附件1：实验报告中的诚信设计

数据结构与算法导论

实验报告

实验题目：\_\_\_\_实验5：用链表实现大整数加减\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

姓 名：\_\_耿翊中\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

学 号：\_\_\_\_\_\_\_\_\_\_\_\_\_2021213382\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

日 期：\_\_\_\_\_\_\_\_\_\_\_\_\_\_2022.3.27\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

自我评分：\_\_\_\_\_\_\_\_\_【 A 】\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

自我评分说明：A+，A，B+，B，B-，C，D，分别对应分数95、90、85、80、75、70、60

诚信声明

本人郑重承诺：本实验程序和实验报告均是本人独立学习和工作所获得的成果。尽我所知，实验报告中除特别标注的地方外，不包含其他同学已经发表或撰写过的成果；实验程序中对代码工作的任何帮助者所作的贡献均做了明确的说明，并表达了谢意。

如有抄袭，本人原因承担因此而造成的任何后果。

特此声明。

签名：\_\_\_耿翊中\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

日期：\_\_\_\_\_\_\_\_\_2022.3.27\_\_\_\_\_\_\_\_\_\_\_\_\_

程序引用说明

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 引用项 | 来源 | 代码引用行数 |
| 1 | 链表节点声明 | 课本 | 4 |
| 2 | 大数结构体 | <https://blog.csdn.net/Binary_Heap/article/details/77823325> | 4 |
| 3 | 读取字符串并转换为数组形式存贮 | <https://blog.csdn.net/Binary_Heap/article/details/77823325> | 12 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| 小计 | | |  |

总代码行数\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_\_\_\_\_; 引用占比\_\_\_\_\_\_\_11.8%\_\_\_\_\_\_\_\_\_

1、实验简介

【实验内容的简要说明，具体说明实验完成的功能和性能要求】

单独设计算法。在这里，可以用链表每个结点存储大整数的每一位的十进制数字，则可以进行大整数的算数运算，该实验仅实现加减法操作。

2、程序框架

【实验程序共包含哪些函数，一一列举函数名和函数功能】

Linklist(T a[], int n);

构造函数，用长度为n的数组来完成链表的构造

~Linklist();

析构函数

void Printlist();//打印函数

将链表的每一位元素以带空格的方式打印出来，用于检查程序用，由于真正演示结果的函数要对链表数据进行如0092消去00变为92的操作，所以这个原始的打印函数能方便我检查

void Printshow();//展示结果函数

将最终的运算结果展示出来，涉及加位与消0处理

void Scan\_HugeInt()

读取两个字符串并转换为数字存储在数组中

void Plus(Linklist<T>&link1, Linklist<T>&link2, Linklist<T>&link3)

进行加法运算

void Minus(Linklist<T>&link1, Linklist<T>&link2, Linklist<T>&link3)

进行减法运算

Main主函数思路

1读取两个大数

2构造四个链表：数一，数二，加法结果，减法结果

3调用加法函数

4打印加法结果

5调用减法函数

6打印减法结果

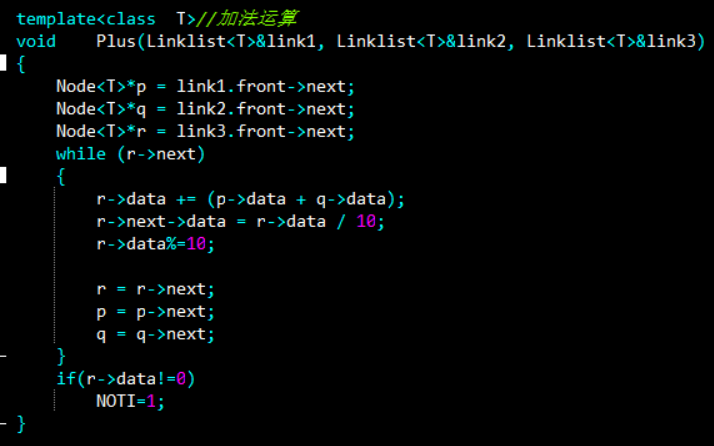
3、关键代实现

【哪些函数是你认为最能体现自己工作成果的函数，说明函数实现基本思想（可用文字或图表示），以及具体的实验步骤（用伪代码或带注释代码）】

3.1 函数名1

void Plus(Linklist<T>&link1, Linklist<T>&link2, Linklist<T>&link3)

