# 第2次编程练习报告

姓名：张丛 学号：2113662 班级：信安一班

##### **编程练习1——平方-乘算法**

* **源码部分：**

#include<iostream>

using namespace std;

int detobi(int a[],int x) //返回二进制位数//a[]储存结果

{

int k = 0;

while (x / 2) //除2取余，倒排余数

{

a[k] = x % 2;

k++;

x = x / 2;

}

a[k] = x; //储存最后一个余数

return k;

}

int pc(int a, int m, int n) //a^m(modn)//返回结果

{

int c = 1;

int bi[50]; //存储二进制

int k = 0; //二进制位数

k = detobi(bi, m);

for (int i = k; i >= 0; i--) //平方-乘算法

{

c = c \* c % n;

if (bi[i] == 1)

{

c = c \* a % n;

}

}

return c;

}

int main()

{

cout << "Calculate a^n(mod m)..." << endl << "Please input:" << endl;

int a, n, m;

cin >> a >> n >> m;

cout << "a=" << a << endl;

cout << "n=" << n << endl;

cout << "m=" << m << endl;

int c = pc(a, n, m); //调用平方-乘算法;

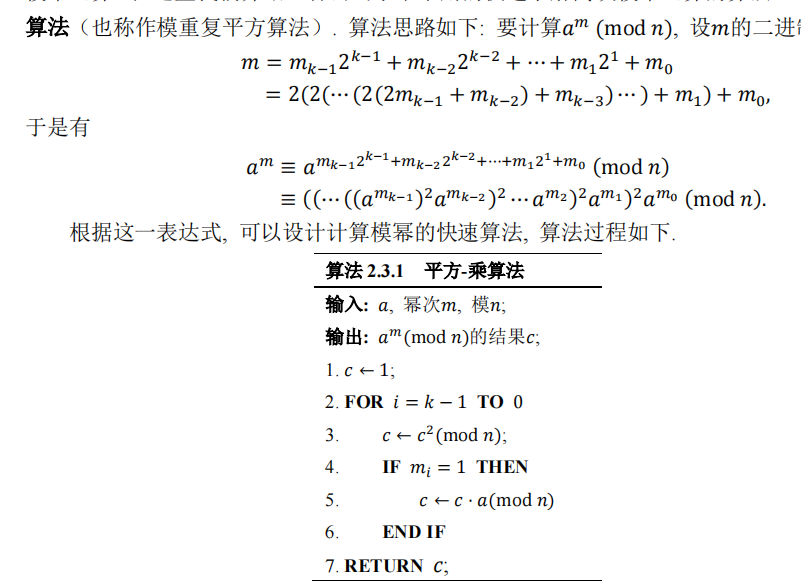
cout << a << "^" << m << " (mod " << n << ")=" << c << endl;

system("pause");

return 0;

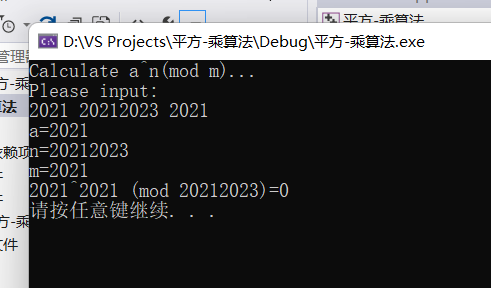
* }
* **说明部分：**

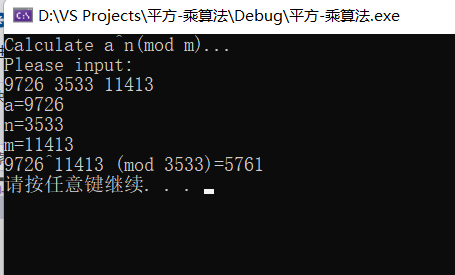
代码实现平方-乘算法：



主要编写部分在于，幂次m的十进制转二进制函数（采用除2取余法），和平方-乘算法主函数

* **运行示例：**





##### **编程练习2——扩展欧几里得算法**

* **源码部分：**

#include<iostream>

using namespace std;

void Euclid(int a,int b,int& gcd,int& lcm,int& an,int& bn) //欧几里得算法求逆元

{

int A = a, B = b; //记录一下初始的a和b，传值时不一定a>b

if (a < 0) //a或b为负，则取绝对值

{

a \*= -1;

}

if (b < 0)

{

b \*= -1;

}

bool flag\_ab = 1; //若初始a>b，则为1

if (a < b)

{

swap(a, b);//交换a,b使得a>b

flag\_ab = 0; //若初始a<b,flag为0

}

int r[20] = { 0 };//存余数

r[0] = a; r[1] = b;

int k = 2;

while (true) //欧几里得算法

{

if (a % b == 0)

{

break;

}

r[k] =a%b;

k++;

a = b;

b = r[k - 1];

}

k--;

gcd = r[k]; //最大公因数//但因注意，两数互素时乘法逆元才有意义

lcm = r[0] \* r[1] / gcd; //最小公倍数

//求逆元

int s[20], t[20], q[20];

s[0] = 1; s[1] = 0;

t[0] = 0; t[1] = 1;

q[0] = -1;

for (int i = 1; i <= k; i++)

{

q[i] = r[i - 1] / r[i];

}

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

{

s[i] = s[i - 2] - q[i - 1] \* s[i - 1];

t[i] = t[i - 2] - q[i - 1] \* t[i - 1];

}

an = s[k];

bn = t[k];

if (flag\_ab==0)//如果a,b交换过，逆元相应交换

{

swap(an, bn);

}

while (an < 0)//逆元<0时，加模数使为正

{

an += B;

}

while (bn < 0)

{

bn += A;

}

return;

}

int main()

{

cout << "Please input a,b:" << endl;

int r0, r1;

cin >> r0 >> r1;

cout << "a=" << r0 << endl;

cout << "b=" << r1 << endl;

int gcd, lcm, r0n, r1n;

Euclid(r0, r1, gcd, lcm, r0n, r1n);//调用函数

cout<<"gcd(a,b)=" << gcd << endl;

cout<<"lcm(a,b)=" << lcm << endl;

cout<<"a^(-1)=" << r0n <<" (mod "<<r1<<")"<< endl;

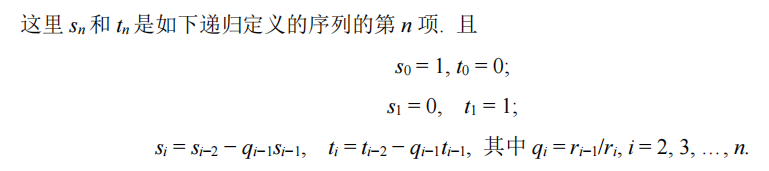
cout<<"b^(-1)=" << r1n <<" (mod "<<r0<<")"<< endl;

system("pause");

return 0;

* }
* **说明部分：**

编程实现扩展欧几里得算法，主要难度在于求逆元过程：用数组r[]记录余数，后导出数组q[],再求出数组s[]和t[]。关系如下：



其中数组的下标需要注意。

另外的细节有，算法中的r0>r1，传值时却并不一定，所以可能有交换，最后的逆元结果也要交换。逆元为负时，找最小的正同余。

因为两数互素时才有逆元，故在求逆元的情况下gcd(a,b)=1，lcm=a\*b。

* **运行示例：**

