# 高精度计算程序设计报告

作者：李俊烨 栾子骞

1. **实验目的：**

原本题目要求：用整型数组表示10进制大整数（超过2^32的整数），数组的每个元素存储大整数的一位数字，实现大整数的加减法。

**增加难度**：使用链表完成理论上无限位数的高精度加减法，并且储存的整数有正负之分

1. **实验内容和实验方案**

**1.数据结构：**

**List**

**Head:**找到第一个Node，即最高位

**Tail：**找到最后一个Node，即最低位

**Sign：**表示整数的正负，即符号

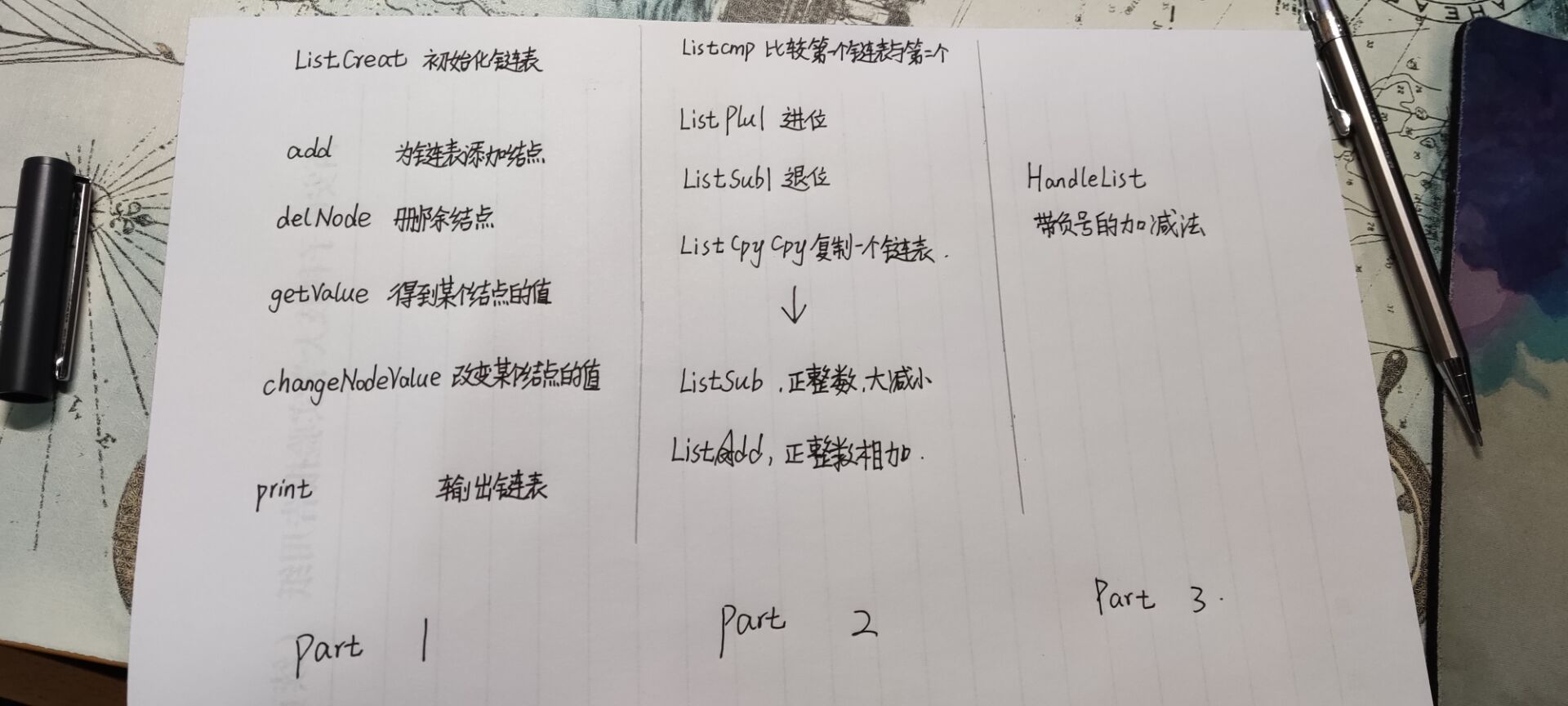
**Node**

next：指向下一个Node

last：指向上一个Node

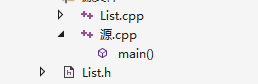
**2.方法实现：**

考虑到链表的加入使得程序变得较为复杂，为了降低开发难度，我采用了先写第一部分函数，实现对链表的基本操作。然后再写第二部分函数，实现对计算的基本实现，再写最终的ListHandle函数，达成目标。这样层层叠加，将复杂问题拆分，从程序到现实问题，从实际代码到抽象计算，循序渐进。



