欧几里得算法实验

       学号：17116060110  姓名：宋子凯

1. 欧几里得算法原理

欧几里得算法是初等数论中的一个基本算法，也是密码学中最常用的算法之一。它利用辗转相除法求得另个给定整数r0、r1的最大公约数，还可以将辗转相除法的过程倒退回去，利用r0和r1的线性组合表示其最大公约数。进而当两者互素（最大公约数为1）时，可以求得各自相对应的乘法模逆元(扩展的欧几里得算法）。

欧几里得算法：设a=qb+r，其中a，b，q，r都是整数，则gcd(a,b)=gcd(b,r)，即gcd(a,b)=gcd(b,a%b)。

扩展欧几里得算法：对于不完全为 0 的非负整数 a，b，gcd（a，b）表示 a，b 的最大公约数，必然存在整数对 x，y ，使得 gcd（a，b）=ax+by。

1. 算法实现（要求具体的代码及必要的注释说明）

#include<stdio.h>

int gcd(int i,int j){//定义求最大公因数的函数，有两个参数i,j是要求最大公因数的两个数

while(j!=0)

{

int r=j; //定义r为中间变量，用于交换i,j的值

j=i%j; //求i,j的余数

i=r;

} //根据欧几里得算法，gcd(i,j)=gcd(j,i%j)，在j！=0的条件下循环计算

return i;

}

int main(){

int i,j;

printf("input i,j:");

scanf("%d %d",&i,&j);

printf("gcd of i,j is:%d",gcd(i,j));

return 0;

}

#include<stdio.h>

int exgcd(int m,int n,int &x,int &y)//扩展欧几里得算法

{

int x1,y1,x0,y0;

x0=1; y0=0;

x1=0; y1=1;

x=0; y=1; //初始化x0,y0,x1,y1

int r=m%n; //令r=i mod j

int q=(m-r)/n;

while(r)

{

x=x0-q\*x1; y=y0-q\*y1;

x0=x1; y0=y1; //

x1=x; y1=y; //x0=y1,y0=x1-(m/n)y1

m=n; n=r; r=m%n;

q=(m-r)/n;

} //可以求出x,y的值

return n; //返回gcd(m,n)

}

int mod\_reverse(int a,int n)//ax=1(mod n) 求a的逆元x

{

int d,x,y;

d=exgcd(a,n,x,y); //调用扩展欧几里得算法求最大公因数

if(d==1) //如果做大公因数是1，gcd(a,n)=1则存在逆元

return (x%n+n)%n; //求a的逆相当于求解ax=1(mod n),这个方程可以转化为ax-my=1,

//然后套用二元一次方程的方法，用扩展欧几里得算法求得一组x0,y0和gcd

else

return -1;

}

int main(){

int i,j;

printf("input a,n(ax=1(mod n)):") ;

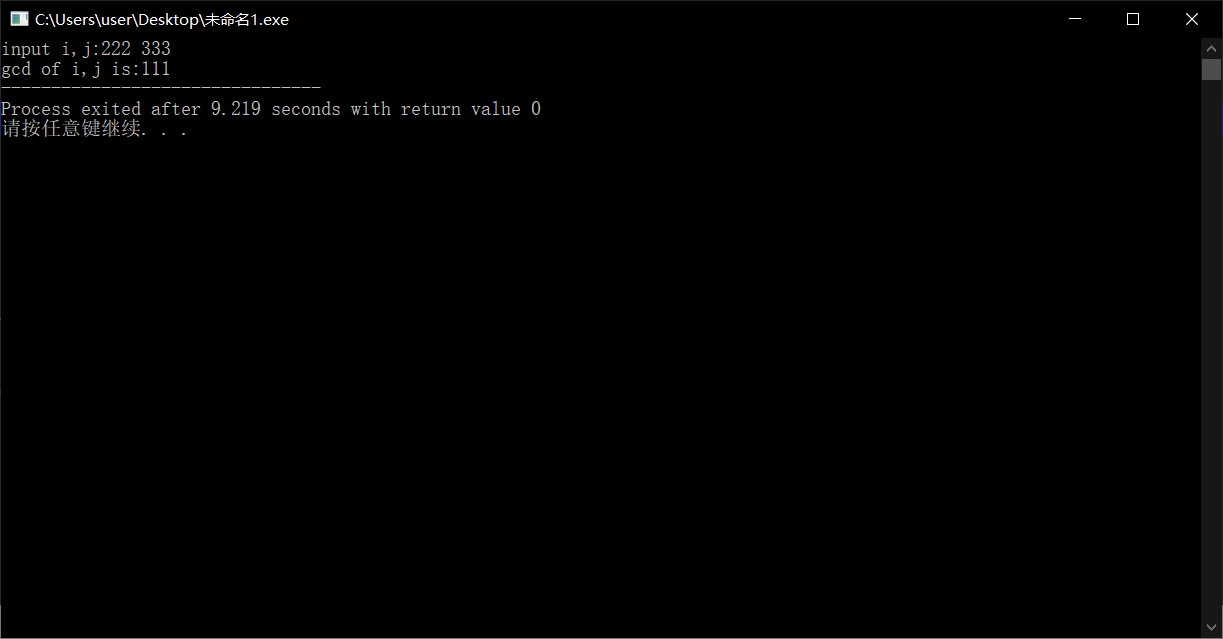
scanf("%d%d",&i,&j);

