

**1.) Lower Bound-STL**

```
#include <cmath>
#include <cstdio>
#include <vector>
#include <iostream>
#include <algorithm>
using namespace std;

int main() {
    /* Enter your code here. Read input from STDIN. Print output to STDOUT */

    int n;
    cin >> n;

    vector<int> v(n);
    for (int i = 0; i < n; i++) {
        cin >> v[i];
    }

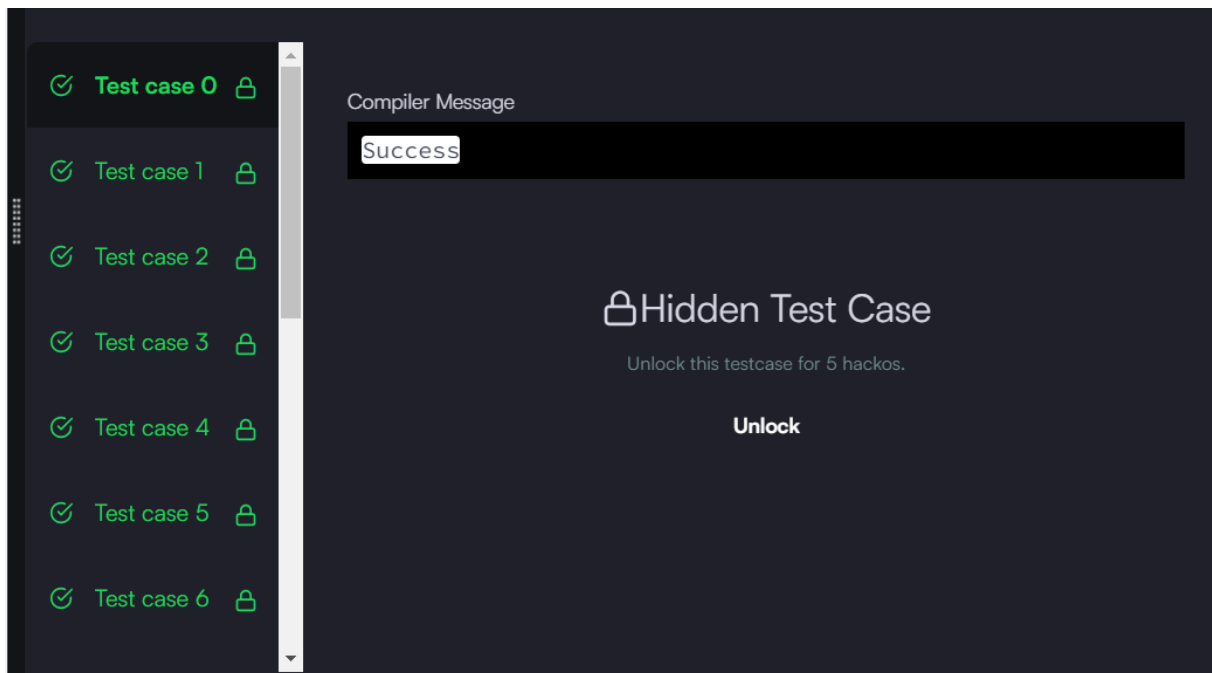
    int q;
    cin >> q;

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

        vector<int>::iterator low = lower_bound(v.begin(), v.end(), x);

        if (v[low - v.begin()] == x) {
            cout << "Yes " << (low - v.begin() + 1) << endl;
        } else {
            cout << "No " << (low - v.begin() + 1) << endl;
        }
    }
}
```

```
    }  
}  
return 0;  
}
```



**2.) Sets-STL**

```
#include <cmath>
#include <cstdio>
#include <vector>
#include <iostream>
#include <set>
#include <algorithm>
using namespace std;

int main() {
    /* Enter your code here. Read input from STDIN. Print output to STDOUT */
    set<int> s;
    int Q;
    cin>>Q;
    for(int i=0; i<Q; i++){
        int query;
        int x;
        cin>>query;
        cin>>x;
        switch (query) {
            case 1:
                s.insert(x);
                break;
            case 2:
                s.erase(x);
                break;
            case 3:
                auto itr=s.find(x);

                if(itr !=s.end()) {
                    cout<<"Yes"<<endl;
```

```
    }  
    else{  
        cout<<"No"<<endl;  
    }  
    break;  
}  
}  
return 0;  
}
```

The screenshot shows a HackerRank submission interface. On the left, a list of test cases is displayed, each with a green checkmark and a lock icon. The test cases are labeled 'Test case 20' through 'Test case 26'. A vertical scrollbar is visible next to the list. On the right, the 'Compiler Message' section shows a 'Success' message. Below this, a 'Hidden Test Case' section is visible, indicating that the test case is locked and can be unlocked for 5 hackos. An 'Unlock' button is present.

