**密码学实验二实验报告**

姓名：孟衍璋 学号：16337183

1. **实验目的**

在有限域下，实现一些基本运算——加法、求模、乘法、平方和求逆。

1. **实验内容**

利用自己的学号，进行求逆，然后输出计算值。

如：

学号为2，即表示为：（即：10二进制），然后计算的逆为：

（即：11000000 00001000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 001 二进制）

由左到右，依次为1，x ,…

1. **实验及算法原理**

我在做本实验的过程中，主要是通过阅读助教发给我们的那篇论文与给出的实验参考的PPT才理解了该如何做。所以代码中的算法是复现了论文中的对应算法。

在实现中我使用了bitset结构，它可以用来进行一些状态储存的操作，类似于一个标记数组。移位或者异或运算比较方便。

**加法部分：**

只需要将对应输入的串最低位对齐，依次做按位异或运算。函数输入是string，转换为bitset之后做按位异或，再转换为string输出。

**求模部分：**

将f（x）转为对应的二进制表示形式，然后把输入按64位分组（不足的在右边补零）。之后进行一系列的移位与异或操作，得到结果转换为string输出。

**乘法部分：**

假设相乘的多项式是a与b，得到的结果为c。如果a的最低项次数为1，则将b的值赋给c；如果a的最低项次数为0，则将c全部置0。之后遍历a的每一位，b的值更新为左移一位后模f（x）的结果，如果a[i]为1，将c的值更新为c+b，这里可以调用之前的加法操作。结束之后输出c即为结果。

**平方部分：**

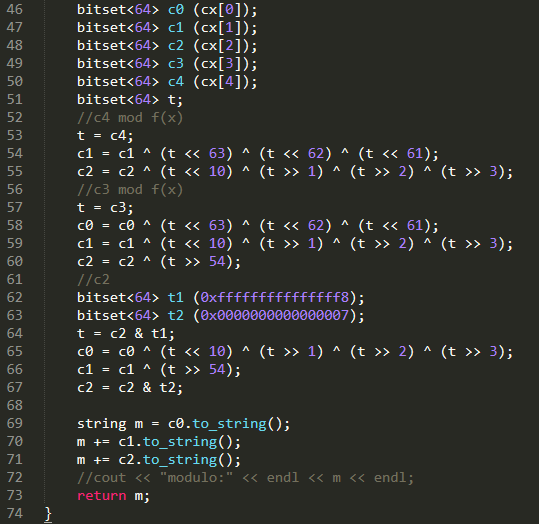
直接调用乘法运算，其中两个系数为要平方的多项式。

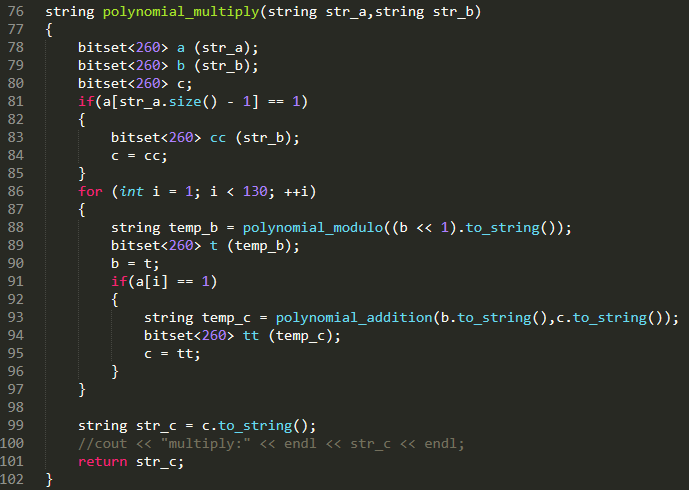
**求逆部分：**

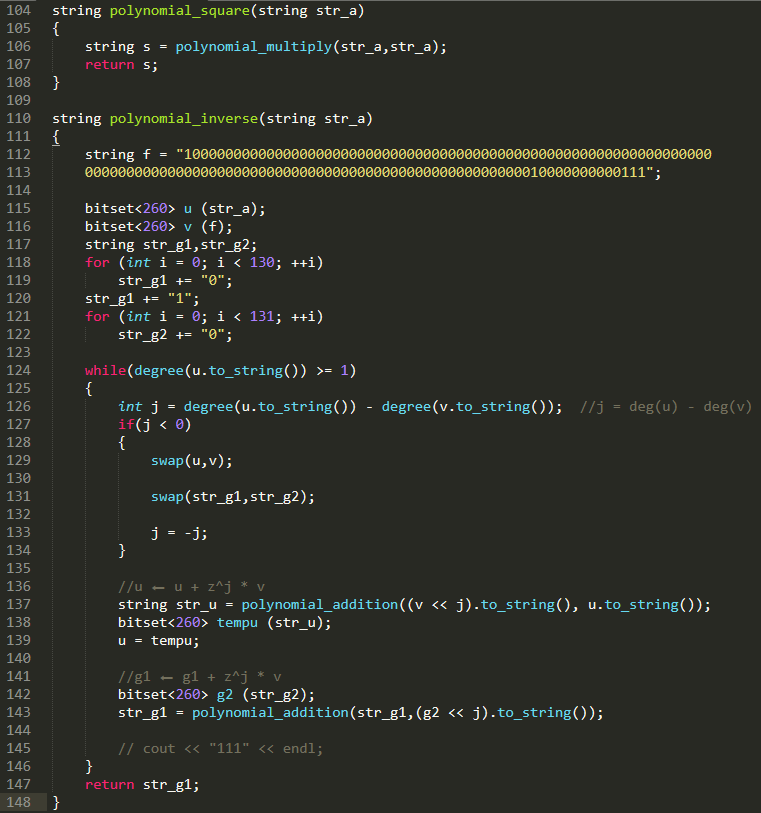
首先要会求两个多项式的最大公因式的操作，这里采用欧几里得算法。之后便利用gcd（a，f） = 1，求出输入a的逆。

