# 第 3 次编程练习报告

姓名：武桐西　　学号：2112515　　班级：信安一班

## 一、编程练习 1——中国剩余定理(CRT)

### ➢ 源码部分：

```
#include<iostream>
#include<vector>
using namespace std;

bool isCoprime(int a, int b) {
    //辗转相除法求最大公因数，判断a,b是否互素
    a = a > 0 ? a : -a;//转为正数
    b = b > 0 ? b : -b;//转为正数
    if (a < b) {//令a为较大者，b为较小者
        int tmp = a;
        a = b;
        b = tmp;
    }
    while (a % b) {//余数不为0，继续循环
        int r = a % b;
        a = b;
        b = r;
    }
    return b == 1;
}

int Euclid(int a, int b) {
    //扩展欧几里得算法，求a模b的乘法逆元
    vector<int> r;//余数序列
    r.push_back(a > b ? a : b);//a, b中的大者
    r.push_back(a < b ? a : b);//a, b中的小者
    vector<int> q;//商序列
    q.push_back(-1);//q[0]中的值无效
    vector<int> s;
    s.push_back(1);
    s.push_back(0);
    vector<int> t;
    t.push_back(0);
```

```
    t.push_back(1);

    int x = 0;//索引
    while (r[x] % r[x + 1]) {//余数非零，则循环
        r.push_back(r[x] % r[x + 1]);
        q.push_back(r[x] / r[x + 1]);
        s.push_back(s[x] - s[x + 1] * q[x + 1]);
        t.push_back(t[x] - t[x + 1] * q[x + 1]);
        x++;
    }

    int l = r.size() - 1;//序列的末尾元素下标
    if (r[l] == 1) {
        //乘法逆元存在
        if (a > b) {//根据a, b的大小讨论
            //转为最小正缩系中
            if (s[l] < 0)
                s[l] = b + s[l];
            if (t[l] < 0)
                t[l] = a + t[l];
            return s[l];
        }
        else {
            //转为最小正缩系中
            if (s[l] < 0)
                s[l] = a + s[l];
            if (t[l] < 0)
                t[l] = b + t[l];
            return t[l];
        }
    }
    return 0;//gcd(a, b) != 1, 乘法逆元不存在
}


bool CRT(int n, int* b, int* m) {
    //Chinese Remainder Theorem
    for (int i = 0; i < n; i++)
        for (int j = i + 1; j < n; j++)
            if (!isCoprime(m[i], m[j]))
                return false;//模数不互素，不符合CRT的适用条件，返回false
    //Now, all m[i] and m[j] (where i != j) is Coprime to each other
    int M = 1;//保存 \sum m[i]
    for (int i = 0; i < n; i++)
```

```
        M *= m[i];
    int Ans = 0;//保存结果
    for (int i = 0; i < n; i++) {
        Ans += (M / m[i]) * Euclid((M / m[i]), m[i]) * b[i];
        Ans %= M;
    }
    cout << "x ≡ " << Ans << " (mod " << M << ")\n";
    return true;
}

int main() {
    int n;
    cout << "n = ";
    cin >> n;
    int* b = new int[n];
    int* m = new int[n];
    for (int i = 0; i < n; i++) {
        cout << "  b_" << (i + 1) << " = ";
        cin >> b[i];
        cout << "  m_" << (i + 1) << " = ";
        cin >> m[i];
    }
    if (!CRT(n, b, m))
        cout << "模数不满足两两互素的条件，CRT不适用!\n";
    system("pause");//暂停窗口
    return 0;
}
```

- **说明部分：**

中国剩余定理（CRT）的程序主要分为以下三个部分：

(1) 判断模数是否两两互素：

利用辗转相除法，求每两个模数的最大公因子，若最大公因子为1，则二者互素；否则不互素，此时不符合中国剩余定理（CRT）的使用条件，输出提示信息。

用形式化语言描述如下：

For any $m_i$ and $m_j$ $(i \neq j)$, Check if $\gcd(m_i, m_j) == 1$.

IF $\gcd(m_i, m_j) \neq 1$:

