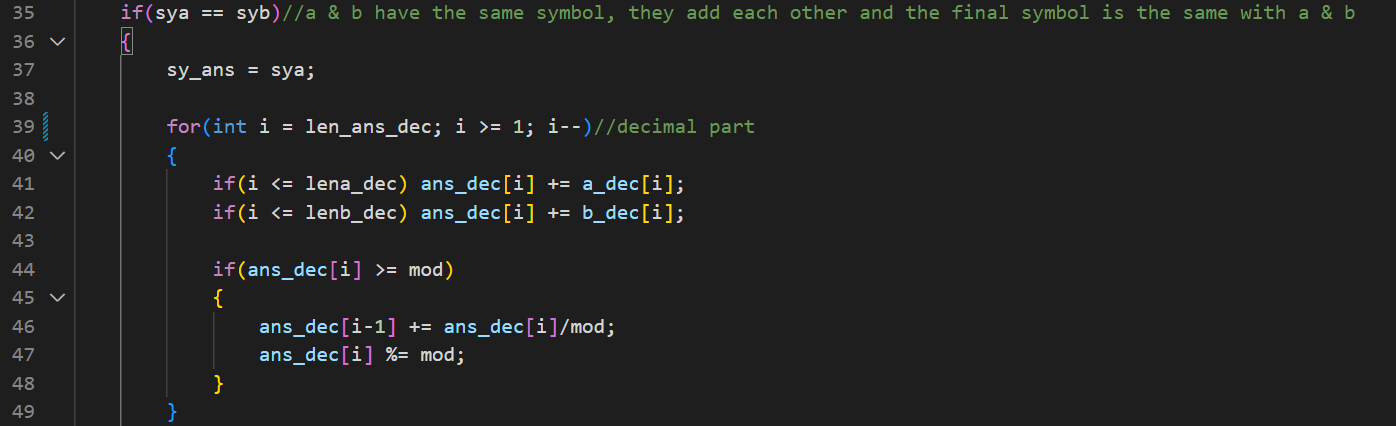
* 1. 处理完之后就变成了要解决的第一个问题

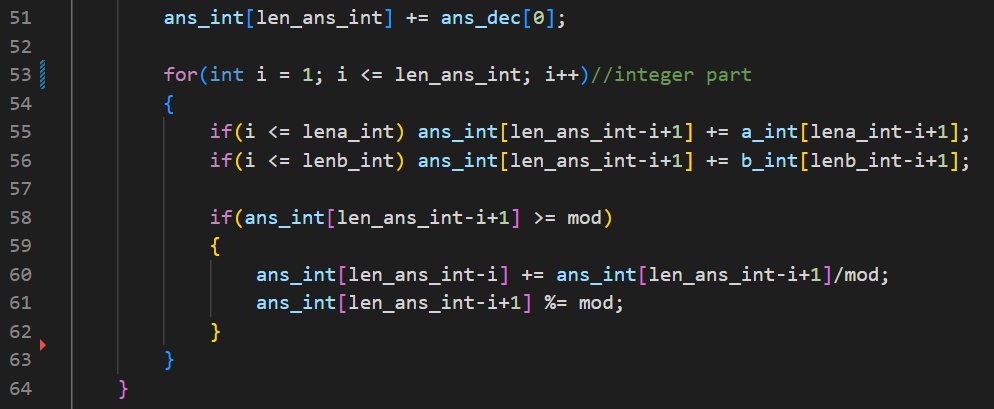
1. 无限精度的实现

将所有数据都存小数和整数两个部分即可

1. 加法（分为小数加法和整数加法）
2. 同号情况

直接将整数和小数部分相加

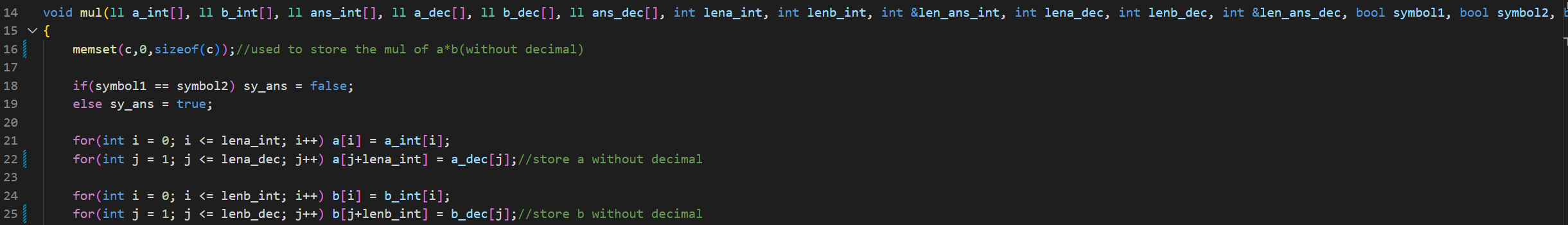


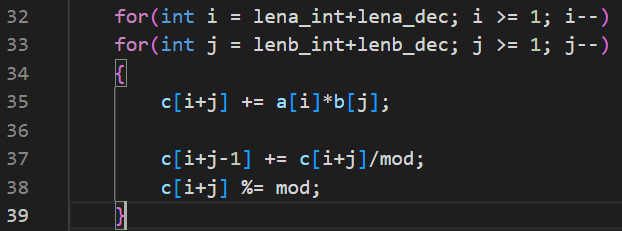


