欧几里得算法是辗转相除求ab最大公约数，直到b==0：结束，a就是最大公约数

比如求7和5；

7%5=2

7和5最大公约数等价于5和2

5%2=1；

5和2最大公约数等价于2和1  
2%1=0

2和1最大公约数等价于1和0

任何数和0最大公约都是那个数本身；

gcd(7,5)=gcd(1,0)=1；

结束

## 扩展欧几里得：

注意条件：a\*b>=0;且a和b不同时得0

它是求解关于x和y的方程ax+by=gcd(a,b)通解和特解，

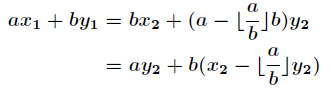
当a=0时；b=1;

当b=0时；a=1;

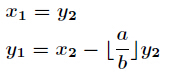
其他，类似求最大公约数方法 b替换a;a%b替换b;

http://img.blog.csdn.net/20140614204840046

http://img.blog.csdn.net/20140614204955390，所以进一步得到



所以当进行了一次替换后；

一直进行这个步骤，直到b是0

程序里用递归求解，类似普通的求最大公约数，而且每次递归要对x1赋值y2,对y1赋值成x2-a/b\*y2; x1和y1代表下一层的x和y

到这里还没结束；利用特解x0和y0可以求得通解，通解如下

p=x0 +b/g\*t;

q=y0 - a/g\*t;

p，q是通解

## 裴蜀定理 :

*ax*+*by*=*c*,成立的充要条件是c%gcd(a,b)==0所有数是正整数，可以知道给定正整数a和b,存在整数x和y使s=A\*x+b\*y最小s是gcd(a,b)

扩展到给定长度是n正整数序列A，让你找到一个整数序列X，使得s=sum(A[i]\*X[i])最小，答案是所有A[i]求最大公约数

## 问题模型：

1求解不定方程，线性不定方程ax+by=c整数解，如果c%gcd(a,b)==0才存在，否则无解

解是ax+by=gcd(a,b)的解分别/gcd(a,b)\*c

即x1/gcd\*c,y1/gcd\*c，但这不是全部ax+by=c整数解，仅仅是一部分

全部ax+by=c整数解是：

xt=x1+b/gcd\*t;

yt=y1-a/gcd\*t;

如果想调整结果可以：b/gcd可以调整x的结果，a/gcd可以调整y结果,

怎么调整，参考上面的yt xt的公式

对于ax+by=c中，a,b,c其中有负数的情况，应当成整数去求解，然后在结果中的x,y去取反

2 求解逆元根据逆元定义，ax%p=1,称x是a对p的逆元，且逆元存在条件a与p互质，所以可以转换为：求ax-py=1的解，利用求解线性方程组的方法去求x和y，x是逆元；

3  ax≡b (mod n)，求解同余方程，同余方程可以表示为ax+ny=b;

按照求解线性方程组的方法去求解，x是同余方程解

## 代码：

ll 是宏定义的long long

ll gcd(ll a,ll b){//普通求最大公约数

return b?gcd(b,a%b):a;

}

ll exgcd(ll a,ll b,ll &x,ll &y){ //返回最大公约数，扩展的欧几里得算法

if(!b){

x=1;y=0;return a;

}

ll e=exgcd(b,a%b,x,y);

ll kkz=x;x=y;y=kkz-a/b\*y;// x1= y2;y1= x2-a/b\*y2; x1和y1代表下一层的x和y

return e;

}

经典题目：

poj1061 poj2115