return false

End IF

End For

return true

(2) 求$M_i$的模$m_i$的乘法逆元：

利用扩展欧几里得算法，求解$M_i$的模$m_i$的乘法逆元$M_i^{'}$。

(3) 线性同余方程组求解：

利用$x \equiv \sum_{i=1}^{n} M_i^{'} M_i b_i (\mathrm{mod}\, m)$，得出线性同余方程组的解。

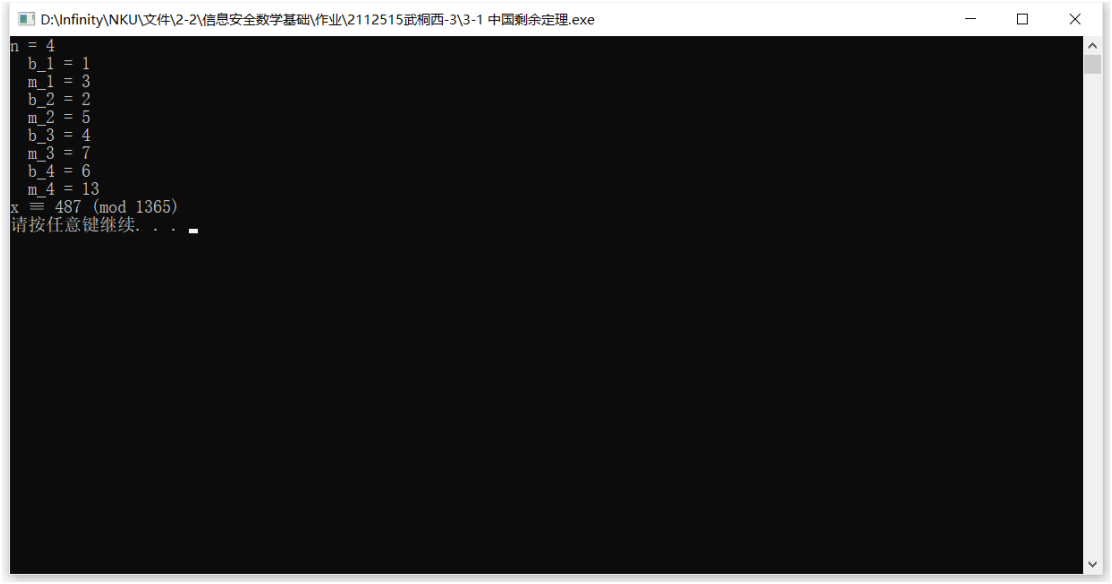
*Tricks*：

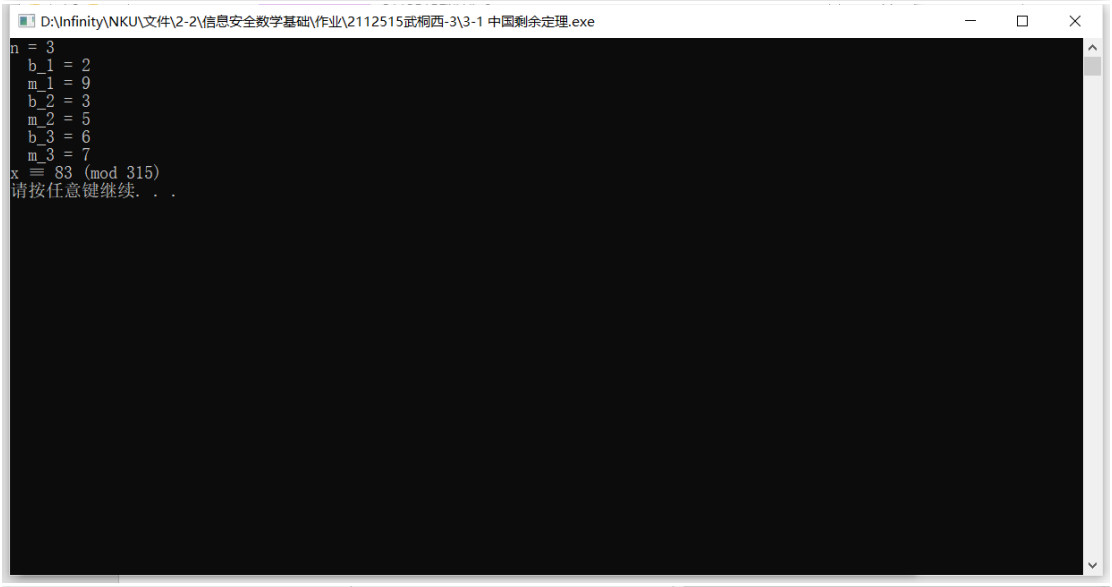
① 可以先计算并保存下变量$m = \prod_{i=1}^{n} m_i$，然后当计算$M_i = m_1 m_2 \cdots m_{i-1} m_{i+1} \cdots m_n$时，可以直接利用$M_i = \frac{m}{m_i}$进行计算，从而使得计算量大大减少，且无需重复计算，提高了程序的性能。

