

组合计数

暴力打表

a, b ≤ 2000

```
#include <algorithm>
#include <iostream>

using namespace std;

const int N = 2100, mod = 1e9 + 7;

int c[N][N];

void init(){
    for(int i = 0; i < N; i++){
        for(int j = 0; j <= i; j++){
            if(!j) c[i][j] = 1;
            else c[i][j] = (c[i - 1][j] + c[i - 1][j - 1]) % mod;
        }
    }
}

int main(){
    init(); //预处理组合数
    int t;
    cin >> t;
    while(t--){
        int a, b;
        cin >> a >> b;
        cout << c[a][b] << endl; //查表输出
    }

    return 0;
}
```

预处理阶乘&阶乘逆元

a, b ≤ 1e5 + 10, 预处理阶乘和阶乘的逆元

```
#include <algorithm>
#include <cstring>
#include <cstdio>
#include <iostream>

using namespace std;

typedef long long ll;

const int N = 1e5 + 100, mod = 1e9 + 7;
```

```

int n ;
int a,b;
int fact[N],infact[N] ;           //fact存阶乘 , infact存阶乘的逆元

int qmi(int a,int n,int mod){ //快速幂 求得  $a^n \% mod$ 
    int res = 1 ;
    while(n){
        if(n & 1) res = (ll)res * a % mod ;
        a = (ll)a * a % mod ;
        n >>= 1 ;
    }
    return res ;
}

int main(){
    fact[0] = infact[0] = 1;
    for(int i = 1 ; i < N ; i++){
        fact[i] = (ll)fact[i-1] * i % mod;
        infact[i] = (ll)infact[i-1] * qmi(i,mod - 2,mod) % mod;
    }

    scanf("%d",&n) ;

    while(n--){
        scanf("%d%d",&a,&b) ;
        printf("%d\n",(ll)fact[a] * infact[b] % mod * infact[a - b] % mod );
    }
    return 0;
}

```

Lucas定理

$a, b \leq 1e18$

```

#include <algorithm>
#include <cstring>
#include <cstdio>
#include <iostream>

using namespace std;

typedef long long ll ;

const int N = 1e5 + 100 ;

ll n ;
ll a,b,p ;

ll qmi(ll a ,ll n){ //快速幂
    ll res = 1;
    while(n){
        if(n & 1) res = res * a % p ;
        a = a * a % p ;
    }
}

```

```

        n >= 1 ;
    }
    return res ;
}

11 c(11 a,11 b){ //使用组合数定义，结合逆元（因为p是素数）求得组合数
    11 res = 1;
    for(int i = a ,j = 1; j <= b ; i-- , j++){
        res = res * i % p ;
        res = res * qmi(j,p-2) % p ;
    }
    return res;
}

11 lucas(11 a,11 b){ // lucas定理递归处理
    if(a < p && b < p) return c(a,b) ;
    return lucas(a/p,b/p) * c(a%p,b%p) % p ;
}

int main(){
    cin >> n ;
    while(n--){
        cin >> a >> b >> p ;
        cout << lucas(a,b) << endl ;
    }
    return 0;
}

```