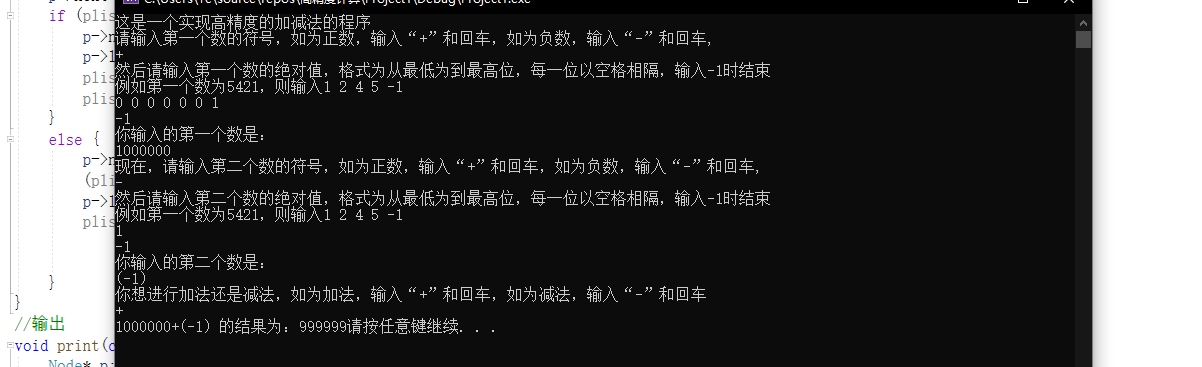
**3.封装**

因为程序相对较小，没有封装成安装包的必要，这里我只把它做成了头文件，方便扩展和他人引用

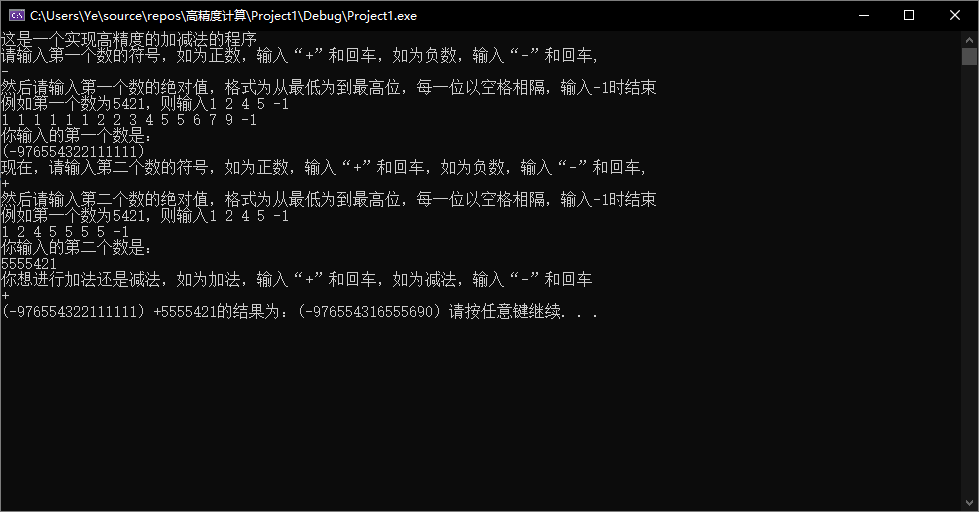


1. **实验效果**

1.



2，



1. **实验结论**

**收获：**

1. 复习了链表的写法，然后自己从单向链表延伸到双向，并加入sign参数表示正负
2. 改正了一些bug，如对于链表的复制，一开始只是简单用等号，然后发现复制后因为Node中存在指针，指针的地址复制过去后，值会随本体变化而变化，所以对指针有了更深入地了解。又如头文件的编写，要把结构的详细声明搬过来，而函数只需抽象声明

**优化思路：**

在编写一些函数时，接受值设成了地址，但大多数情况下这些数据只需查看内容，而不需改变，如ListAdd函数中的加数与被加数，这样方便链表的使用，也降低复制大量数据所带来的内存压力，但是增加了程序的风险，后续希望能使用设const之类方法进行数据保护。