编程练习-2012431-李佩诺

1. 编写计算正整数欧拉函数的程序
2. 程序代码
3. 编程实现中国剩余定理

#include<iostream>

using namespace std;

bool sushu(int x) {

//判断是否为素数

if (x == 2) { return 1; }

for (int i = 2; i <= x-1; i++) {

if (x % i == 0) {

return 0;

}

}

return 1;

}

int f(int x) {

//因数的乘积

// p-1/p

int k = x ;

int s1 = 1;//分子 p-1的乘积

int s2 = 1;//分母 p的乘积

for (int i = 2; i <= k; i++) {

if (x % i == 0) {

if (sushu(i) == 1) {

s1 \*= (i-1);

s2 \*= i;

}

}

}

return x \* s1 / s2;

}

int main()

{

int x;

cout << "请输入一个正整数：";

cin >> x;

int result = f(x);

cout << "欧拉函数的值为：" << result<<endl;

}

1. 运行结果测试



1. 编程实现中国剩余定理
2. 程序代码

#include<iostream>

#include<cmath>

using namespace std;

bool sushu(int x) {

//判断是否为素数

if (x == 2) { return 1; }

for (int i = 2; i <= x-1; i++) {

if (x % i == 0) {return 0;}

}

return 1;}

int f(int x) {

//欧拉函数

// p-1/p

int k = x ;

int s1 = 1;//分子 p-1的乘积

int s2 = 1;//分母 p的乘积

for (int i = 2; i <= k; i++) {

if (x % i == 0) {

if (sushu(i) == 1) {

s1 \*= (i-1);

s2 \*= i;

}

}

}

return x \* s1 / s2;

}

int niyuan(int M,int b,int m) {

//求乘法逆元

M = M % m;

long long N = pow(M, (f(m)-1)) \* b;

N = N % m;

return N;

}

int main() {

// MN=b(mod m) -> N=(M,f(M))\*b

cout << "请输入方程个数：";

int n;

cin >> n;

int\* M = new int[n];

int\* N = new int[n];

int\* b = new int[n];

int\* m = new int[n];

cout << "请依次输入余数b和除数m：" << endl;

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

cout << "b" << i+1 << " m" << i+1 << " :";

cin >> b[i] >> m[i];

}

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

M[i] = 1;

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

if (i != j) { M[i] \*= m[j]; }

}

}

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

N[i] = niyuan(M[i], 1, m[i]);

}

int x=0;//最终解

int ms=1;

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

ms \*= m[i];

x += N[i] \* M[i] \* b[i];

}

cout << "最终解为 ： x≡" << x%ms<<" (mod "<<ms<< ")";

delete[]M;

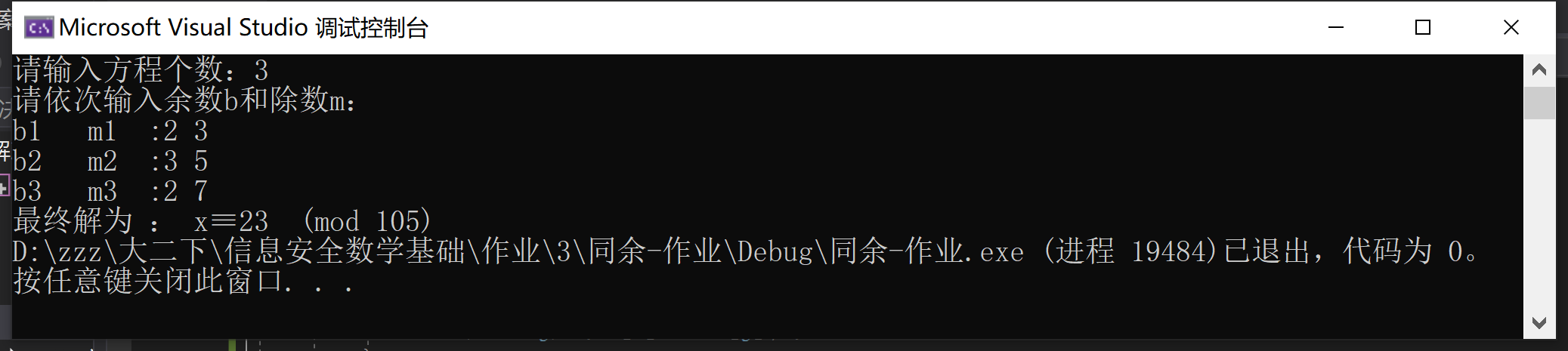
delete[]N;

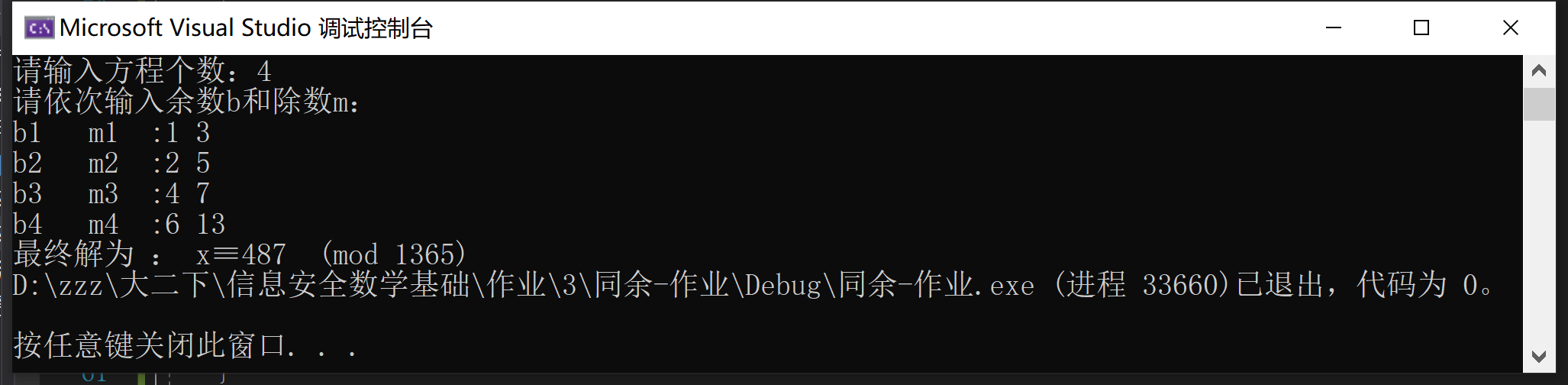
delete[]b;

delete[]m;

}

(2)运行结果测试





经检验全部正确。