printf("mod\_reverse a is:%d",mod\_reverse(i,j));

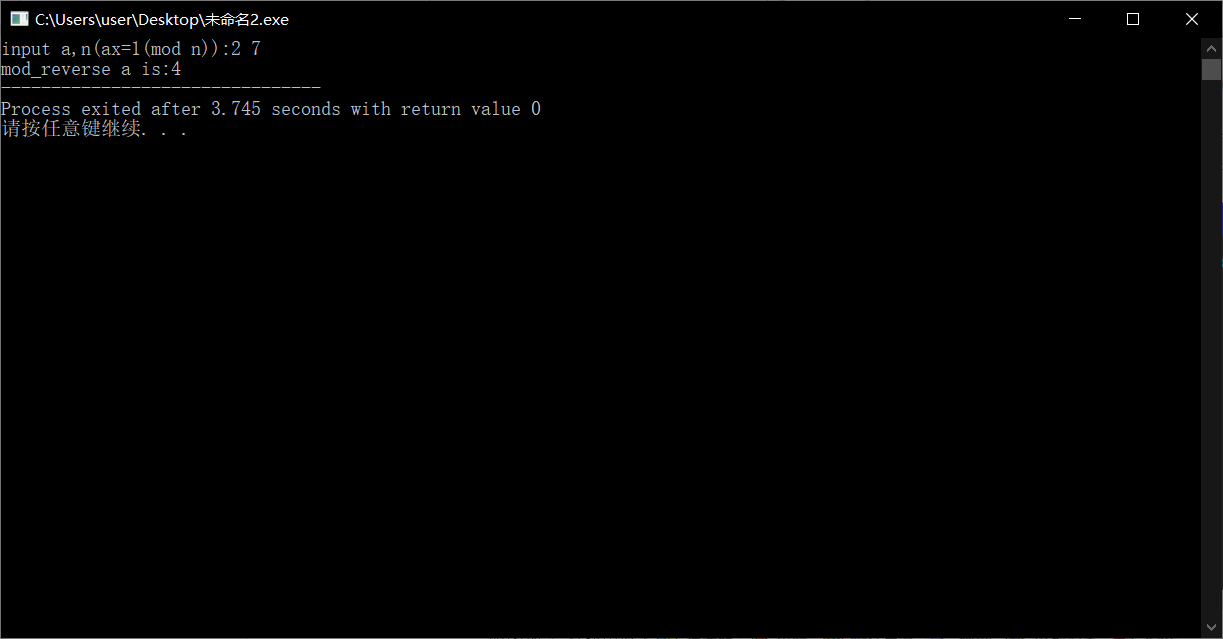
return 0;

}

1. 最大公约数及逆元求解演示（文字描述及相应的运行截图）



如图所示为利用欧几里得算法求222与333的最大公因数111



如图所示为利用扩展欧几里得算法求2模7的逆元等于4

1. 总结体会

通过这次试验，我明白了欧几里得算法与扩展欧几里得算法的原理以及具体的实现代码，并学会了使用扩展欧几里得算法求逆元，以及不定方程的解。