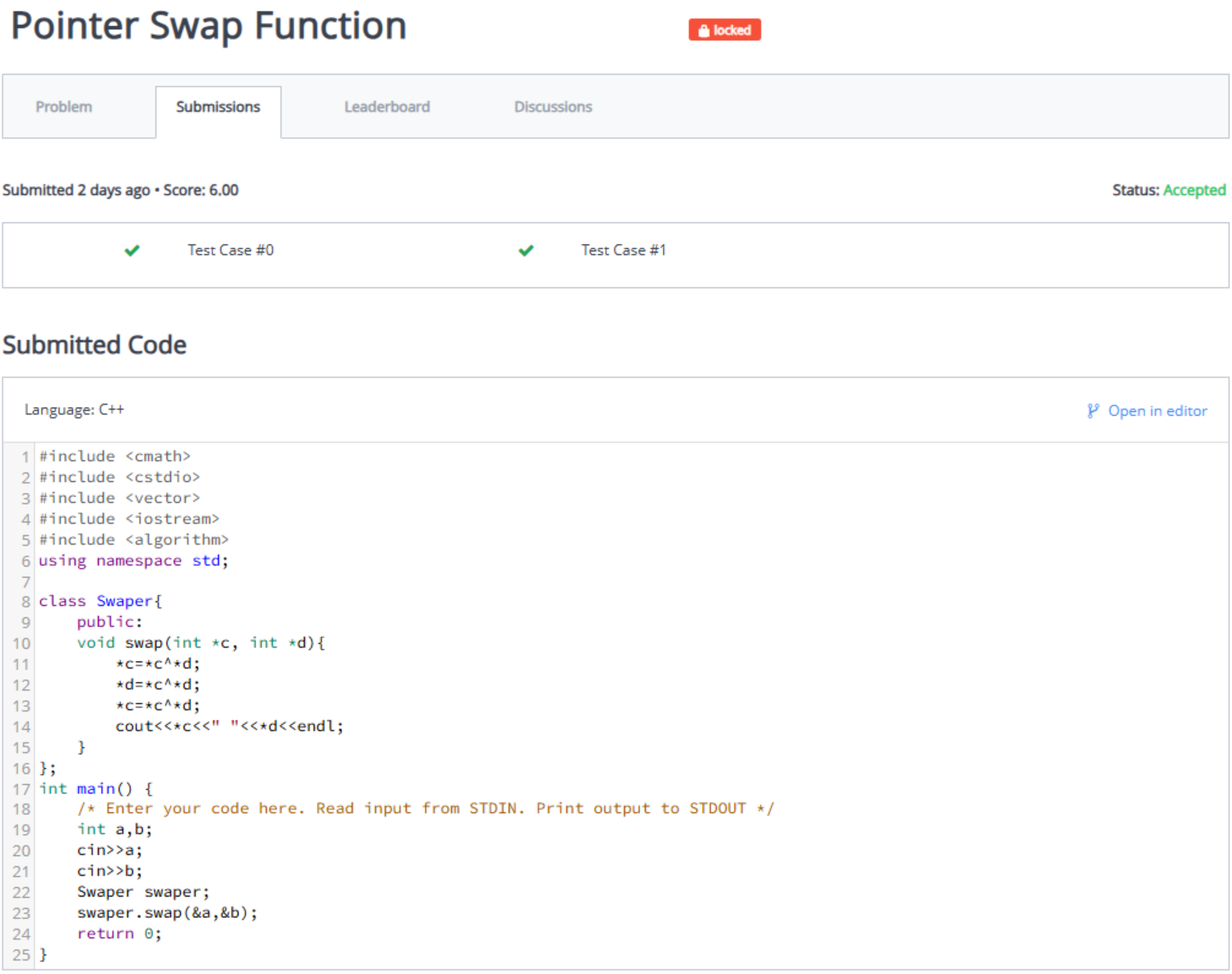
# Nishanth P – 24BAD405

# Week 3

## 1. Pointer Swap Function



Pointer Swap Function

Submitted 2 days ago • Score: 6.00

Status: Accepted

✓ Test Case #0 ✓ Test Case #1

Submitted Code

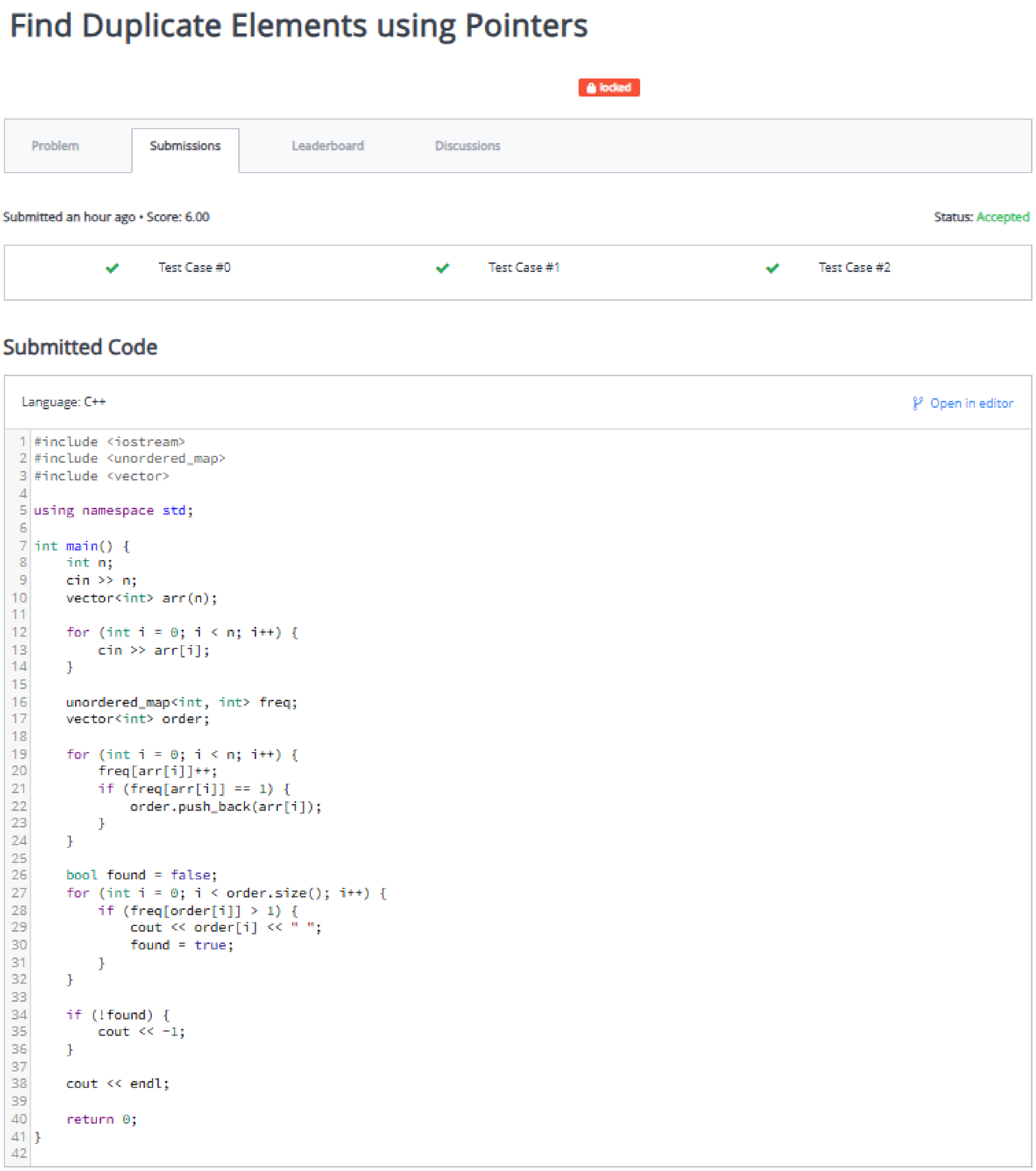
Language: C++

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

class Swaper{
    public:
    void swap(int *c, int *d){
        *c=*c^*d;
        *d=*c^*d;
        *c=*c^*d;
        cout<<*c<<" "<<*d<<endl;
    }
};
int main() {
    /* Enter your code here. Read input from STDIN. Print output to STDOUT */