Compiler Message

Success

Hidden Test Case

Unlock this testcase for 5 hackos.

Unlock

**3.) Maps-STL**

```
#include <cmath>
#include <cstdio>
#include <vector>
#include <iostream>
#include <set>
#include <map>
#include <algorithm>
using namespace std;

int main() {
    /* Enter your code here. Read input from STDIN. Print output to STDOUT */

    int n;
    int query;
    map<string, int> students;
    cin>>n;
    string name;
    int marks;
    for(int i=0;i<n;i++)
    {
        cin>>query;
        if(query==1)
        {
            cin>>name>>marks;
            students[name]+=marks;
        }
        if(query==2 || query==3)
        {
            cin>>name;
        }
    }
```

```
switch(query)
{
case 1:
students.insert(make_pair(name,marks));
break;

case 2:
students.erase(name);
break;

case 3:
cout<<students[name]<<"\n";
break;
}

return 0;
}
```

The screenshot shows a HackerRank interface with a dark theme. On the left, a vertical list of test cases is visible, each with a green checkmark and a lock icon. The test cases are labeled 'Test case 0' through 'Test case 6'. The main area on the right is titled 'Compiler Message' and displays a 'Success' message. Below this, a large 'Hidden Test Case' notification is shown, featuring a lock icon and the text 'Unlock this testcase for 5 hackos.' and an 'Unlock' button.

Test case 0

Test case 1

Test case 2

Test case 3

Test case 4

Test case 5

Test case 6

Compiler Message

Success

Hidden Test Case

Unlock this testcase for 5 hackos.

Unlock

#### 4.) Inheritance Introduction

```
#include <cmath>

#include <cstdio>

#include <vector>

#include <iostream>

#include <algorithm>


using namespace std;

class Triangle{
public:
    void triangle(){
        cout<<"I am a triangle\n";
    }
};

class Isosceles : public Triangle{
public:
    void isosceles(){
        cout<<"I am an isosceles triangle\n";
    }
    //Write your code here.
    void description() {
        cout << "In an isosceles triangle two sides are equal\n";
    }
};

int main(){
    Isosceles isc;
    isc.isosceles();
    isc.description();
}
```

```
isc.triangle();  
return 0;  
}
```

 **Test case 0**

Compiler Message

**Success**

Expected Output

[Download](#)

1	<b>I am an isosceles triangle</b>
2	<b>In an isosceles triangle two sides are equal</b>
3	<b>I am a triangle</b>



**5.) Rectangle Area**

```
#include <iostream>

using namespace std;

/*
 * Create classes Rectangle and RectangleArea
 */

class Rectangle{
public:
    int height,width;
    void display(){
        cout<<width<<" "<<height<<endl;
    }
};

class RectangleArea: public Rectangle{
public:
    void read_input(){
        cin>>width>>height;
    }
    void display(){
        cout<<(width*height);
    }
};

int main()
{
    /*
     * Declare a RectangleArea object
     */
```

```
RectangleArea r_area;  
  
/*  
 * Read the width and height  
 */  
r_area.read_input();  
  
/*  
 * Print the width and height  
 */  
r_area.Rectangle::display();  
  
/*  
 * Print the area  
 */  
r_area.display();  
  
return 0;  
}
```

The screenshot displays the HackerRank submission interface. On the left, a sidebar lists test cases 4 through 10, all marked as successful with green checkmarks. The main area on the right shows the 'Compiler Message' as 'Success'. Below this, the 'Input (stdin)' section shows a single test case with input '10 5'. The 'Expected Output' section shows two lines of output: '10 5' for the first line and '50' for the second line. Each of these sections has a 'Download' link to the right.

Test Case	Status
Test case 4	Success
Test case 5	Success
Test case 6	Success
Test case 7	Success
Test case 8	Success
Test case 9	Success
Test case 10	Success

Compiler Message: Success

Input (stdin): 1 10 5

Expected Output: 1 10 5, 2 50