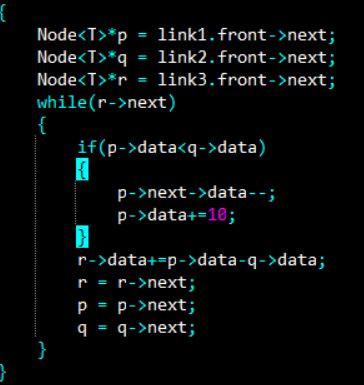
加法算法（实现大数的加法）



用三个节点依次标记三个链表里的结点，通过结果节点整除，判断是否相加超过10；如超过，则提前为下一位节点加一，以得到进位的效果。然后进行模除，留下本节点数字。随后将三个结点都指向下一个节点next；特殊处理：当加法得到数字位数超过原本两数中较大数时，即最后一个节点的next不为0，用全局变量NOTI标记，在打印的时候作特殊标记。

3.2 函数名2

void Minus(Linklist<T>&link1, Linklist<T>&link2, Linklist<T>&link3)



用三个结点分别指向链表1，链表2，结果链表；由于在输入时已经做过调整，所以链表1所代表的数字一定大于链表2所代表的数字，不用担心在作减法的时候出现负数。处理减不够借位：减法前判断两数大小，如不够，则将下一位先减一，然后给本位加十。最后将三位数字都指向next。

3.3 整体实验实现思路

考虑到用链表实现大数的加减法，所想到的首先是用数组将字符串中的数字存储下来，然后用数组初始化链表；考虑到进行操作的方便，所以选择了用双链表的形式。然而如果正向存储的话会迫使加减法时，从链表的末尾向前运算，并不符合从头节点front开始的习惯，所以进行反向存储，正向运算，反向打印。其中加减法的运算用两个以对象为形参的函数实现，将运算结果存储在一个新的链表中。

3.4 几个细节

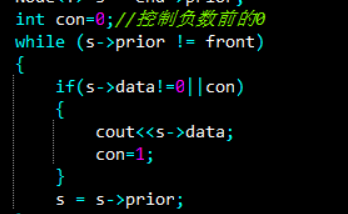
1，将字符串转换为数字存储：减去‘0’的ascall码即可；

2，减法运算是出现负数怎么办：在读取两个大数时就将两数按大小顺序交换，确保在减法时是大数减小数，同时设立一个negative的bool形全局变量，在最终输出的时候如果是负数就在前加负号；

3，加法进位：通过链表的前后继关系实现，具体见上述函数

4，减法借位：通过链表的前后继关系实现，具体见上述函数

5，输出时怎么解决例如0092输出为92，消去前面那些0：



4、不足

【实验程序哪些函数功能还有缺陷或不足，或者程序架构有不足，或者性能还有待提高、或者代码不和规范等等，所有你自己对程序不满意的地方】

1. 加减法函数中仍然依赖全局标量，耦合性高，例如加法中判断位数是否增加，减法判断结果是否为负数；
2. 感觉没有将链表的优势彻底发挥出来，这样进行大数运算首先要依赖数组进行初始化，但所执行功能数组同样可以以相同的效率执行。

5、心得体会

【所有你在实验中的感受和想和老师说的话都可以放在这里，篇幅不限】

加深了对于链表存储结构的理解，编写构造函数析构函数及加减进位借位时在纸上反复画图寻找关系。

考虑问题严谨性：第一遍加减法的程序很快完成，但在多次样例测试中出现了1十二位数加12位数进位到13位数没有考虑到2加法进位后如果还需进位不能连续进位3析构函数重复析构报错的问题。

整体问题比较简单，但由于细节处理和对于面对对象编程的不熟悉不习惯导致还是耗费了比较多的时间。