

题意

区间修改（同时加上一个值），区间查询线段和

分析

我对打lazy标记的理解：

如果某一个线段，完全包含于给定的区间，这就是最后一层，下面不必要再分，在一层打上标记，标志着这是我最后一次下分的地方，如果下次我要在这个地方继续下分，我就必须先把这个标记向下推，然后再访问，推标记的同时也要改变下面的结点值

在这里，同时加上一个值的lazy标记，无先后区分，如果一个结点先加了x,然后加了y，可以看作直接加了x+y,所以lazy标记在下推的过程中可以直接累加

代码

```
1  #include<iostream>
2  #include<algorithm>
3  #include<cstring>
4  #define For(i,a,b) for(int i=(a); i<=(b) ; i++)
5  #define _For(i,a,b) for(int i=(a); i>=(b) ; i--)
6  #define Memset(a,b); memset((a),(b),sizeof((a)));
7  #define Cin(a); scanf("%d",&(a));
8  #define Cinc(a); scanf(" %c",&(a));
9  #define Cins(a); scanf("%s",(a));
10 #define Cout(a,b); printf("%d",(a));printf(b);
11 #define Coutc(a,b); printf("%c",(a));printf(b);
12 #define Couts(a,b); printf("%s",(a));printf(b);
13 using namespace std;
14 typedef long long LL;
15 typedef unsigned long long ULL;
16 typedef long double LDB;
17 inline int readint() {int x;cin>>x;return x;}
18
19 LL tree[400005],lazy[400005];
20 int n,Q;
21 char cmd;
22 int x,y,a;
23 LL ans;
24 void pushdown(int o,int m)
25 {
```

```

26     lazy[o<<1] += lazy[o];
27     lazy[o<<1|1] += lazy[o];
28     int mm = (m>>1);
29     tree[o<<1] += (lazy[o] *(m-mm));
30     tree[o<<1|1] += (lazy[o] * mm);
31     lazy[o] = 0;
32     return;
33 }
34 void build(int o,int l,int r)
35 {
36     if(l == r){
37         scanf("%lld",&tree[o]); return;
38     }
39     int M = (l+r)>>1;
40     build(o<<1,l,M);
41     build(o<<1|1,M+1,r);
42     tree[o] = tree[o<<1]+tree[o<<1|1];
43     return ;
44 }
45 void query(int o,int l,int r,int L,int R)
46 {
47
48     if(l>=L && r<=R) {
49         ans+=tree[o];return;
50     }
51     if(lazy[o]) pushdown(o,r-l+1);
52     int M = (l+r)/2;
53     if(M+1<=R) query(o<<1|1,M+1,r,L,R);
54     if(M>=L) query(o<<1,l,M,L,R);
55 }
56 void update(int o,int l,int r,int L,int R,int d)
57 {
58     if(l>=L && r<=R){
59         lazy[o] += d;
60         tree[o] += (LL)d*(r-l+1);
61         return;
62     }
63     if(lazy[o]) pushdown(o,r-l+1);
64     int M = (l+r)>>1;
65     if(M>=L)
66         update(o<<1,l,M,L,R,d);
67     if(M+1<=R)
68         update(o<<1|1,M+1,r,L,R,d);
69     tree[o] = tree[o<<1] + tree[o<<1|1];
70 }
71 int main()

```

```
72 {
73     Cin(n);Cin(Q);
74     build(1,1,n);
75     For(i,1,Q)
76     {
77         Cinc(cmd);
78         if(cmd == 'Q'){
79             Cin(x);Cin(y);ans=0LL;
80             query(1,1,n,x,y);
81             cout<<ans<<endl;
82         }
83         else if(cmd == 'C'){
84             Cin(x);Cin(y);Cin(a);
85             update(1,1,n,x,y,a);
86         }
87     }
88 }
89 }
```