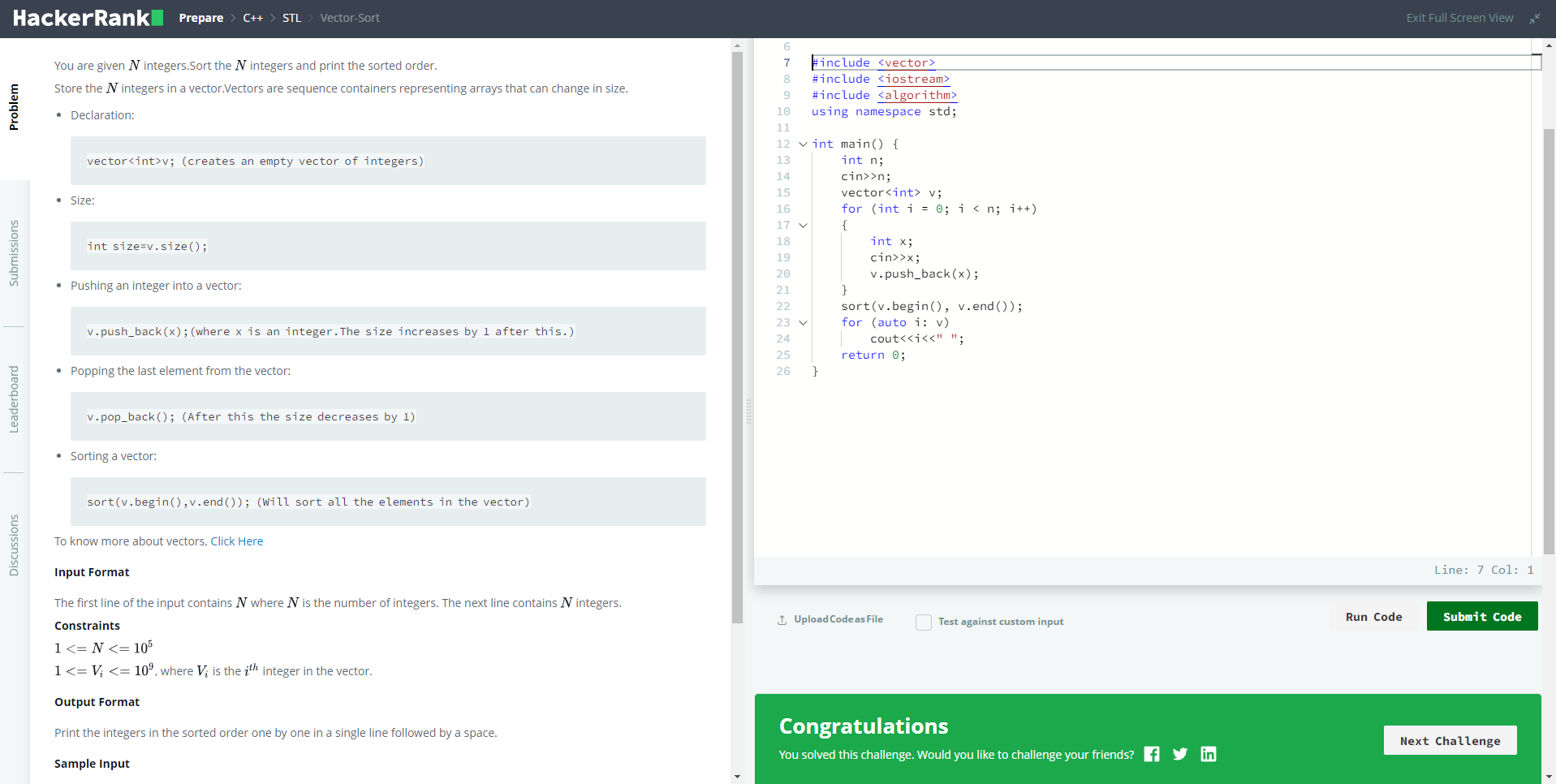
**Vector-Sort:**



#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

int main() {

int n;

cin>>n;

vector<int> v;

for (int i = 0; i < n; i++)

{

int x;

cin>>x;

v.push\_back(x);

}

sort(v.begin(), v.end());