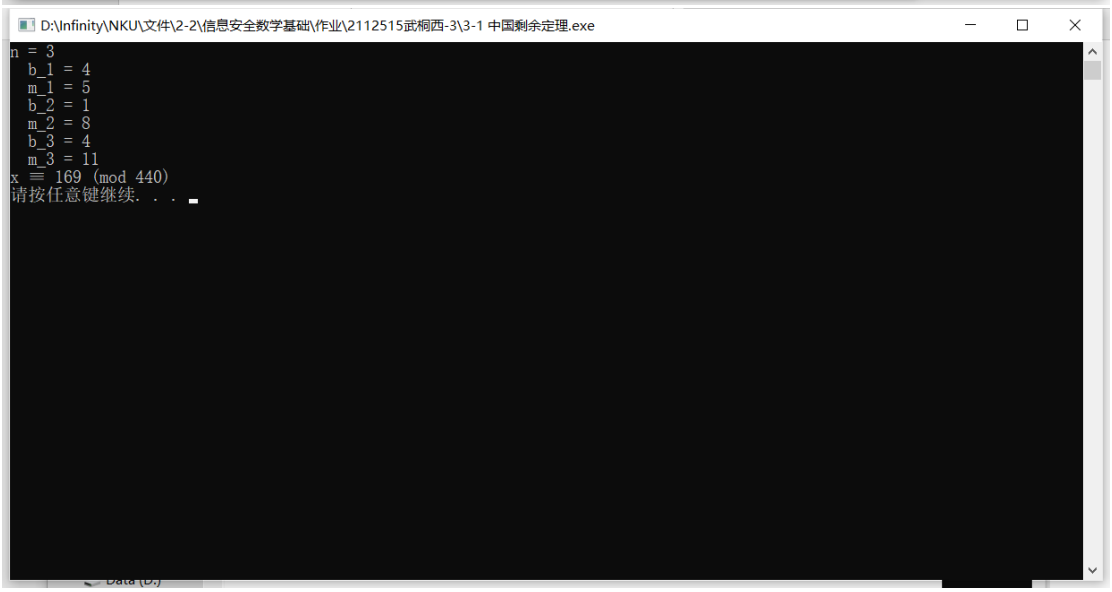
② 在计算方程组的解时，每加一项$M_i^{'} M_i b_i$，可以将当前所得结果模$m$，以加快计算，同时防止数据溢出。

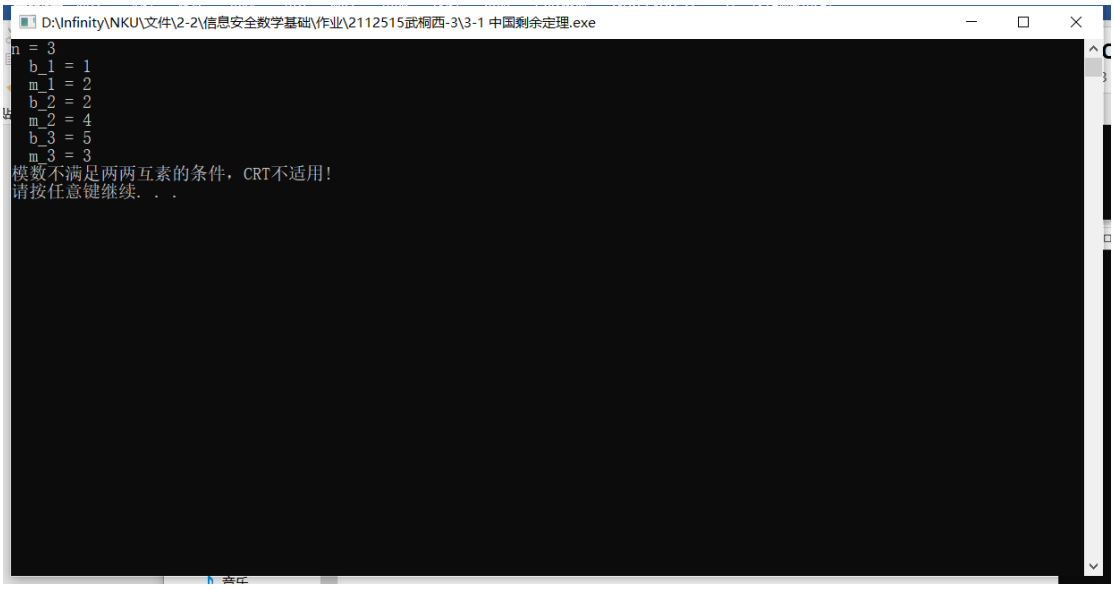
注意事项：为了直观方便，本程序的输入顺序与样例的输入顺序不同，烦请老师或者助教学长在核验程序结果时注意本程序的输入顺序。

➢ 运行示例：


D:\Infinity\NKU\文件\2-2\信息安全数学基础\作业\2112515武桐西-3\3-1 中国剩余定理.exe
n = 4
b_1 = 1
m_1 = 3
b_2 = 2
m_2 = 5
b_3 = 4
m_3 = 7
b_4 = 6
m_4 = 13
x ≡ 487 (mod 1365)
请按任意键继续. . .


D:\Infinity\NKU\文件\2-2\信息安全数学基础\作业\2112515武桐西-3\3-1 中国剩余定理.exe
n = 3
b_1 = 2
m_1 = 9
b_2 = 3
m_2 = 5
b_3 = 6
m_3 = 7
x ≡ 83 (mod 315)
请按任意键继续. . .


D:\Infinity\NKU\文件\2-2\信息安全数学基础\作业\2112515武桐西-3\3-1 中国剩余定理.exe
n = 3
b_1 = 4
m_1 = 5
b_2 = 1
m_2 = 8
b_3 = 4
m_3 = 11
x ≡ 169 (mod 440)
请按任意键继续. . .


D:\Infinity\NKU\文件\2-2\信息安全数学基础\作业\2112515武桐西-3\3-1 中国剩余定理.exe
n = 3
b_1 = 1
m_1 = 2
b_2 = 2
m_2 = 4
b_3 = 5
m_3 = 3
模数不满足两两互素的条件，CRT不适用!
请按任意键继续. . .