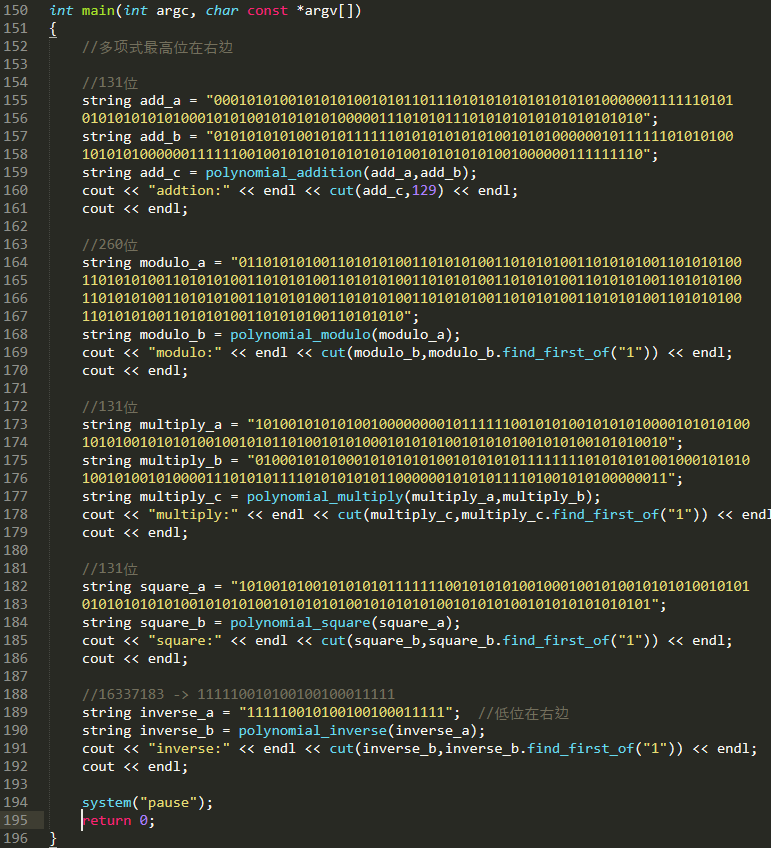
1. **实验代码**



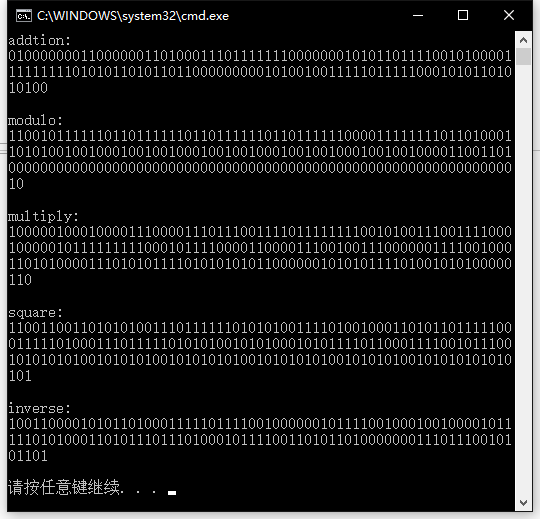








1. **运行截图**



1. **结果分析**

输出依次是加法、求模、乘法、平方、求逆运算的结果。其中求逆运算的输入是将自己的学号转为二进制之后输入进去得到的结果。

1. **总结**

这次的算法部分都是根据论文里给出的算法来完成的。主要的挑战就是看懂论文，其实看了之后实现起来也相对不是那么复杂。我用了bitset的结构，在实验过程中因为申请的空间不够大，导致赋值不成功，所以求逆运算一直有很大的问题，后来发现这个问题之后改大空间之后就解决了。

这次的实验让我对有限域的概念有了进一步了解，实际操作之后理解更加深刻了。