    int a,b;
    cin>>a;
    cin>>b;
    Swaper swaper;
    swaper.swap(&a,&b);
    return 0;
}
```

# Nishanth P – 24BAD405

# Week 3

## 2. Find Duplicate Elements using Pointers



### Find Duplicate Elements using Pointers

Submitted an hour ago • Score: 6.00

Status: Accepted

✓ Test Case #0 ✓ Test Case #1 ✓ Test Case #2

### Submitted Code

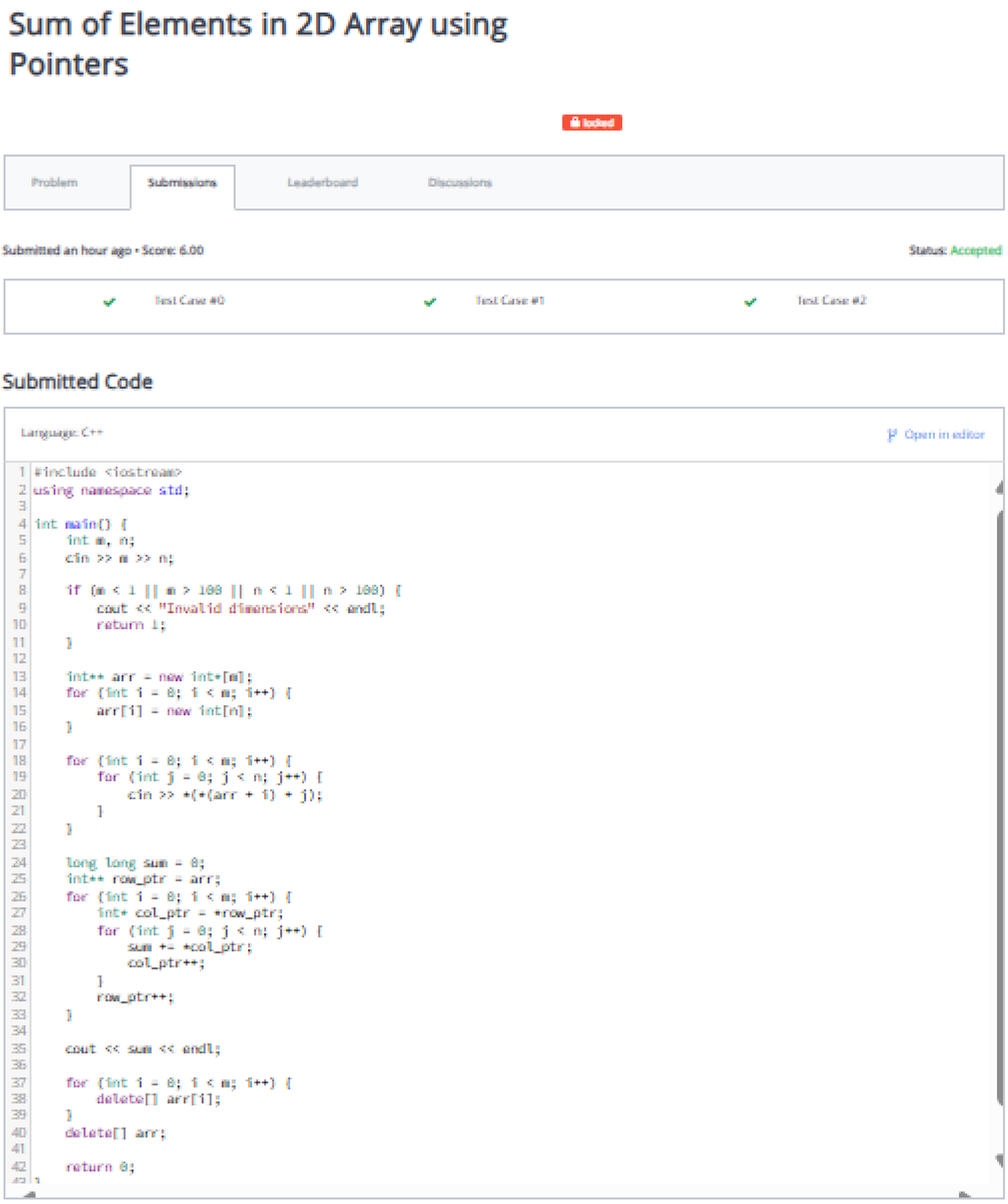
Language: C++

```cpp
#include <iostream>
#include <unordered_map>
#include <vector>

using namespace std;

int main() {
    int n;
    cin >> n;
    vector<int> arr(n);

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

    unordered_map<int, int> freq;
    vector<int> order;

    for (int i = 0; i < n; i++) {
        freq[arr[i]]++;
        if (freq[arr[i]] == 1) {
            order.push_back(arr[i]);
        }
    }

    bool found = false;
    for (int i = 0; i < order.size(); i++) {
        if (freq[order[i]] > 1) {
            cout << order[i] << " ";
            found = true;
        }
    }

    if (!found) {
        cout << -1;
    }

    cout << endl;

