```cpp
#include <bits/stdc++.h>
using namespace std;

typedef long long int ll;
typedef pair<ll,ll> ii;
typedef vector<ii> vii;
typedef vector<ll> vi;

vi st,lazy;
int n;

ll query(int p, int L, int R, int i,int j){

    if(lazy[p]!=0){
        st[p] += (R-L+1)*lazy[p];
        if(R!=L){
            lazy[p<<1] += lazy[p];
            lazy[(p<<1)+1] += lazy[p];
        }
        lazy[p] = 0;
    }
    // no overlap
    if(i>R || j<L) return 0;

    // total overlap
    if(L>=i && R<=j) return st[p];

    // partial overlap
    int nxt = p << 1;
    int mid = (L + R) >> 1;

    return query(nxt,L,mid,i,j) + query(nxt + 1,mid +1,R,i,j);
}
void update(int P,int L,int R, int i,int j, ll value){

    if(lazy[P]!=0){
        st[P] += (R-L+1)*lazy[P];
        if(L!=R){
            lazy[P << 1] += lazy[P];
            lazy[(P << 1)+1] += lazy[P];
        }
        lazy[P] = 0;
    }

    // no overlap
    if( L > j or R < i) return;

    // total overlap
    if(L >= i and R <= j){
        st[P] += (R-L+1)*value;
        if(L!=R){
            lazy[P<<1] += value;
            lazy[(P<<1)+1] += value;
        }

        return;
    }

    // partial overlap
    int nxt = P << 1;
    int mid = (L+R) >> 1;

    update(nxt, L, mid, i, j, value);
    update(nxt+1,mid+1,R,i,j,value);

    st[P] = st[nxt]+st[nxt+1];

}

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
```