

```

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 }

```

```

61 ll query1(ll nd,ll b,ll e,ll l,ll r){
62     if(l>r)return 0;
63     if(b>r||e<l)return 0;
64     if(b>=l&&e<=r){
65         return (tree[nd])%MOD;
66     }
67
68     if(lazy[nd]!=0){
69         pushDown(nd,b,e);
70     }
71
72     ll m = (b+e)/2;
73     ll u1 = query1(2*nd,b,m,l,r)%MOD;
74     ll u2 = query1(2*nd+1,m+1,e,l,r)%MOD;
75     return (u1+u2)%MOD;
76 }
77 ll query2(ll nd,ll b,ll e,ll l,ll r){
78     if(l>r)return 0;
79     if(b>r||e<l)return 0;
80     if(b>=l&&e<=r){
81         return (sum[nd])%MOD;
82     }
83
84     if(lazy[nd]!=0){
85         pushDown(nd,b,e);
86     }
87
88     ll m = (b+e)/2;
89     ll v1 = query2(2*nd,b,m,l,r)%MOD;
90     ll v2 = query2(2*nd+1,m+1,e,l,r)%MOD;
91     return (v1+v2)%MOD;
92 }
93 int main(){
94     ios::sync_with_stdio(false); cin.tie(0); cout.tie(0);
95
96     ll tt; cin>>tt;
97     for(int ks=1; ks<=tt; ks++){
98         ll n,q; cin>>n>>q;
99         for(int i=1; i<=n; i++)cin>>a[i];
100
101         init(1,1,n);
102
103         cout<<"Case "<<ks<<":"<<endl;
104         while(q--){
105             ll c,l,r,x; cin>>c>>l>>r>>x;
106             if(c==1){
107                 update(1,1,n,l,r,x);
108             }
109             else{
110                 ll u = query1(1,1,n,l,r)%MOD;
111                 ll w = query1(1,1,n,l+1,r)%MOD;
112                 ll v = query2(1,1,n,l+1,r)%MOD;
113                 w = (l*w)%MOD;
114                 ll vw = (v-w+MOD)%MOD;
115                 vw = (vw*x)%MOD;
116                 ll ans = (u+vw)%MOD;
117                 cout<<ans<<endl;
118             }
119         }
120     }
121     return 0;
122 }

```