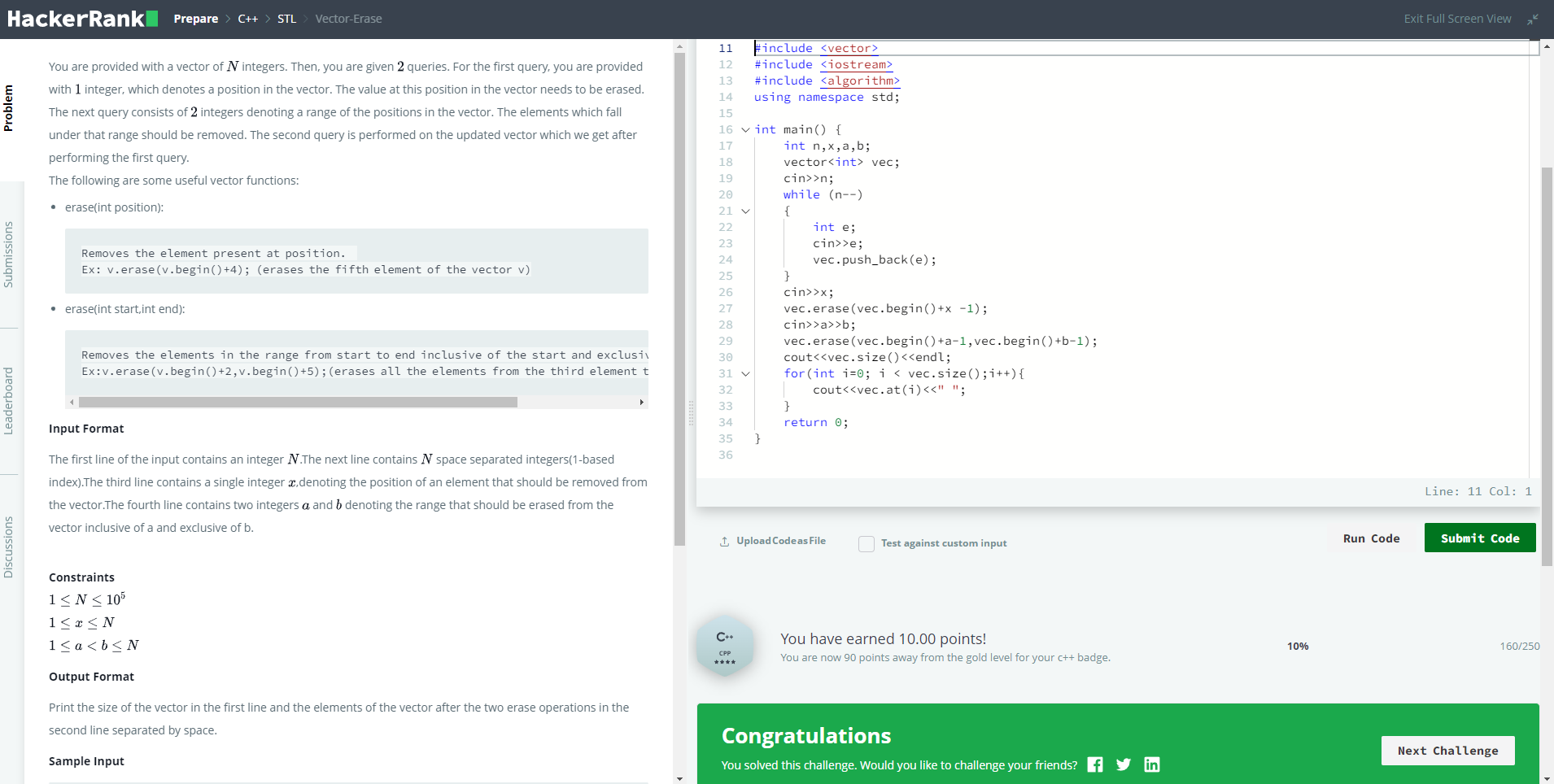
for (auto i: v)

cout<<i<<" ";

return 0;

}

**Vector-Erase:**



#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

int main() {

int n,x,a,b;

vector<int> vec;

cin>>n;

while (n--)

{

int e;

cin>>e;

vec.push\_back(e);

