#include<bits/stdc++.h>

using namespace std;

#define ll long long

#define fastio ios\_base::sync\_with\_stdio(false);cin.tie(NULL);cout.tie(NULL);

#define endl '\n'

#define test ll t; cin >> t; while(t--)

#define vec(ver, n)  vector<ll> ver(n); for (ll i = 0; i < n; i++) cin >> ver[i];

#define pb push\_back

#define all(x) (x).begin(),(x).end()

bool pow2(ll n)

{

    if(n==0) return false;

    return (ceil(log2(n)) == floor(log2(n)));

}

const int32\_t N=1e5+10;

const long long INF=1e18 + 2;

const long long \_INF=-1e18;

const int32\_t mod=1000000007;

const int32\_t MM=998244353;

ll bin\_expo(ll x, ll y) { ll res = 1; while (y) {if (y % 2) res = (res \* x % mod) % mod; x = (x \* x) % mod; y /= 2; } return res;}

ll mod\_inv(ll x) {return bin\_expo(x, mod - 2);}

ll mod\_add(ll a, ll b) {a = a % mod; b = b % mod; return (((a + b) % mod) + mod) % mod;}

ll mod\_mul(ll a, ll b) {a = a % mod; b = b % mod; return (((a \* b) % mod) + mod) % mod;}

ll mod\_sub(ll a, ll b) {a = a % mod; b = b % mod; return (((a - b) % mod) + mod) % mod;}

ll mod\_div(ll a, ll b) {a = a % mod; b = b % mod; return (mod\_mul(a, mod\_inv(b)) + mod) % mod;}

// ll fact[N]={}; rep(i,1,N) fact[i] = mod\_mul(fact[i-1],i,mod);

// ll nCr(ll n,ll r) return mod\_mul(mod\_mul(fact[n],mod\_inv(fact[r],mod),mod),mod\_inv(fact[n-r],mod),mod);

signed main(){

    fastio

    test{

    }

    return 0;

}