1.exgcd可以求解二元一次方程gcd(a,b)=ax+by的特殊解，

因为gcd(a,b)\*t可以为很多数，故可以推广到其他二元一次方程的特殊解，

若有方程ax+by=c,当c%gcd(a,b)==0时,(此为先决条件,不满足则无解),

设d=gcd(a,b), (可以在exgcd中当b==0时用d=a来记录一下gcd(a,b)),

先求出ax+by=gcd(a,b)的一组特解x0,y0, 然后令x0,y0同时乘上c/d，

就得到了ax+by=c的一组特解(c/d)x0,(c/d)y0,

通解x=c/d\*x0+k\*b/d,y=c/d\*y0-k\*a/d,(k为整数)

2.还有一种题，求a\*x=b(mod m),等价求a\*x-b=m\*(-y),

即为a\*x+m\*y=b,只需解不定方程a\*x+m\*y=b即可

线性同余方程有解当且仅当b%gcd(a,m)==0有解

3.还有些题目,要求三元(多元)一次方程, ax+by+cz=d

这种问题,可以先求e=gcd(a,b)然后就变成了ke+cz=d,那么可以用exgcd求出来解

然后就变成了ax+by=ke,再用exgcd,就可以求出来解了

ll exgcd(ll a,ll b,ll &x,ll &y)

{ll d;//d就是a,b的gcd

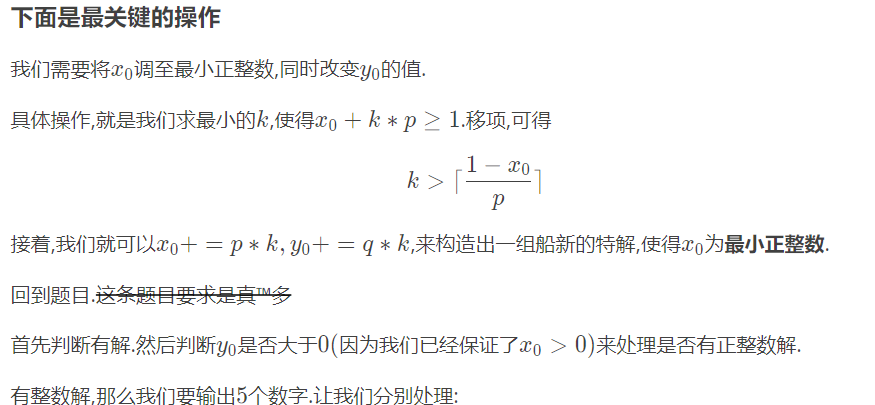
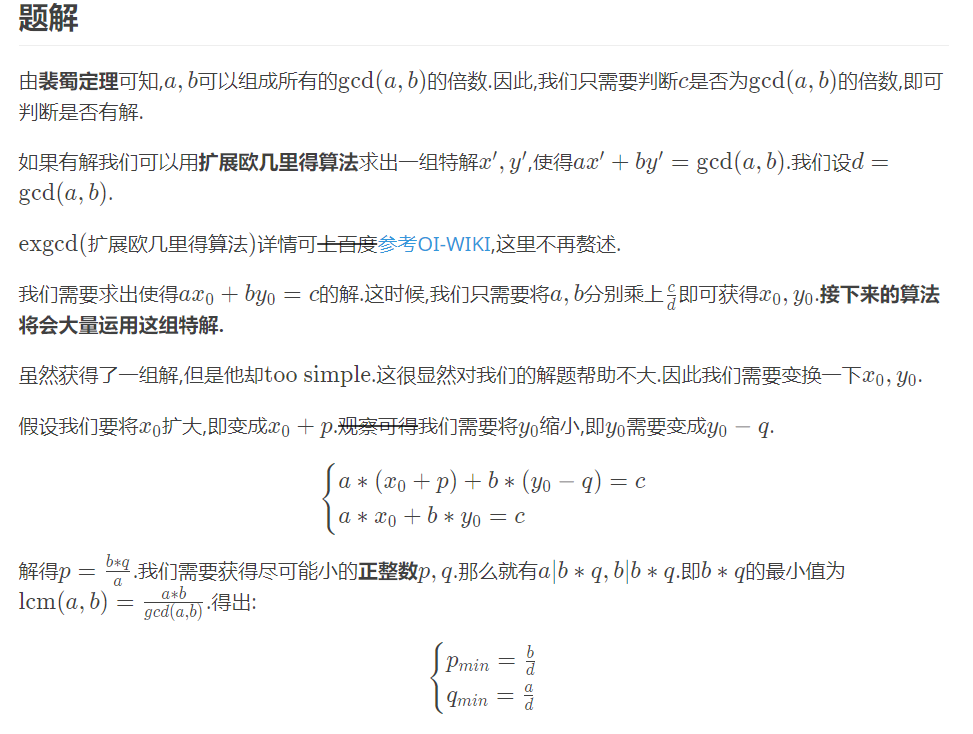
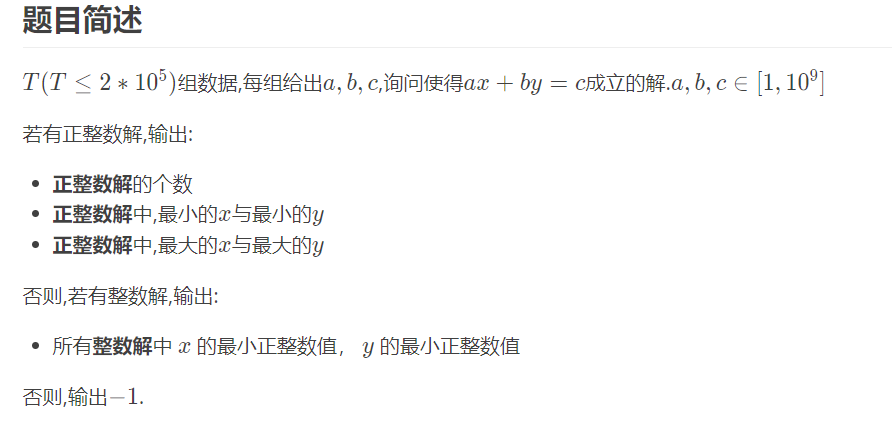
if(b==0){x=1;y=0;d=a;}