    return 0;
}

```

# Nishanth P – 24BAD405

# Week 3

## 3. Sum of Elements in 2D Array using Pointers



Sum of Elements in 2D Array using Pointers

Submitted an hour ago • Score: 6.00

Status: Accepted

✓ Test Case #0 ✓ Test Case #1 ✓ Test Case #2

Submitted Code

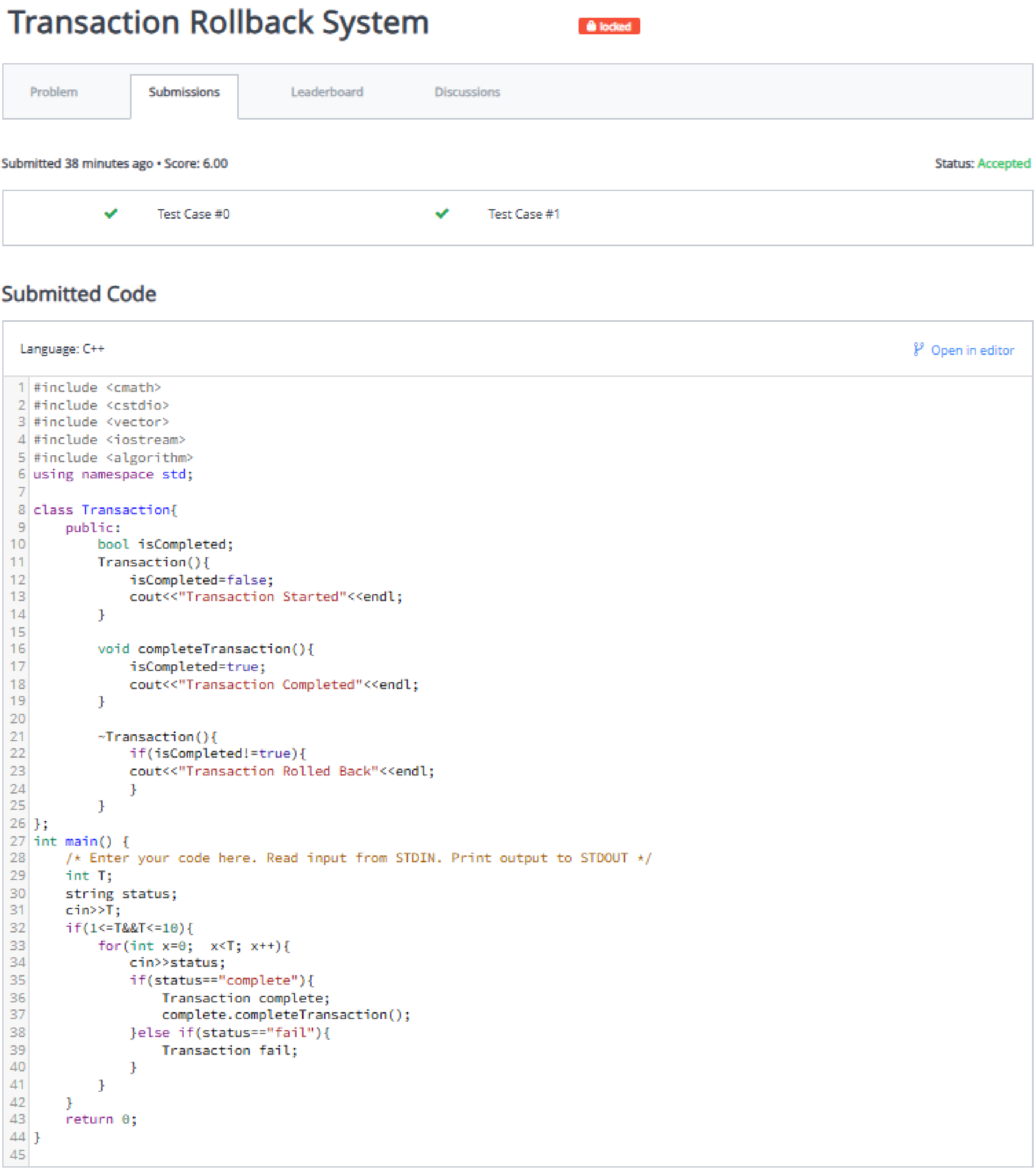
Language: C++

```cpp
#include <iostream>
using namespace std;

int main() {
    int m, n;
    cin >> m >> n;

    if (m < 1 || m > 100 || n < 1 || n > 100) {
        cout << "Invalid dimensions" << endl;
        return 1;
    }

    int** arr = new int*[m];
    for (int i = 0; i < m; i++) {
        arr[i] = new int[n];
    }

