

[back](#)

```
#include<bits/stdc++.h>
using namespace std;
const int N=100005;
typedef long long ll;
ll tr[N*4],tag[N*4],a[N];
inline int ls(int o){
    return o<<1;
}
inline int rs(int o){
    return o<<1|1;
}
void push_up(int o){
    tr[o]=tr[ls(o)]+tr[rs(o)];
}
void build(int o,int l,int r){
    if(l==r){
        tr[o]=a[l];
        return;
    }
    int mid=l+r>>1;
    build(ls(o),l,mid);
    build(rs(o),mid+1,r);
    push_up(o);
}
void f(int o,int l,int r,ll k){
    tag[o]+=k;
    tr[o]+=(r-l+1)*k;
}
void push_down(int o,int l,int r){
    int mid=l+r>>1;
    f(ls(o),l,mid,tag[o]);
    f(rs(o),mid+1,r,tag[o]);
    tag[o]=0;
}
void xg(int o,int nl,int nr,int l,int r,ll k){
    if(nl<=l&&r<=nr){
        tag[o]+=k;
        tr[o]+=(r-l+1)*k;
        return;
    }
    push_down(o,l,r);
    int mid=l+r>>1;
    if(nl<=mid) xg(ls(o),nl,nr,l,mid,k);
    if(nr>mid) xg(rs(o),nl,nr,mid+1,r,k);
    push_up(o);
}
ll cx(int o,int nl,int nr,int l,int r){
    if(nl<=l&&r<=nr) return tr[o];
```

```
    push_down(o,l,r);
    int mid=l+r>>1;
    ll ret=0;
    if(nl<=mid) ret+=cx(ls(o),nl,nr,l,mid);
    if(nr>mid) ret+=cx(rs(o),nl,nr,mid+1,r);
    return ret;
}
int main(){
    int n,m;
    scanf("%d %d",&n,&m);
    for(int i=1;i<=n;i++){
        scanf("%d",&a[i]);
    }
    build(1,1,n);
    for(int i=1;i<=m;i++){
        int op;
        scanf("%d",&op);
        if(op==1){
            int x,y;ll k;
            scanf("%d %d %lld",&x,&y,&k);
            xg(1,x,y,1,n,k);
        }
        else{
            int x,y;
            scanf("%d %d",&x,&y);
            printf("%lld\n",cx(1,x,y,1,n));
        }
    }
    return 0;
}
```

From:

<https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link:

https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal_string:code2



Last update: **2020/05/07 11:12**