Calculate the value of (x^n)%m :

```cpp
int modPow(int x, int n, int m) {
    if (n == 0) return 1 % m;
    ll u = modPow(x , n/2 , m);
    u = (u * u) % m;
    if (n%2 == 1) u = (u * x) % m;
    return u;
}
```

Operation functions with mod:

```cpp
int mul(int x , int y , int m)
{
    return (ll) x*y % m;
}
int sum(int x , int y , int m)
{
    return (x+y) % m;
}
int sub(int x , int y , int m)
{
    return sum((x-y)%m , m , m);
}
int po(int x , int y , int m)
{
    if(!y) return 1;
    if(y&1) return mul(x , po(x,y-1,m) , m);
    int z = po(x , y/2 , m);
    return mul(z , z , m);
}
int inv(int x , int m)
{
    return po(x , m-2 , m);
}
// note: 3/5 % MOD ⟹ 3*inv(5 , MOD)
```

Operation function with mod:

```cpp
//this function faster then the other
void add_self(int& x, int y)
{
    if((x += y) ⩾ MOD) x -= MOD;
}
int add(int x, int y)
{
    return add_self(x, y), x;
}
void sub_self(int& x, int y)
{
    if((x -= y) < 0)    x += MOD;
}
int sub(int x, int y)
{
    return sub_self(x, y), x;
}
int mul(int x, int y)
{
    return (long long) x * y % MOD;
}
```