1. 异号情况

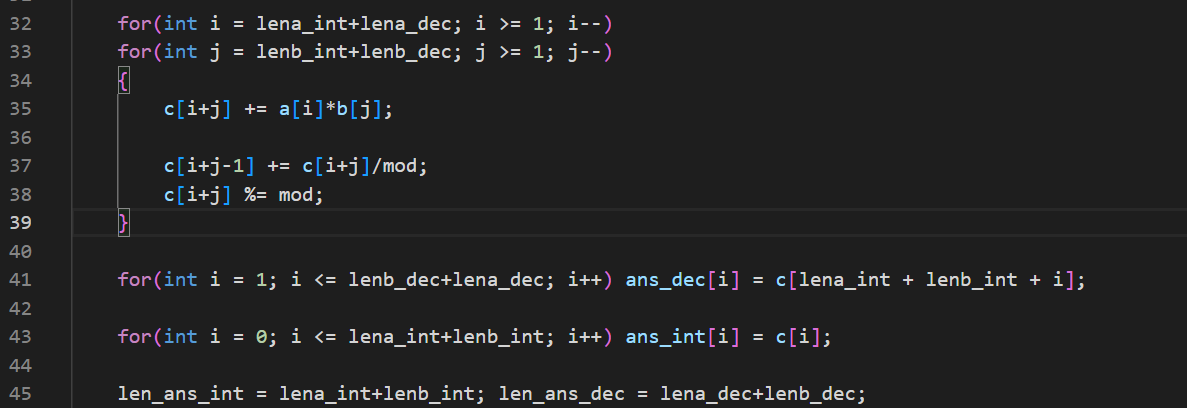
用大数减小数即可

1. 乘法实现
   1. 乘法处理较为粗暴，直接将a 和 b 的整数和小数部分结合到一起，然后进行乘法



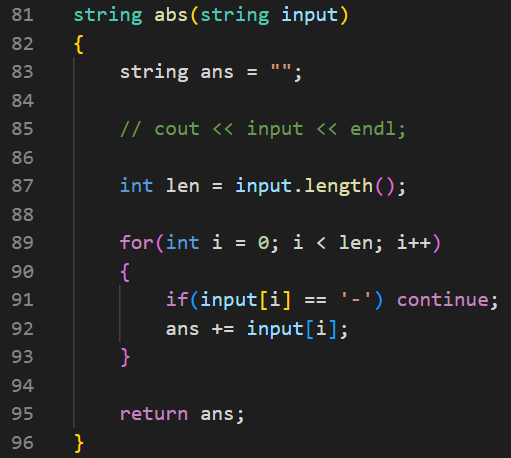


* 1. 乘完之后，再将小数部分提取出来，即最后的lena\_dec+lenb\_dec位为小数位；剩余均为整数

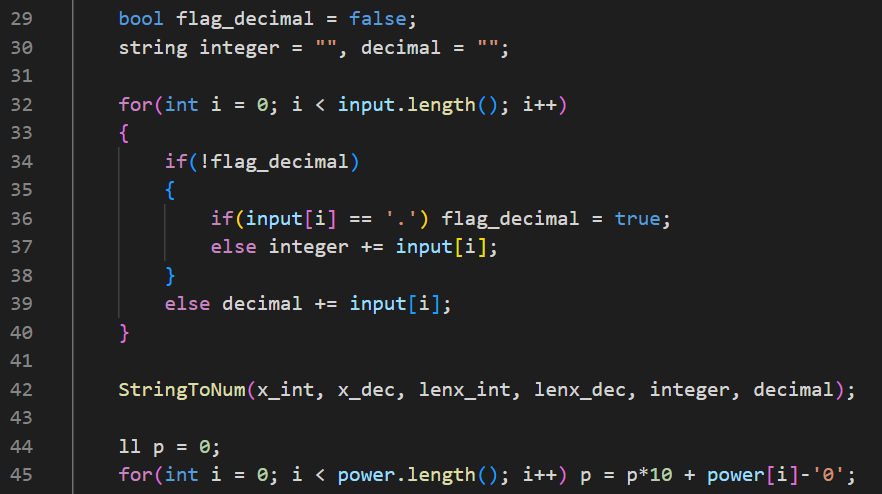


