

```
1 #include<bits/stdc++.h>
2 using namespace std;
3 #define ll long long
4 #define mx 100005
5 ll MOD = 1000000007;
6 ll a[mx],lazy[4*mx],tree[4*mx],sum[4*mx];
7 void setValue(ll nd,ll b,ll e,ll p){
8     ll uu = ((e-b+1)*p)%MOD;
9     tree[nd] = (tree[nd]+uu)%MOD;
10    ll ee = (e*(e+1LL))/2LL;
11    ll bb = ((b-1LL)*b)/2LL;
12    ll xx = (ee-bb+MOD)%MOD;
13    ll vv = (xx*p)%MOD;
14    sum[nd] = (sum[nd]+vv)%MOD;
15 }
16 void pushDown(ll nd,ll b,ll e){
17     ll m = (b+e)/2;
18     setValue(2*nd,b,m,lazy[nd]);
19     setValue(2*nd+1,m+1,e,lazy[nd]);
20
21     lazy[2*nd] += lazy[nd];
22     lazy[2*nd+1] += lazy[nd];
23     lazy[2*nd] %= MOD;
24     lazy[2*nd+1] %= MOD;
25     lazy[nd]=0;
26 }
27 void init(ll nd,ll b,ll e){
28     if(b==e){
29         tree[nd] = a[b]%MOD;
30         sum[nd] = (b*a[b])%MOD;
31         lazy[nd] = 0;
32         return;
33     }
34
35     ll m = (b+e)/2;
36     init(2*nd,b,m);
37     init(2*nd+1,m+1,e);
38     tree[nd] = (tree[2*nd]+tree[2*nd+1])%MOD;
39     sum[nd] = (sum[2*nd]+sum[2*nd+1])%MOD;
40     lazy[nd]=0;
41 }
42 void update(ll nd,ll b,ll e,ll l,ll r,ll p){
43     if(b>r||e<l) return;
44     if(b>=l&&e<=r){
45         setValue(nd,b,e,p);
46         lazy[nd] = (lazy[nd]+p)%MOD;
47         return;
48     }
49
50     if(lazy[nd]!=0){
51         pushDown(nd,b,e);
52     }
53
54     ll m = (b+e)/2;
55     update(2*nd,b,m,l,r,p);
56     update(2*nd+1,m+1,e,l,r,p);
57
58     tree[nd] = (tree[2*nd]+tree[2*nd+1])%MOD;
59     sum[nd] = (sum[2*nd]+sum[2*nd+1])%MOD;
60 }
```

```
11 query1(ll nd,ll b,ll e,ll l,ll r){
12     if(l>r) return 0;
13     if(b>r||e<l) return 0;
14     if(b>=l&&e<=r){
15         return (tree[nd])%MOD;
16     }
17
18     if(lazy[nd]!=0){
19         pushDown(nd,b,e);
20     }
21
22     ll m = (b+e)/2;
23     ll u1 = query1(2*nd,b,m,l,r)%MOD;
24     ll u2 = query1(2*nd+1,m+1,e,l,r)%MOD;
25     return (u1+u2)%MOD;
26 }
27 ll query2(ll nd,ll b,ll e,ll l,ll r){
28     if(l>r) return 0;
29     if(b>r||e<l) return 0;
30     if(b>=l&&e<=r){
31         return (sum[nd])%MOD;
32     }
33
34     if(lazy[nd]!=0){
35         pushDown(nd,b,e);
36     }
37
38     ll m = (b+e)/2;
39     ll v1 = query2(2*nd,b,m,l,r)%MOD;
40     ll v2 = query2(2*nd+1,m+1,e,l,r)%MOD;
41     return (v1+v2)%MOD;
42 }
43 int main(){
44     ios::sync_with_stdio(false); cin.tie(0); cout.tie(0);
45
46     ll tt; cin>>tt;
47     for(int ks=1; ks<=tt; ks++){
48         ll n,q; cin>>n>>q;
49         for(int i=1; i<=n; i++)cin>>a[i];
50
51         init(1,1,n);
52
53         cout<<"Case "<<ks<< ":"<<endl;
54         while(q--){
55             ll c,l,r,x; cin>>c>>l>>r>>x;
56             if(c==1){
57                 update(1,1,n,l,r,x);
58             }
59             else{
60                 ll u = query1(1,1,n,l,r)%MOD;
61                 ll w = query1(1,1,n,l+1,r)%MOD;
62                 ll v = query2(1,1,n,l+1,r)%MOD;
63                 w = (l*w)%MOD;
64                 ll vw = (v-w+MOD)%MOD;
65                 vw = (vw*x)%MOD;
66                 ll ans = (u+vw)%MOD;
67                 cout<<ans<<endl;
68             }
69         }
70     }
71     return 0;
72 }
```