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

#define mp make pair
#define F first
#define S second
typedef long long int ll;
typedef pair<ll,ll> ii;
typedef vector<ll> vi;
typedef vector<ii > vii;

vi A;
vii st;
int n,m;

int getMax(vector<ll> e){
    ll s = -1;
    int p=-1;
    for(int i=0;i<e.size();i++){
        if(e[i]!=-1 and s < A[e[i]]){
            s = A[e[i]];
            p = i;
        }
    }
    return p;
}

void build(int P,int L,int R){
    if(L==R){
        st[P] = mp(L,-1);
        return;
    }
    if(L>R || R<L)
        return;
    int nxt = P << 1;
    int mid = (L+R) >> 1;
    build(nxt,L,mid);
    build(nxt+1,mid+1,R);

    vi e;
    ll s1,s2;
    int p;
    e.push back(st[nxt].F);
    e.push back(st[nxt].S);
    e.push back(st[nxt+1].F);
    e.push back(st[nxt+1].S);
    p = getMax(e);
    s1 = p==-1 ? -1:e[p];
    e[p]=-1;
    p = getMax(e);
    s2 = p==-1 ? -1:e[p];

    st[P] = mp(s1,s2);

}

void update (int p, int L, int R, int i, int value) {

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

    // total overlap
    if(L == R and L == i) {
        A[i] = value;
        st[p] = mp(i,-1);
        return;
    }

    int nxt = p << 1;
    int mid = (L + R) >> 1;
```

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

    vi e;
    ll s1,s2;
    int V;
    e.push_back(st[nxt].F);
    e.push_back(st[nxt].S);
    e.push_back(st[nxt+1].F);
    e.push_back(st[nxt+1].S);
    V = getMax(e);
    s1 = V==-1? -1:e[V];
    e[V]=-1;
    V = getMax(e);
    s2 = V==-1? -1:e[V];
    st[p].F = s1;
    st[p].S = s2;
}
ii query(int p, int L, int R, int i, int j){
    // no overlap
    if(i>R || j<L) return mp(-1,-1);

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

    // partial overlap
    int nxt = p << 1;
    int mid = (L + R) >> 1;
    ii p1 = query(nxt,L,mid,i,j);
    ii p2 = query(nxt + 1,mid +1,R,i,j);

    if(p1.F==-1 and p1.S==-1) return p2;
    if(p2.F==-1 and p2.S==-1) return p1;

    vi e;
    ll s1,s2;
    int V;
    e.push_back(p1.F);
    e.push_back(p1.S);
    e.push_back(p2.F);
    e.push_back(p2.S);
    V = getMax(e);
    s1 = V==-1? -1:e[V];
    e[V] = -1;
    V = getMax(e);
    s2 = e[V];

    return mp(s1,s2);

}

main(){
    int i,j,k,a,b;
    char o;
    cin >> n;
    st.resize(4*n);
    A.resize(2*n);
    st.assign(4*n,mp(-1,-1));
    A.assign(2*n,-1);
    for(i=0;i<n;i++){
        cin >> A[i];
    }
    build(1,0,n-1);
    cin >> m;
    for(i=0;i<m;i++){
        cin >> o >> a >> b;
        if(o=='Q'){
            ii aux = query(1,0,n-1,a-1,b-1);
            cout << A[aux.F]+A[aux.S]  << endl;
        }
        else{
            update(1,0,n-1,a-1,b);
        }
    }

}
```