# 第3次编程练习报告

姓名：刘修铭 学号：2112492 班级：信安一班

##### 一、编程练习1——中国剩余定理

* **源码部分：**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

void extended\_euclidean\_algorithm(int a, int b, int\* gcd, int\* x, int\* y) {

// a和b的最大公约数

int r, q, x1, x2, y1, y2;

if (b == 0) {

\*gcd = a;

\*x = 1;

\*y = 0;

return;

}

x2 = 1; x1 = 0; y2 = 0; y1 = 1;

while (b > 0) {

q = a / b;

r = a - q \* b;

\*x = x2 - q \* x1;

\*y = y2 - q \* y1;

a = b;

b = r;

x2 = x1;

x1 = \*x;

y2 = y1;

y1 = \*y;

}

\*gcd = a;

\*x = x2;

\*y = y2;

}

int chinese\_remainder\_theorem(int\* a, int\* m, int n) {

int M = 1, x = 0, gcd, y;

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

M \*= m[i];

}

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

int Mi = M / m[i];

extended\_euclidean\_algorithm(Mi, m[i], &gcd, &y, &y);

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

}

return x % M;

}

int main() {

int n;

printf("请输入同余方程个数n：");

scanf\_s("%d", &n);

int\* a = (int\*)malloc(n \* sizeof(int));

int\* m = (int\*)malloc(n \* sizeof(int));

printf("请输入同余方程：\n");

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

printf("x ≡ ");

scanf\_s("%d", &a[i]);

printf("(mod ");

scanf\_s("%d", &m[i]);

printf(")\n");

}

int x = chinese\_remainder\_theorem(a, m, n);

printf("x ≡ %d (mod ", x);

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

printf("%d", m[i]);

if (i != n - 1) {

printf(",");

}

}

printf(")\n");

free(a);

free(m);

system("pause");

return 0;

}

* **说明部分：**

这段代码实现了中国剩余定理。

1.kr和kq为全局变量，分别记录循环中商和余数数组的下标，初始值为2和1。它们在循环中用于记录每次欧几里得算法迭代的商和余数。

2.ojld函数实现了欧几里得算法，用于计算两个数的最大公约数。参数a和b为需要计算最大公约数的两个数，res数组用于记录每次欧几里得算法迭代的余数，q数组用于存储每次迭代的商，s和t数组用于存储最终得到的贝祖等式中的系数。函数中，首先将s和t的初值设置为1和0以及0和1，分别对应于上一次和这一次迭代的结果，然后将a和b取绝对值并分别存储在max和min变量中。接着，在while循环中进行欧几里得算法迭代计算，直至余数为0。在每次迭代中，计算商和余数，并将商存储在q数组中，将余数存储在res数组中，然后根据贝祖等式迭代计算s和t，并将它们存储在相应的数组中。最后，更新max和min的值，将max设置为原来的min，将min设置为余数。在每次迭代结束后，kr和kq分别自增1，以便记录下一次迭代的商和余数。

3.在main函数中，首先输入需要计算的方程组的个数n，以及每个方程的b和m的值，分别存储在动态分配的b和m数组中。然后，根据所有方程的m值计算出整个方程组的M值，存储在动态分配的M数组中。接着，动态分配所需的数组空间，并定义一个变量x，用于存储最终结果。在接下来的循环中，调用ojld函数计算出每个方程中的s和t值，并根据中国剩余定理的公式，计算出方程组的解。循环结束后，输出结果并释放动态分配的数组空间。

4.在输出结果前，需要对s和t的值进行一些调整。如果s或t为负数，则需要加上一个b或m的值，以保证它们为正数。

* **运行示例：**