1. 绝对值实现

简单的将负数的负号除去即可



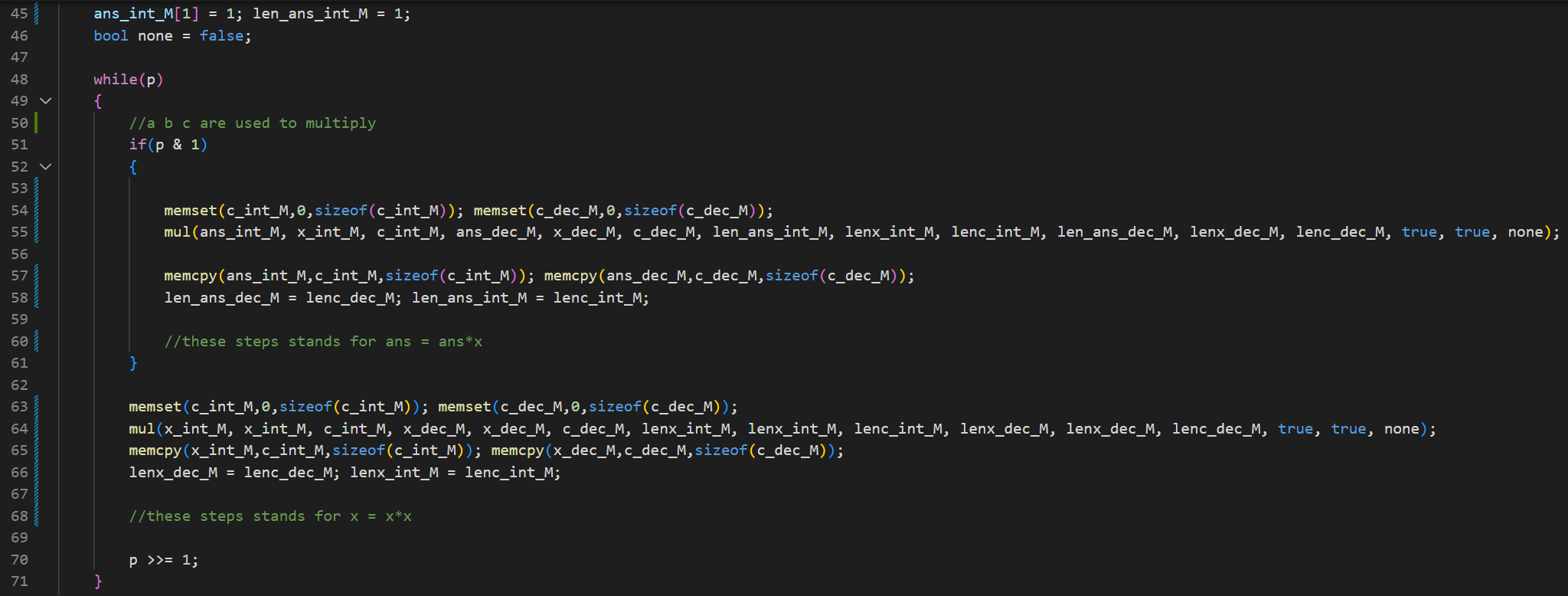
1. 幂次函数实现
   1. 将所有字符串数据转成数字储存



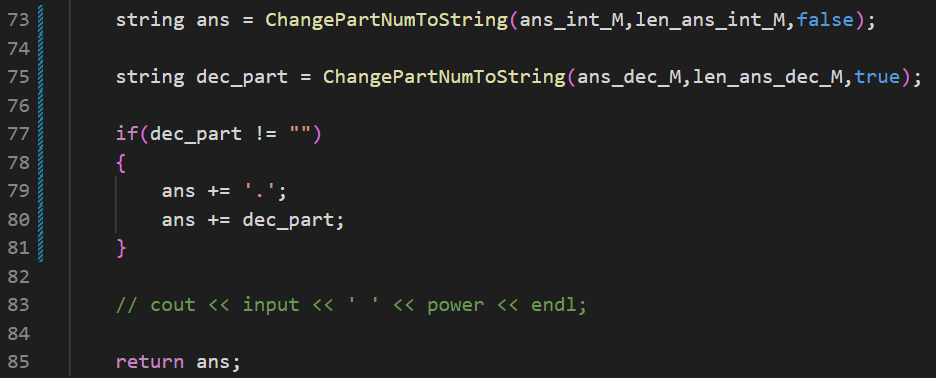
* 1. 使用快速幂思路来解决此问题（此处要求幂次必须是正整数）

（快速幂可参考：

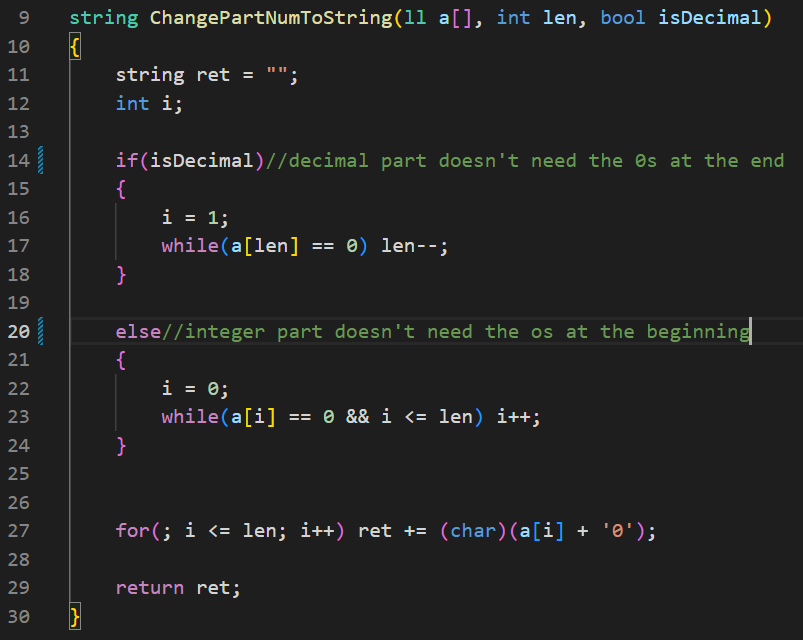