}

cin>>x;

vec.erase(vec.begin()+x -1);

cin>>a>>b;

vec.erase(vec.begin()+a-1,vec.begin()+b-1);

cout<<vec.size()<<endl;

for(int i=0; i < vec.size();i++){

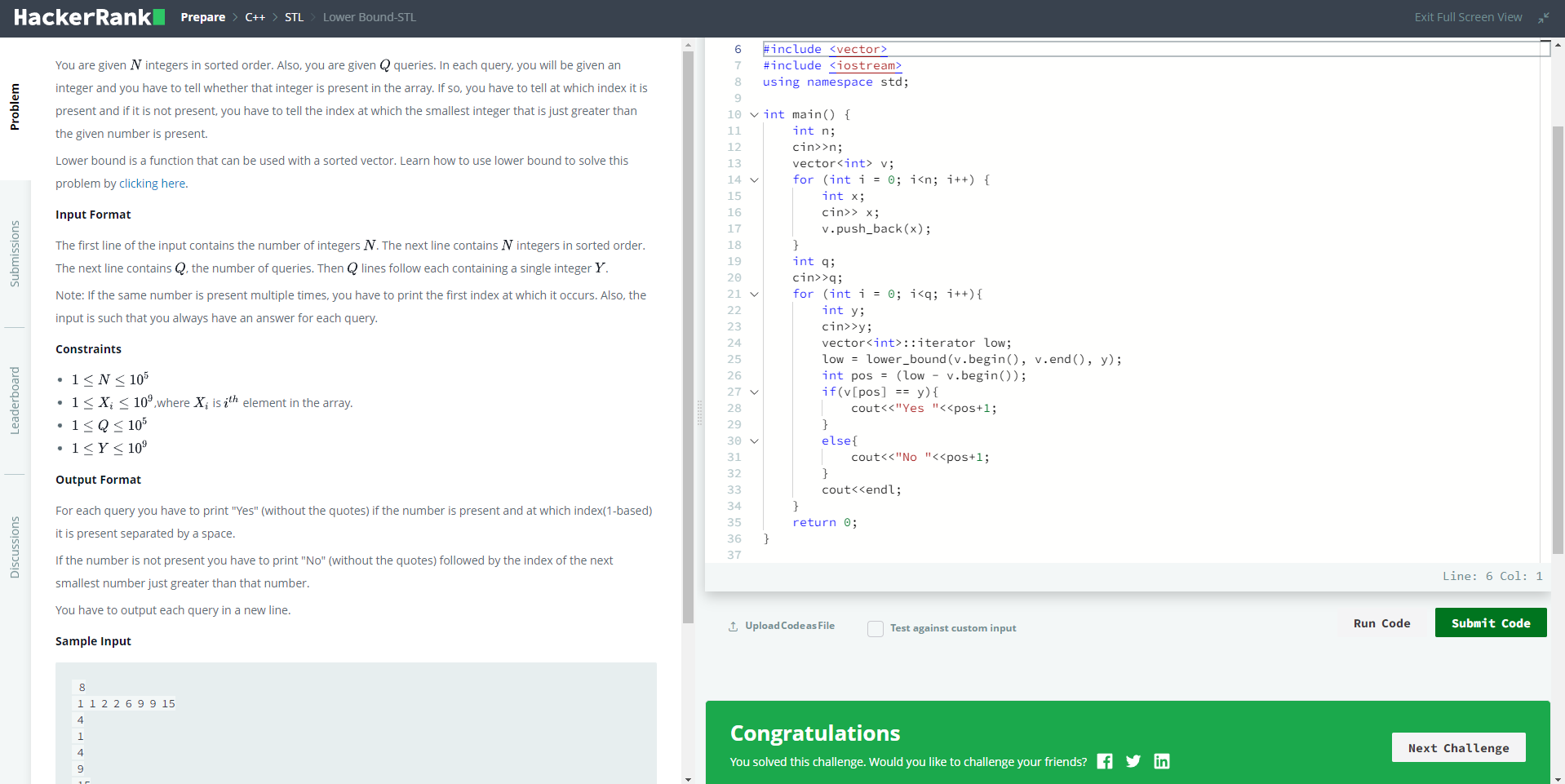
cout<<vec.at(i)<<" ";

}

return 0;

}

**Lower Bound-STL:**



#include <vector>

#include <iostream>

using namespace std;

int main() {

int n;

cin>>n;

vector<int> v;

for (int i = 0; i<n; i++) {

int x;

cin>> x;

v.push\_back(x);

}

int q;

cin>>q;

for (int i = 0; i<q; i++){

int y;

cin>>y;

vector<int>::iterator low;

low = lower\_bound(v.begin(), v.end(), y);

int pos = (low - v.begin());

if(v[pos] == y){

cout<<"Yes "<<pos+1;

}

else{

cout<<"No "<<pos+1;

