```cpp
#include <bits/stdc++.h>
using namespace std;
#define MOD (int)(1e9+7)
#define ll long long
```

Useful define:

```cpp
#define pi 3.14159265358979323846
#define outf(x) printf("%.15f\n" , x)
#define outf(x) cout<<fixed<<setprecision(15)<<x<<endl
```

Calculate combination for small constraint:

```cpp
ll ncr(int n,int r)
{
    ll fac1 = 1 , fac2 = 1 , fac;
    for (int i = r; i >= 1; i-- , n--)
    {
        fac1 = fac1 * n;
        if(fac1%i==0)
            fac1 = fac1 / i;
        else
            fac2 = fac2 * i;
    }
    fac = fac1 / fac2;
    return fac % MOD;
}
```

Fast power and CNR:

```cpp
const int N = 300500;
const int mod = 1000000007;
ll fact[N] , invFact[N];

ll fast_pow(ll a, ll p) {
    ll res = 1;
    while (p) {
        if (p % 2 == 0) {
            a = (a * a) % mod;
            p /= 2;
        } else {
            res = (res * a) % mod;
            p--;
        }
    }
    return res;
}

ll C(int n, int k) {
    if(n==k) return 1;
    if (k > n) {
        return 0;
    }
    return (fact[n] * invFact[k] % mod * invFact[n - k] % mod)%mod;
}

//call it first in main
void init(){
    fact[0] = 1;
    for (int i = 1; i < N; i++) {
        fact[i] = (fact[i - 1] * i) % mod;
        invFact[i] = fast_pow(fact[i], mod - 2);
    }
}
```