else {d=exgcd(b,a%b,y,x);y-=a/b\*x;}

return d;

}

模板题: <https://www.luogu.com.cn/problem/P5656>

ll exgcd(ll a,ll b,ll &x,ll &y)

{ll d;

if(b==0){x=1;y=0;d=a;}

else {d=exgcd(b,a%b,y,x);y-=a/b\*x;}

return d;

}

int main(){

dn(qread(),1,T){

ll a=qread(),b=qread(),c=qread(),x,y;

ll d=exgcd(a,b,x,y);

if(c%d!=0) puts("-1");

else{

x\*=c/d,y\*=c/d; ll p=b/d,q=a/d,k;

if(x<0) k=ceil((1.0-x)/p),x+=p\*k,y-=q\*k;//将x提高到最小正整数 ,1.0就是其想要提高到的最小数,如果要提高到0就变成0

else if(x>=0)k=(x-1)/p ,x-=p\*k,y+=q\*k; //将x降低到最小正整数 ,1可以改变,同上

if(y>0){ //有正整数解

printf("%lld ",(y-1)/q+1); //将y减到1的方案数即为解的个数

printf("%lld ",x); //当前的x即为最小正整数x

printf("%lld ",(y-1)%q+1); //将y取到最小正整数

printf("%lld ",x+(y-1)/q\*p); //将x提升到最大

printf("%lld ",y); //特解即为y最大值

} else{ //无正整数解

printf("%lld " ,x); //当前的x即为最小的正整数x

printf("%lld",y+q\*(ll)ceil((1.0-y)/q)); //将y提高到正整数

}

puts("");

}

}

return 0;

}

### 裴蜀定理内容

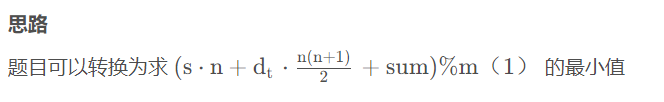
*ax*+*by*=*c*,*x*∈*Z*,*y*∈*Z*成立的充要条件是gcd(*a*,*b*)∣*c*。*Z*表示正整数集。

显然,这个也可以推广到n个数,所以,要使得s尽量的小并且>0,则s一定为所有ai的gcd,求出即可

然后具体的xi,可以通过我上面说的那种求多元一次方程的特殊解的exgcd的方法反推出来

下面就有一道类似的例题:

<https://codeforces.com/gym/104090/problem/A>



n,sum,m为已知值,s,dt和最小值都是需要求的

首先可以把%m转化为+x\*m,那么就变成了

s\*n+d\*n\*(n+1)/2+x\*m=min-sum

很显然右边满足裴蜀定理,则一定为gcd倍数,则min就为sum+k\*gcd>0的那个最小值,求出来之后再通过exgcd反推特殊解即可