# 第3次编程练习报告

姓名：胡博浩 学号：2212998 班级：信息安全

##### 编程练习1——中国剩余定理

* **源码部分：**

#include <iostream>

#include <vector>

using namespace std;

//扩展欧几里得算法求模逆元

int extend\_Euclid(int a, int b, int& x, int& y) {

int x0 = 1, y0 = 0, x1 = 0, y1 = 1;

while (b) {

int q = a / b;//计算商和余数

int temp = b;

b = a % b;

a = temp;

temp = x1;//更新x和y

x1 = x0 - q \* x1;

x0 = temp;

temp = y1;

y1 = y0 - q \* y1;

y0 = temp;

}

x = x0;//返回最大公约数，并更新x和y

y = y0;

return a;

}

//中国剩余定理求解同余方程组的解

int CRT(vector<int>& a, vector<int>& m) {

int M = 1;//计算模数的乘积

for (int mi : m) {

M \*= mi;

}

int x = 0;

for (int i = 0; i < m.size(); ++i) {

int Mi = M / m[i];

int x0, y0;

extend\_Euclid(Mi, m[i], x0, y0);

x += a[i] \* Mi \* x0;

}

return x % M;

}

int main() {

int n;

cout << "n=";

cin >> n;

vector<int> b(n), m(n);//创建vector存储方程组的系数和模数

for (int i = 0; i < n; ++i) {

cout << " b\_" << i << "=";

cin >> b[i];

}

int total = 1;//计算模数

for (int i = 0; i < n; ++i) {

cout << " m\_" << i << "=";

cin >> m[i];

total \*= m[i];

}

int x = CRT(b, m);

cout << "x≡" << x << " (mod " << total << ")";

system("pause");

return 0;

}

* **说明部分：**

这段代码实现了中国剩余定理（CRT），可以计算同余方程组。

1. extend\_Euclid函数：计算模逆元

采用扩展欧几里得算法，通过迭代计算最大公约数的同时更新两个变量x和y，最终返回最大公约数，并将模逆元存储在参数x 和y中。

2. CRT函数：实现中国剩余定理

两个变量作为输入，一个是系数a，另一个是模数m。首先，计算所有模数的乘积M。然后，对于每个同余方程，计算除去当前模数后的总模数Mi，并调用extend\_Euclid函数来找到相应模数的模逆元。最后，根据中国剩余定理的公式计算出x的值，并返回结果。

3. main函数：

在main函数中，首先询问用户输入同余方程的数量n。然后，循环n次，提示用户输入每个方程的系数和模数，并将它们存储在相应的向量b和m中。接下来，计算所有模数的乘积，作为总模数。最后，调用CRT函数来计算解，并打印出结果。

* **运行示例：**