<https://baike.baidu.com/item/%E5%BF%AB%E9%80%9F%E5%B9%82/5500243?fr=aladdin> ）



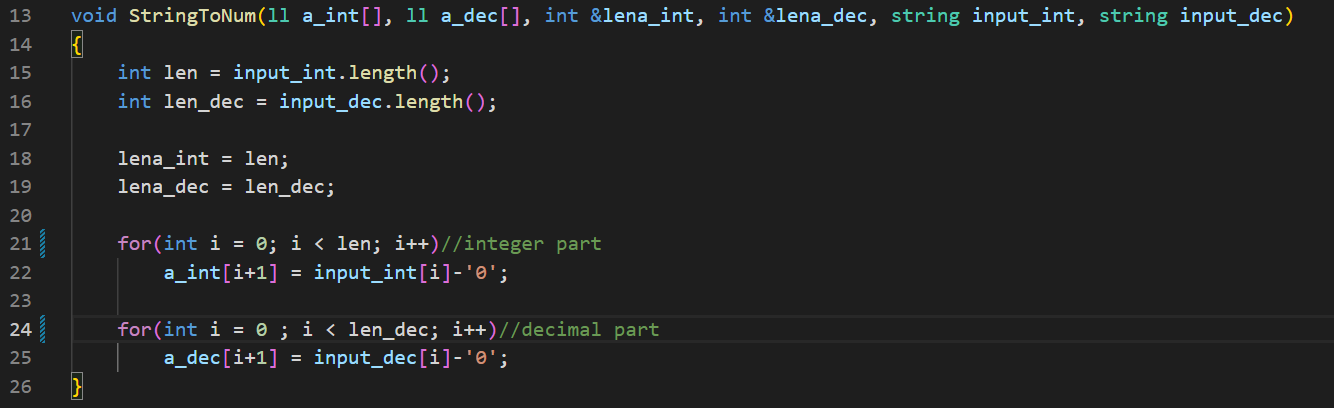
* 1. 最终结果转化为字符串返回



1. 数字转字符串



1. 字符串转数字



# Part IV 程序的局限之处

1. 当前程序运行效率较低，针对高精度的部分目前的算法为n^2(n为位数)，处理的数据位数能力有限
2. 数学函数插入较少
3. 目前暂不支持负数读入，abs函数除外