}

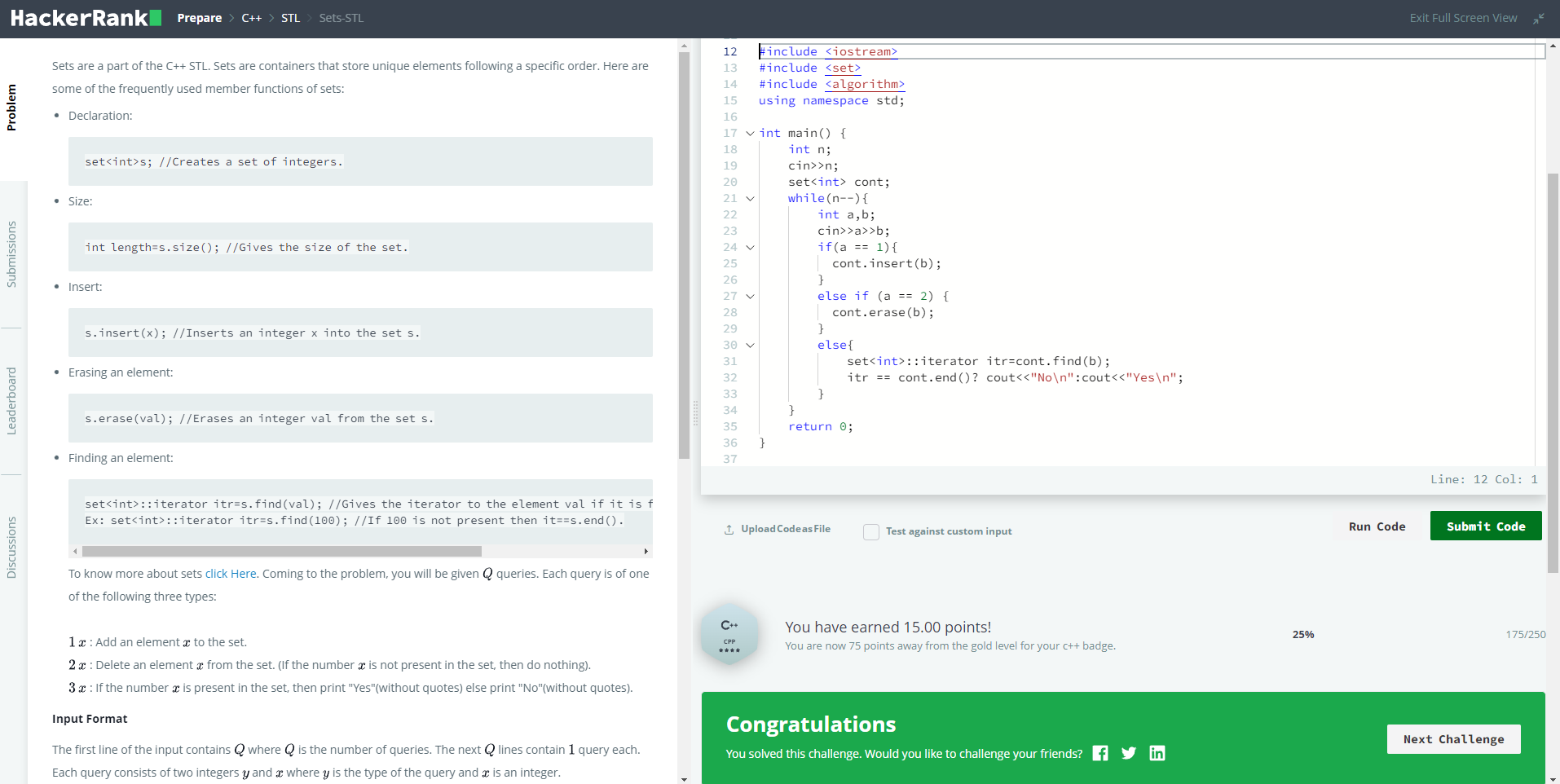
cout<<endl;

}

return 0;

}

**Set-STL:**



#include <iostream>

#include <set>

#include <algorithm>

using namespace std;

int main() {

int n;

cin>>n;

set<int> cont;

while(n--){

int a,b;

cin>>a>>b;

if(a == 1){

cont.insert(b);

}

else if (a == 2) {

cont.erase(b);

}

else{

set<int>::iterator itr=cont.find(b);

itr == cont.end()? cout<<"No\n":cout<<"Yes\n";

