数论和杂项

```
/* 取模 */
int P;
int add(int x, int y) {return x + y >= P ? x + y - P : x + y;}
int mul(int x, int y) {return (long long)x * y % P;}
int qp(int a, int b){
    int r = 1;
    for(; b; b >>= 1, a = mul(a, a)) if(b & 1) r = mul(r, a);
    return r;
}

/* 逆元 */
int inv[CN];
inv[1] = 1; for(int i = 2; i <= n; i++) inv[i] = (LL)(p - p / i) * inv[p % i] %
p;

/* 欧拉筛 */
int p[CN], md[CN]; bool np[CN]; // md[] 最小质因数
void sieve(int n){
    np[1] = 1, md[1] = 0;
    for(int i = 2; i <= n; i++){
        if(!np[i]) p[++p[0]] = i, md[i] = i;
        for(int j = 1; j <= p[0] && i * p[j] <= n; j++){
            int x = i * p[j]; np[x] = 1, md[x] = p[j];
            if(!(i % p[j])) break;
        }
    }
}

/* 高斯消元 */
#define DB double
bool equ(){
    int p = 1; // 最后一个确定主元的方程+1
    for(int i = 1; i <= n; i++){
        int q = -1;
        for(int j = p; j <= n; j++){
            if(fabs(a[j][i]) < EPS) continue;
            if(q == -1 || fabs(a[j][i]) > fabs(a[q][i])) q = j; // 精度优化
        }
        if(q == -1) continue; swap(a[p], a[q]), p++;
        for(int j = 1; j <= n; j++){ // 消元成对角矩阵
            if(fabs(a[j][i]) < EPS || j == p - 1) continue;
            DB t = a[j][i] / a[p - 1][i];
            for(int k = i; k <= n + 1; k++) a[j][k] -= t * a[p - 1][k];
        }
    }
    if(p <= n){ // 不满秩则不可能解出所有变量
        for(int i = p; i <= n; i++)
            if(fabs(a[i][n + 1]) > EPS) return puts("-1"), 0;
        return puts("0"), 0;
    }
    for(int i = 1; i <= n; i++) // 解对角矩阵
```

```
        printf("x%d=%.2lf\n", i, a[i][n + 1] / a[i][i]);
    return 1;
}

/* GCD 欧几里得算法 */
int gcd(int a, int b) {return b ? gcd(b, a % b) : a;}
void exgcd(int a, int &x, int b, int &y){
    if(!b) return (void)(x = 1, y = 0);
    exgcd(b, x, a % b, y); int t = x; x = y, y = t - (a / b) * y;
}
bool solve(int a, int &x, int b, int &y, int c){
    int g = gcd(a, b); if(c % g) return false;
    return exgcd(a, x, b, y), c /= g, x *= c, y *= c, true;
}

/* 卢卡斯 组合数取模 */
int qpow(int a,int b,int r){
    int rec = 1;
    while(b){
        if(b & 1) (rec *= a) %= r;
        (a *= a) %= r; b >>= 1;
    }
    return rec;
}
int cal(int n,int m,int p){
    if(m > n) return 0;
    if(m > n - m) m = n - m;
    int fm = 1,fn = 1;
    for(int i=2;i<=m;i++) (fm *= i) %= p;
    for(int i=n-m+1;i<=n;i++) (fn *= i) %= p;
    return (fn * qpow(fm, p - 2, p)) % p;
}
int C(int n,int m,int p){
    if(!m || m == n) return 1;
    return (C(n / p, m / p, p) * cal(n % p, m % p, p)) % p;
}

/* BSGS 离散对数 */
int mul(int x, int y, int P) {return (long long)x * y % P;}
int qp(int a, int b, int P){
    int r = 1;
    for(; b; a = mul(a, a, P), b >>= 1) if(b & 1) r = mul(r, a, P);
    return r;
}
int B; map<int, int> vis;
void bd(int a, int P){
    B = ceil(sqrt(P)), vis.clear(); int t = qp(a, B, P);
    for(int i = 1, p = t; i <= B; i++, p = mul(p, t, P))
        if(!vis.count(p)) vis[p] = i * B;
}
int qu(int a, int b, int P){
    int ans = 2e9;
    for(int i = 0, p = 1; i < B; i++, p = mul(p, a, P)){
        int cur = mul(b, p, P);
        if(vis.count(cur)) ans = min(ans, vis[cur] - i);
```

```
    }
    return ans < int(2e9) ? ans : -1;
}

/* Cipolla 二次剩余 */
class COMP {public: int x, y;}; int w;
COMP mk(int a, int b) {COMP o; o.x = a, o.y = b; return o;}
COMP mul(COMP a, COMP b, int p){
    COMP r;
    r.x = (1ll * a.x * b.x % p + 1ll * a.y * b.y % p * w % p) % p;
    r.y = (1ll * a.x * b.y % p + 1ll * a.y * b.x % p) % p;
    return r;
}
int qp(int a, int b, int p){
    int r = 1;
    for(; b; b >>= 1, a = 1ll * a * a % p) if(b & 1) r = 1ll * r * a % p;
    return r;
}
COMP qp(COMP a, int b, int p){
    COMP r = mk(1, 0);
    for(; b; b >>= 1, a = mul(a, a, p)) if(b & 1) r = mul(r, a, p);
    return r;
}
bool ck(int n, int p) {return qp(n, (p - 1) / 2, p) == 1;}
int sqrt(int n, int p){
    n %= p;
    if(p == 2) return n; if(!n) return 0;
    if(!ck(n, p)) return -1;
    int a = rand() % p;
    while(!a || ck((1ll * a * a % p - n + p) % p, p)) a = rand() % p;
    COMP x = mk(a, 1); w = (1ll * a * a % p - n + p) % p;
    return qp(x, (p + 1) / 2, p).x;
}
```

杂项

```
/* 并查集 DSU */
class DSU{
  public: int fa[CN];
    DSU() {for(int i = 0; i < CN; i++) fa[i] = i;}
    int fd(int x) {return fa[x] ^ x ? fa[x] = fd(fa[x]) : x;}
    bool mg(int x, int y) {return x = fd(x), y = fd(y), x ^ y ? fa[x] = y, 1 :
0;}
} ;

/* 离散化 */
int casy(int y) {return lower_bound(toty + 1, toty + ty + 1, y) - toty;}
sort(toty + 1, toty + ty + 1), ty = unique(toty + 1, toty + ty + 1) - toty - 1;
```