乘法逆元

乘法逆元一般用来求形同下式的值，一般来说，p是一个质数

$$\frac{a}{b}\ (mod\ p)$$

逆元定义

$$如果a*x\equiv 1\ (mod\ b)，且a与b互质，那么称x为a的逆元，记作a^{-1}$$

$$\frac{a}{b}\ (mod\ p) = \frac{a}{b}*1\ (mod\ p) = \frac{a}{b}*bb^{-1}\ (mod\ p) = ab^{-1}\ (mod\ p)$$
$$bb^{-1}\equiv 1\ (mod\ p)$$

用快速幂求解逆元

欧拉定理

$$对于任何正整数a和n有gcd(a,n)=1，则a^{\phi(n)}\equiv 1\ (mod\ n)$$

证明：

$$假设A=a_1,a_2,...,a_{\phi(n)}是1\ n中\phi(n)个互不相同的且与n互质的数$$
$$且0<a_1<a_2<...<a_{\phi(n)}<n$$
$$构造一组新的数Z=a*a_1,a*a_2,...,a*a_{\phi(n)}$$
$$首先，z_i之间两两模n不同余$$
$$如果\exists z_i\equiv z_j(mod\ n),i\neq j$$
$$z_i-z_j\equiv 0(mod\ n) \implies a*(a_i-a_j)\equiv 0(mod\ n) \implies a_i-a_j\equiv 0(mod\ n) \implies a_i\equiv a_j(mod\ n)$$
$$矛盾$$
$$其次，gcd(z_i,n)=1,i=1,...,\phi(n)$$
$$则A=Z，即z_i(i=1..\phi(n))是a_i(i=1...\phi(n))的某个排列$$
$$\prod a_i\equiv\prod z_i(mod\ n) \implies (a^{\phi(n)}-1)a_1*...*a_{\phi(n)}\equiv 0(mod\ n)$$
$$a^{\phi(n)}\equiv 1(mod\ n)$$

费马小定理

$$若p是质数，a是正整数，且a,p互质。那么，a^{p-1}\equiv 1\ (mod\ p)$$

根据费马小定理，可以很容易的知道当p是质数，a是正整数时，$a^{-1}=a^{p-2}$

利用乘法逆元求组合数

原题链接 Acwing 886 求组合数II

```
#include <iostream>

using namespace std;

```

```
typedef long long LL;

const int N = 100010, P = 1e9 + 7;
int fact[N], infact[N];
int n;

int qmi(int a, int p, int mod)
{
    int res = 1;
    while (p)
    {
        if (p & 1) res = (LL) res * a % mod;
        a = (LL) a * a % mod;
        p >>= 1;
    }
    return res;
}

int main()
{
    fact[0] = infact[0] = 1;
    for (int i = 1; i < N; i ++)
    {
        fact[i] = (LL) fact[i - 1] * i % P;
        infact[i] = (LL) infact[i - 1] % P * qmi(i, P - 2, P) % P;
    }
    cin >> n;
    for (int i = 0; i < n; i ++)
    {
        int a, b;
        cin >> a >> b;
        cout << (LL) fact[a] % P * infact[b] % P * infact[a - b] % P << endl;
    }
    return 0;
}
```