```cpp
/// https://www.spoj.com/problems/SEGSQRSS/
#include<bits/stdc++.h>
using namespace std;
#define ll long long
#define mx 100005
#define sqr(x)  (x)*(x)
struct data{
    ll sum,sqrsum,lazy,upd;
};
data tree[3*mx];
ll ara[mx];

void push_down(int n, int b, int e){
    int l=n*2,r=l+1,mid=(b+e)/2;

    if(tree[n].upd){
        tree[l].lazy=0;
        tree[r].lazy=0;
        tree[l].sum=(mid-b+1)*tree[n].upd;
        tree[l].sqrsum=(mid-b+1)*sqr(tree[n].upd);
        tree[r].sum=(e-mid)*tree[n].upd;
        tree[r].sqrsum=(e-mid)*sqr(tree[n].upd);
        tree[l].upd=tree[n].upd;
        tree[r].upd=tree[n].upd;
        tree[n].upd=0;
    }
    if(tree[n].lazy){
        tree[l].sqrsum+=(tree[l].sum*(2*tree[n].lazy))+(mid-b+1)*sqr(tree[n].lazy);
        tree[r].sqrsum+=(tree[r].sum*(2*tree[n].lazy))+(e-mid)*sqr(tree[n].lazy);
        tree[l].sum+=(mid-b+1)*tree[n].lazy;
        tree[r].sum+=(e-mid)*tree[n].lazy;
        tree[l].lazy+=tree[n].lazy;
        tree[r].lazy+=tree[n].lazy;
        tree[n].lazy=0;
    }
}

void init(int n, int b, int e){
    if(b==e){
        tree[n].sum=ara[b];
        tree[n].sqrsum=sqr(ara[b]);
        tree[n].lazy=0;
        tree[n].upd=0;
        return;
    }

    int l=n*2,r=l+1,mid=(b+e)/2;

    init(l,b,mid);
    init(r,mid+1,e);

    tree[n].sum=tree[l].sum+tree[r].sum;
    tree[n].sqrsum=tree[l].sqrsum+tree[r].sqrsum;
    tree[n].lazy=0;
    tree[n].upd=0;
}

void update(int n, int b, int e, int x, int y, ll val, int type){
    if(b>y || e<x) return;
    if(b>=x && e<=y){
        if(type==0){
            tree[n].sum=(e-b+1)*val;
            tree[n].sqrsum=(e-b+1)*sqr(val);
            tree[n].lazy=0;
            tree[n].upd=val;
        }else{
            tree[n].sqrsum+=(tree[n].sum*2*val)+((e-b+1)*sqr(val));
            tree[n].sum+=(e-b+1)*val;
            tree[n].lazy+=val;
        }
        return;
    }
```

```
73
74      push_down(n,b,e);
75
76      int l=n*2,r=l+1,mid=(b+e)/2;
77
78      update(l,b,mid,x,y,val,type);
79      update(r,mid+1,e,x,y,val,type);
80
81      tree[n].sum=tree[l].sum+tree[r].sum;
82      tree[n].sqrsum=tree[l].sqrsum+tree[r].sqrsum;
83  }
84
85  ll query(int n, int b, int e, int x, int y){
86      if(b>y || e<x) return 0;
87      if(b>=x && e<=y) return tree[n].sqrsum;
88
89      push_down(n,b,e);
90
91      int l=n*2,r=l+1,mid=(b+e)/2;
92
93      ll p=query(l,b,mid,x,y);
94      ll q=query(r,mid+1,e,x,y);
95      return p+q;
96  }
97  int main(){
98      int tt;scanf("%d",&tt);
99
100     for(int ks=1; ks<=tt; ks++){
101
102         int n,q; scanf("%d%d",&n,&q);
103         for(int i=1; i<=n; i++) scanf("%lld",&ara[i]);
104
105         init(1,1,n);
106
107         printf("Case %d:\n",ks);
108         while(q--){
109             int typ,x,y; scanf("%d%d%d",&typ,&x,&y);
110
111             if(typ==2){
112                 ll ans=query(1,1,n,x,y);
113                 printf("%lld\n",ans);
114             }else{
115                 int v; scanf("%d",&v);
116
117                 if(typ==0) update(1,1,n,x,y,v,0);
118                 else update(1,1,n,x,y,v,1);
119             }
120         }
121     }
122
123     return 0;
124 }
```