**实验十三 类的定义与使用**

1. **实验目的和要求**

编写一个程序，定义一个可以处理人一大的正整数类LongLongInt,用一个动态的字符数组存放任意长度的正整数，数组的每个元素存放整型数的一位。成员函数包括：构造函数（根据一个由数字组成的字符串创建一个LongLongInt类的对象），拷贝构造函数、加法函数add，输出函数display。

**二、实验内容**

1. 实验准备
   1. 理论知识介绍
      1. 数据的输入：

要从用户依次读入int类型的变量a,n，可以使用如下语句:

cin >> a>>n;

* + 1. 数据的输出

要将变量的内容显示在显示器上，可以使用cout和流插入运算符。Cout还可以输出表达式的执行结果，比如:

cout << a << ‘+’ << b << ‘=’ << a+b << endl;

* + 1. 算术表达式

C++中，算术运算里有+（加法），-（减法），/（除法），%（取模），但是没有取平方算数运算。

* + 1. 数据类型 （整型）：

C++中一个整型变量可以储存一个整数，可以直接通过cin和cout输入和输出。

* + 1. 数组

一维数组是一个有序数据的集合，数组中的每个元素都有同样的类型。C++中一维数组的定义如下：

类型名 数组名[元素个数]；

C++中的数组的下标从0开始

* + 1. 函数

函数定义的形式如下：

类型名 函数名（形式参数列表）

{

变量定义部分

语句部分

}

* + 1. 类的定义

定义一个类就是定义一组属性和一组堆属性进行操作的函数。属性称为类的数据成员，而函数称为类的成员函数。类定义的一般形式如下：

class 类名{

private:

私有数据成员和成员函数;

public:

公有数据成员和成员函数;

};

* + 1. 对象的使用

在程序中直接定义对象的方法与定义普通函数的方法一样，他的格式如下：

存储类别 类名 对象列表;

对象可以用如下的语句操作：

结构体变量名.成员名

* + 1. 对象的构造

有了构造函数之后，对象的定义的一般形式如下：

类名 对象名(实际参数表)

* + 1. 对象的析构

一般析构函数的定义是：

~类名() {

函数体语句

}

* + 1. 友元

C++在类的定义中可以指定允许某些全局函数、某个其他类的所有成员函数，或某个其他类的某一成员函数直接访问该类的私有成员，他们分别被称为友元函数、友元类和友元成员函数。

友元函数的声明可以通过如下的语句：

friend void f();

友元类的声明如下：

friend class B;

1. 实验项目
   1. 分析
      1. LongLongInt这个类的接口应该包括加法函数Add和输出函数Display，这两个函数可以作为LongLongInt这个类的成员函数，也可以作为他的非成员函数，本程序将其作为非成员函数
      2. 由于Add和Display是LongLongInt的非成员函数，但是它们又要能访问该类中的私有的成员，所以要将它们声明为LongLongInt 的友元函数。
   2. 方案
      1. 需要提供两个文件，一个为long\_long\_int.h,另一个为long\_long\_int.cpp,分别包含LongLongInt类的定义和实现
      2. LongLongInt.cpp里应该还包含一个main函数，用来测试LongLongInt类
      3. 应该在long\_long\_int.h中定义一个LongLongInt类,其中应该包括数字的个数和数组储存的空间这两个私有变量，还应该包括两个重载的构造函数，分别利用一个字符串和另一个long\_long\_int对象构造
      4. 同时由于这个类的储存的空间是动态分配的，所以需要定义一个析构函数来释放该类对象的空间
   3. 测试数据

LongLongInt a("123456789");

LongLongInt b("987654123432");

LongLongInt c = Add(a, b);

LongLongInt d = a;

LongLongInt e("987654");

LongLongInt f = Add(d, e);

Print(c);

Print(f);

* 1. 运行结果

987777580221

124444443

* 1. 异常现象
     1. Print(e)的结果是

000777580221

* + 1. 在未修改过的程序中,Print(f)的结果是

495052525252525251

* 1. 分析
     1. 异常现象1的原因是因为Add函数的实现存在问题，缺少如下的语句

for (int i = rhs.count\_ + 1; i <= lhs.count\_; i++) {

long\_long\_int.storage\_[long\_long\_int.count\_ - i] =

lhs.storage\_[lhs.count\_ - i];

}

这样的话就会存在高位的数字全为0的情况

* + 1. 异常现象2的原因是LongLongInt的一个构造函数

LongLongInt::LongLongInt(char \*ch)

的实现存在问题，因为在将字符的数字转换为整型类型的数字的时候使用了int(ch)这样的语法，导致了0,1,2…依次转换为了48,49,50…

* 1. 总结

在程序中存在数组的情况下，数组的下标的正确的处理永远是易出错的点，所以要注意函数中的数组的下标情况，同时字符和整型的转换也容易出错。

**三、实验小结**

1. 完成情况

完成了该类的所有要求的功能并测试了该类的对象的功能。

1. 重点

类的定义，类的构造函数，类的析构函数，类的友元的定义，类的非成员函数。

1. 难点

Add函数的实现。

1. 解决方法

Add函数的实现需要考虑到两个长度不一样的数组的加法，可以先不考虑进位，将每一位都加起来，然后再对进位进行处理。

1. 有待改进之处

Add函数的实现的代码有些多余，应该还有更好的算法来进行两个长度不一样的数的加法。

1. 收获、体会

在类的定义的过程中，我们也要用到结构化的设计的思想，先在头文件中想好我们需要用到的功能，以及类中的应该有的成员，然后再具体实现这些函数。