

```
1: #include<iostream>
2: #include<vector>
3:
4: #define ll long long int
5: #define rep(n) for(int i=0;i<n;i++)
6:
7: //(1)
8: class LazySegTree
9: {
10: public:
11:     //(2)
12:     std::vector<ll> node;
13:     std::vector<ll> lazy;
14:     ll n_;
15:     LazySegTree(ll n)
16:     {
17:         n_=1;
18:         while(n_<n)
19:         {
20:             n_*=2;
21:         }
22:         node.resize(2*n_-1);
23:         rep(2*n_-1)
24:         {
25:             node[i]=0;
26:         }
27:         lazy.resize(2*n_-1);
28:         rep(2*n_-1)
29:         {
30:             lazy[i]=0;
31:         }
32:     }
33:     //(3)
34:     void lazy_eval(ll i,ll l,ll r)
35:     {
36:         if(lazy[i]!=0)
37:         {
38:             node[i]+=lazy[i];
39:             if(r-l>1)
40:             {
41:                 lazy[i*2+1]+=lazy[i]/2;
42:                 lazy[i*2+2]+=lazy[i]/2;
43:             }
44:             lazy[i]=0;
45:         }
46:     }
47:     //(4)
48:     void Add(ll s,ll t,ll x,ll i,ll l,ll r)
49:     {
50:         lazy_eval(i,l,r);
51:         if(r<=s || t<=l)
52:         {
53:             return;
54:         }
55:         if(s<=l && r<=t)
56:         {
57:             lazy[i]=x*(r-l);
58:             lazy_eval(i,l,r);
59:         }
60:         else
61:         {
62:             ll mid=(l+r)/2;
63:             Add(s,t,x,i*2+1,l,mid);
64:             Add(s,t,x,i*2+2,mid,r);
65:             node[i]=node[i*2+1]+node[i*2+2];
66:         }
67:     }
68:
69:     //(5)
70:     ll getSum(ll s,ll t,ll i,ll l,ll r)
```

```
71:     {
72:         if(r<=s || t<=l)
73:         {
74:             return 0;
75:         }
76:         lazy_eval(i,l,r);
77:         if(s<=l && r<=t)
78:         {
79:             return node[i];
80:         }
81:
82:         ll mid=(l+r)/2;
83:         ll leaf_l=getSum(s,t,i*2+1,l,mid);
84:         ll leaf_r=getSum(s,t,i*2+2,mid,r);
85:         return leaf_l+leaf_r;
86:     }
87: };
88: int main()
89: {
90:     ll n,m;
91:     std::cin>>n>>m;
92:     LazySegTree *lst=new LazySegTree(n);
93:     //(6)
94:     rep(n)
95:     {
96:         ll a;
97:         std::cin>>a;
98:         lst->Add(i,i+1,a,0,0,lst->n_);
99:     }
100:     rep(m)
101:     {
102:         ll l,r,x;
103:         std::cin>>l>>r>>x;
104:         l--;
105:         r--;
106:         //(7)
107:         std::cout<<lst->getSum(l,r+1,0,0,lst->n_)<<std::endl;
108:         //(8)
109:         lst->Add(l,r+1,x,0,0,lst->n_);
110:     }
111:     return 0;
112: }
```