（1）最大公约数——欧几里得算法

```
//写法一
int gcd(int a, int b)
{
    //b不为0，返回gcd(b, a % b)；b为0，返回a
    return b ? gcd(b, a % b) : a;
}

//写法二
int gcd(int a, int b)
{
    if(b == 0)    return a;
    return gcd(b, a % b);
}
```

（2）裴蜀定理——扩展欧几里得算法

1. 对于任意正整数a,b，一定存在非零整数x，y，使得 a * x + b * y = gcd(a, b)

```
// x和y 分别为 a和b的系数
int Exgcd(int a, int b, int &x, int &y) {
    if (b == 0){
        x = 1, y = 0;
        return a;
    }
    int d = Exgcd(b, a % b, y, x);
    y -= a / b * x;
    return d;
}
```

2. 线性同余方程

题目链接：https://www.acwing.com/problem/content/880/

> 模n同余：
>
> a * x = b (mod m) //称为a * x 与 b 模 m 同余
>
> 即(a * x) % m == b % m

> a * x = b (mod m) //称为a * x 与 b 模 m 同余
> 即(a * x) % m == b % m，我们可以进一步推出
> (a * x) - b == y * m，y是一个参数
> a * x - m * y == b; 变为扩展欧几里得算法的一般形式

1. 因为 $a * x \equiv b \pmod m$ 等价于 $a * x - b$ 是m的倍数，因此线性同余方程等价为 $a * x + m * y = b$
2. 根据 Bezout 定理，上述等式有解当且仅当 gcd(a,m) | b
3. 因此先用扩展欧几里得算法求出一组整数 x0,y0 使得 $a * x0 + m * y0 = gcd(a,m)$。然后 x = x0 * b / gcd(a,m) % m 即是所求。

（3）最小公倍数

```
int lcm(int a, int b)
{
    return a * b / gcd(a, b);
}
```