扩展欧几里得算法（$exgcd$）

贝祖定理

如果a, b是整数，那么一定存在整数x, y使得$ax + by = gcd(a, b)$
如果$ax + by = m$有解，那么m一定是$gcd(a, b)$的若干倍

我们在做题的过程中，有时会需要我们求出一对(x, y)满足$ax + by = m$，那么我们如何通过贝祖定理求出满足条件的(x, y)呢?

gcd

我们求gcd时，采用的是欧几里得算法，即辗转相除法，代码如下：

```
T gcd(T a, T b) {
    return b == 0 ? a : gcd(b, a % b);
}
```

通过代码可以看出，最终$b = 0$时，a的值即为我们求的最大公约数
`gcd` 函数，是利用了递归的方法求最大公约数。如果我们将目光集中到递归的最深一层，即$b = 0$时，其实我们可以获得一对(x, y)的解：$x = 1, y = 0$，满足：$a \cdot 1 + b \cdot 0 = gcd(a, b) = a$

注意，这组解只适用于递归最深层的a和b，当函数返回至递归的上一层时，a, b的值有所变化，x, y的值同样会有改变，所以，我们要找出递归时相邻两层的x, y的关系，从而在递归一层层返回之后，求出对于我们最开始的a, b，满足$ax + by = m$的解

$exgcd - 特解$

由前面对于 `gcd` 函数的讨论，我们至少可以获得 `exgcd` 函数的部分样貌：

```
T exgcd(T a, T b, T& x, T& y) {
    if (b == 0) {
        x = 1, y = 0;
        return a;
    }
    T d = exgcd(b, a % b, ...①);
    ......②
}
```

以上代码中用 `...` 代替的部分是后面需要进行讨论的
至少现在，我们完成了 `exgcd` 函数递归的返回条件
注意，由于x，y使用的是引用(&)，故每次修改后的x，y值在递归返回上一层后依然会保留
递归调用 `exgcd` 后，函数返回并继续执行②的代码时，已经求出了：

1. 最大公约数d，并且这个d是对于递归每一层的a, b都相等的
2. 满足$bx + (a\%b)y = d$的(x, y)的一组解
   这里的$b, (a\%b)$实际上分别是递归至更深一层时的a, b

接下来，我们就要根据已经求好的值，来计算对于本层递归的(a, b)，满足$ax + by = d$ 的(x, y)
我们有以下已知条件：

$$bx + (a\%b)y = d \tag{1}$$
$$a\%b = a - \frac{a}{b} \cdot b \tag{2}$$

将（2）带入（1）可得：

$$
\begin{aligned}
d &= bx + (a - \frac{a}{b} \cdot b)y \\
&= bx + ay - \frac{a}{b} \cdot by \\
&= ay + b(x - \frac{a}{b} \cdot y) = d
\end{aligned}
$$

如果我们将本层递归的(x, y)设为x_1, y_1，那么有：

$$
\begin{aligned}
x_1 &= y \\
y_1 &= x - \frac{a}{b} \cdot y
\end{aligned}
$$

这不仅求出了对于本层递归的a, b，满足条件的x, y，还求出了递归相邻两层之间x, y的关系，这样一层一层递归回去，不断更新x, y，就可以求出最终我们需要的那组解(x, y)了

等等，那为什么我前面展示 `exgcd` 时，还有一个①部分也用 `...` 替代了？不是直接将x, y传进去就行了吗？

按顺序传入x, y当然完全正确，但如果我们交换传入x, y的顺序，即：

```
T d = exgcd(b, a % b, y, x);
```

可以简化一些对于x, y的运算。
因为，交换以后，相当于x, y的值完全反了，那对于x_1, y_1的运算，要修改为：

$$
\begin{aligned}
x_1 &= x \\
y_1 &= y - \frac{a}{b} \cdot x
\end{aligned}
$$

我们发现，这时x的值，每层递归都会是一样的了，对于y，只需要：

```
y -= a / b * x;
```

即可，代码简化了一些。
于是有完整的 `exgcd` 函数代码：

```
using T = int;
T exgcd(T a, T b, T& x, T& y) {
    if (b == 0) {
        x = 1, y = 0;
        return a;
    }

    T gcd = exgcd(b, a % b, y, x);
    y -= a / b * x;
    return gcd;
}
```

$exgcd$ — 通解

在以上讨论中，其实我们求出的只是关于x, y的一组特解，因为是满足：$ax + by = gcd(a, b) = d$，这里的d一旦a和b确定就确定了。而在某些情况下，我们是需要求出满足$ax + by = m$的x, y的通解。

我们获得x, y特解后，假设x的通解为：$x' = x + \beta$，其中β是整数
那么可得：$x = x' - \beta$，代入$ax + by = gcd(a, b) = d$可得：

$$
\begin{aligned}
d &= a(x' - \beta) + by \\
&= ax' - a\beta + by \\
&= ax' + b(y - \frac{a\beta}{b}) = d
\end{aligned}
$$

可以得到：$y' = y - \frac{a\beta}{b}$

还没结束，因为我们要保证 $\frac{a\beta}{b}$ 是整数才符合条件

可以有以下转化：

$$\frac{a\beta}{b} = \frac{a}{gcd(a,b)} \cdot \frac{gcd(a,b)}{b} \cdot \beta \tag{3}$$

设$a' = \frac{a}{gcd(a,b)}$，$b' = \frac{b}{gcd(a,b)}$

可转化式（3）：

$$\frac{a\beta}{b} = \frac{a'}{b'} \cdot \beta$$

因为$a' = \frac{a}{gcd(a,b)}$，$b' = \frac{b}{gcd(a,b)}$，则 a', b'一定互质，这使得必须有$\beta = kb'(k \in Z)$才能满足$(\frac{a'}{b'} \cdot \beta)$为整数

带入x', y'即可得到答案：

$$x' = x + k \cdot \frac{b}{gcd(a,b)}$$
$$y' = y - k \cdot \frac{a}{gcd(a,b)}$$

其中，$k \in Z$