1. 模下运算

```
#include <bits/stdc++.h>
using namespace std;
typedef long long ll;

namespace MOLDULE
{
    const ll MOD = 1e9 + 7;
    inline ll Get_Mod(ll x) {return (x-x/MOD*MOD+MOD)%MOD;}
    ll qcpow(ll x, ll y) {
        ll res = 1; x = Get_Mod(x);
        for (; y; y >>= 1, x = x * x % MOD)
            if (y & 1) res = res * x % MOD;
        return res;}
    inline ll inv(ll x) {return qcpow(x,MOD-2);}
    inline ll add(ll x, ll y) {return Get_Mod(x + y);}
    inline ll addto(ll &x, ll y) {return x = add(x, y);}
    inline ll sub(ll x, ll y) {return Get_Mod(x - y);}
    inline ll subto(ll &x, ll y) {return x = sub(x, y);}
    inline ll mul(ll x, ll y) {return Get_Mod(1ll*x * y);}
    inline ll multo(ll &x, ll y) {return x = mul(x, y);}
    inline ll mdiv(ll x, ll y) {return Get_Mod(1ll*x*inv(y));}
    inline ll mdivto(ll &x, ll y) {return x = mdiv(x, y);}
}
```