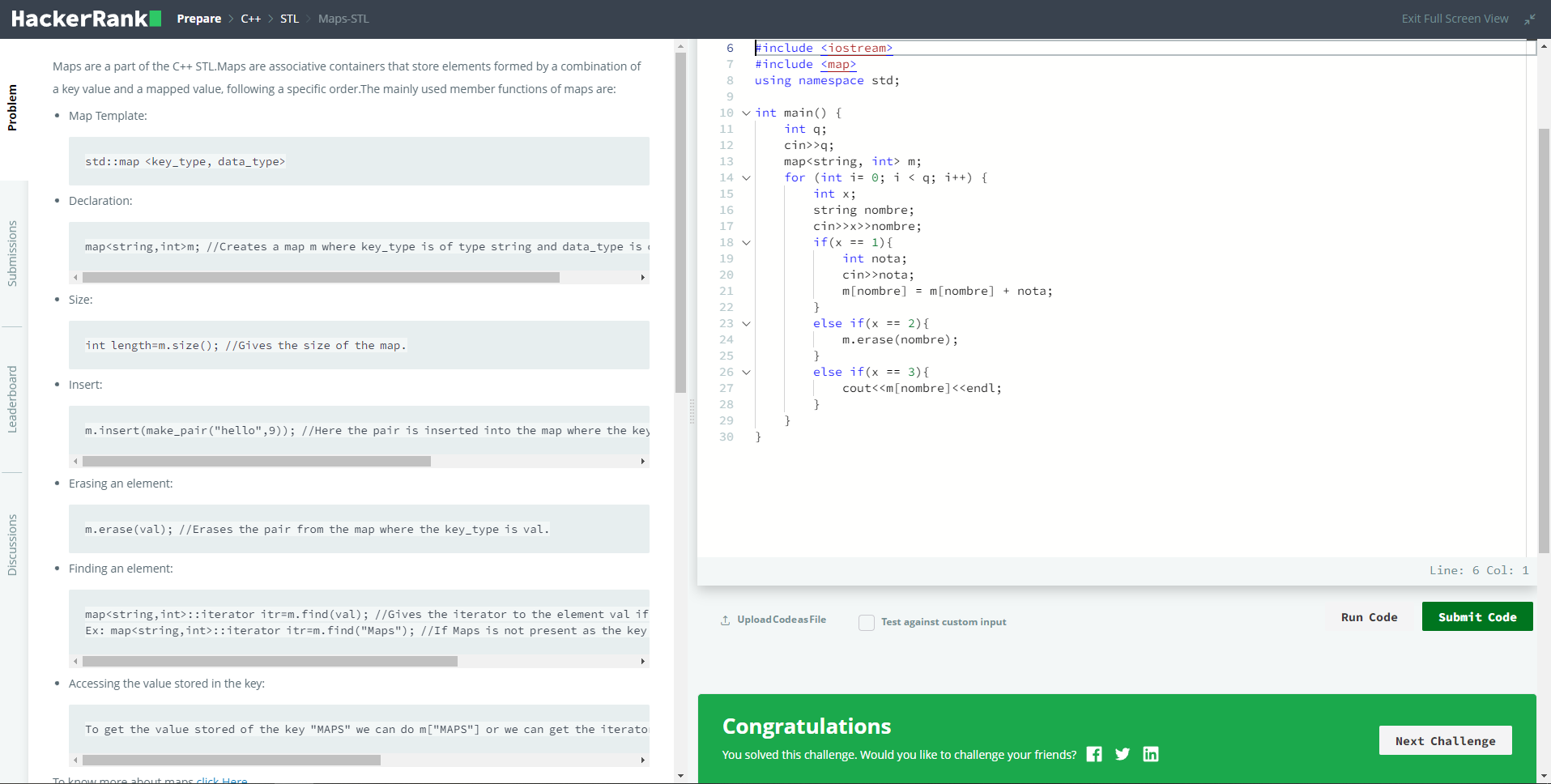
}

}

return 0;

}

**Map-STL:**



#include <iostream>

#include <map>

using namespace std;

int main() {

int q;

cin>>q;

map<string, int> m;

for (int i= 0; i < q; i++) {

int x;

string nombre;

cin>>x>>nombre;

if(x == 1){

int nota;

cin>>nota;

m[nombre] = m[nombre] + nota;

}

else if(x == 2){

m.erase(nombre);

}

else if(x == 3){

cout<<m[nombre]<<endl;

}

}

}