```
///逆元inverse
//定义：如果a * b = 1 (% MOD)，则b 是a的逆元（模逆元，  乘法逆元）
//a的逆元存在条件：gcd(a, MOD) == 1
//性质：逆元是积性函数，如果c = a * b，则 inv[c] = inv[a] * inv[b] % MOD
//方法一：  循环找解法（暴力）
//O(n)  预处理inv[1—n]：O（n^2）
LL getInv(LL x, LL MOD)
{
    for(LL i = 1; i < MOD; i++)
        if(x * i % MOD == 1)
            return i;
    return —1;
}

//方法二：费马小定理 a(p − 1) = 1(//O(logn)(), inv[1 −
    n]O nlogn)LLqpow(LLx, LLk, LLMOD)....LLgetInv(LLx, LLMOD)returnqpow(x, MOD − 2, MOD);////@ :ax
    + by = gcd(a, b)@@@@@@@@@@//a ∗ x//O(logn), inv[1 −
    n]O(nlogn)LLexgcd(LLa, LLb, LLx, LLy)if(b == 0)x = 1; y = 0; returna; LLg = exgcd(b, aLLt = x; x = y; y = t − a/b ∗ y; returng;
    −inv[n − 1], inv[n]MOD > n)//MOD = x ∗ n − y(0 <= y < n), ==> x ∗ n = y(//inv[n] = x ∗ inv[y](x =
    MOD − MOD/n, yMOD//O(logn)inv[1 −
    n]O(n)LLinv[NUM]; voidinv_pre(LLmod)inv[0] = inv[1] = 1LL; for(inti = 2; i < NUM; i + +)inv[i] = (mod − mod/i) ∗ inv[mod//
    //exgcdO(logn)//O(logn)//O(nlogn), O(logn)
```