

```

1  #include <bits/stdc++.h>
2  using namespace std;
3
4  typedef long long int ll;
5  typedef pair<ll,ll> ii;
6  typedef vector<ii> vii;
7  typedef vector<ll> vi;
8
9  vi st,lazy;
10 int n;
11
12 ll query(int p, int L, int R, int i,int j){
13
14     if(lazy[p]!=0){
15         st[p] += (R-L+1)*lazy[p];
16         if(R!=L){
17             lazy[p<<1] += lazy[p];
18             lazy[(p<<1)+1] += lazy[p];
19         }
20         lazy[p] = 0;
21     }
22     // no overlap
23     if(i>R || j<L) return 0;
24
25     // total overlap
26     if(L>=i && R<=j) return st[p];
27
28     // partial overlap
29     int nxt = p << 1;
30     int mid = (L + R) >> 1;
31
32     return query(nxt,L,mid,i,j) + query(nxt + 1,mid +1,R,i,j);
33 }
34 void update(int P,int L,int R, int i,int j, ll value){
35
36     if(lazy[P]!=0){
37         st[P] += (R-L+1)*lazy[P];
38         if(L!=R){
39             lazy[P << 1] += lazy[P];
40             lazy[(P << 1)+1] += lazy[P];
41         }
42         lazy[P] = 0;
43     }
44
45     // no overlap
46     if( L > j or R < i) return;
47
48     // total overlap
49     if(L >= i and R <= j){
50         st[P] += (R-L+1)*value;
51         if(L!=R){
52             lazy[P<<1] += value;
53             lazy[(P<<1)+1] += value;
54         }
55
56         return;
57     }
58
59     // partial overlap
60     int nxt = P << 1;
61     int mid = (L+R) >> 1;
62
63     update(nxt, L, mid, i, j, value);
64     update(nxt+1,mid+1,R,i,j,value);
65
66     st[P] = st[nxt]+st[nxt+1];
67
68 }
69
70 main(){

```

```
71     int i,j,q,z,a,b,o;
72     ll v;
73
74     cin >> z;
75
76     for(i=0;i<z;i++){
77         cin >> n >> q;
78         st.resize(n << 2);
79         st.assign(n << 2,0);
80         lazy.resize(n << 2);
81         lazy.assign(n << 2,0);
82         for(j=0;j<q;j++){
83             cin >> o;
84             if(o==1){
85                 cin >> a >> b;
86                 cout << query(1,0,n-1,a-1,b-1) << endl;
87             }
88             else{
89                 cin >> a >> b >> v;
90                 update(1,0,n-1,a-1,b-1,v);
91             }
92         }
93     }
94 }
95
```