    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            cin >> *(*(arr + i) + j);
        }
    }

    long long sum = 0;
    int** row_ptr = arr;
    for (int i = 0; i < m; i++) {
        int* col_ptr = *row_ptr;
        for (int j = 0; j < n; j++) {
            sum += *col_ptr;
            col_ptr++;
        }
        row_ptr++;
    }

    cout << sum << endl;

    for (int i = 0; i < m; i++) {
        delete[] arr[i];
    }
    delete[] arr;

    return 0;
}
```

# Nishanth P – 24BAD405

# Week 3

## 4. Transaction Rollback System



### Transaction Rollback System

Submitted 38 minutes ago • Score: 6.00

Status: Accepted

✓ Test Case #0

✓ Test Case #1

### Submitted Code

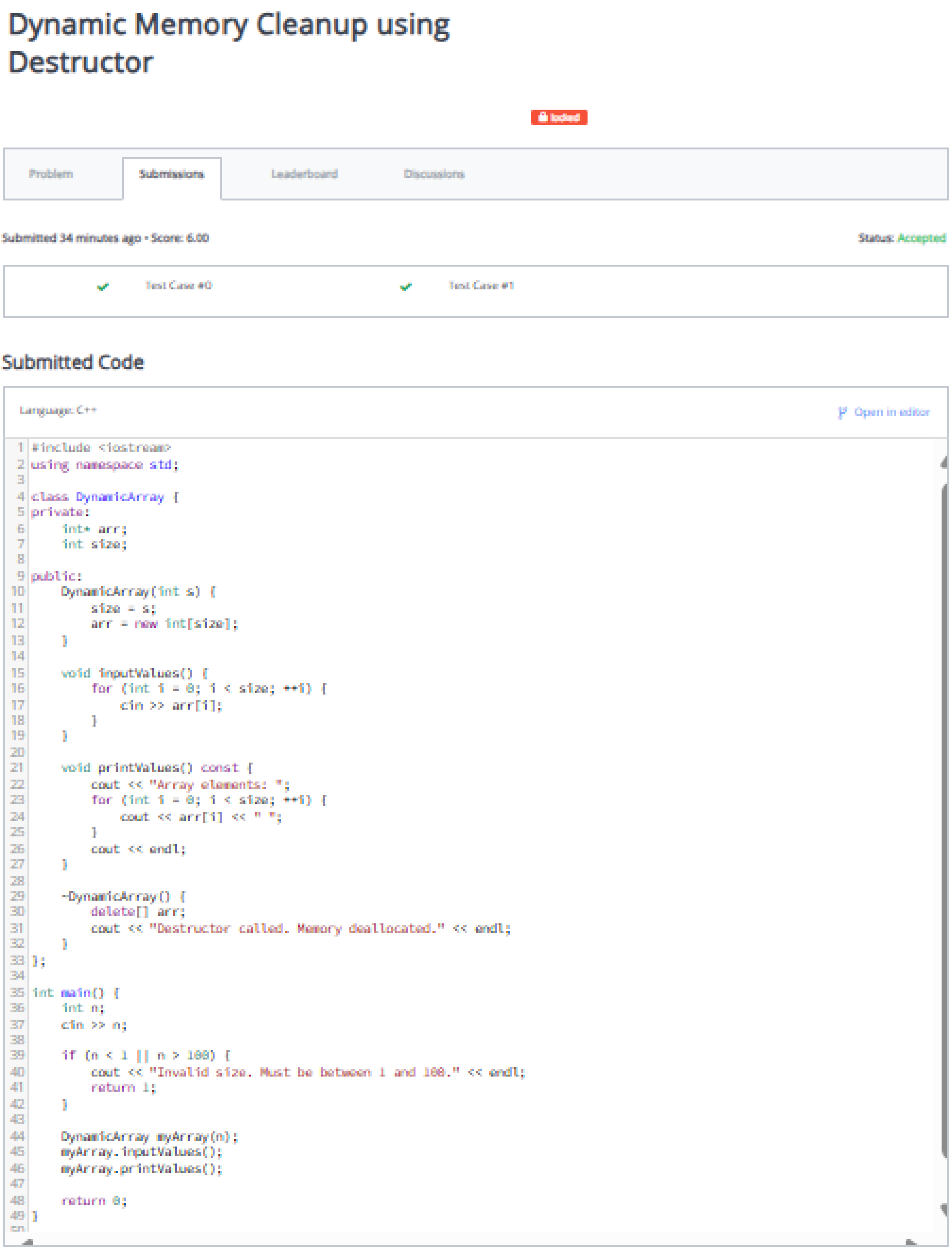
Language: C++

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

class Transaction{
    public:
        bool isCompleted;
        Transaction(){
            isCompleted=false;
            cout<<"Transaction Started"<<endl;
        }

        void completeTransaction(){
            isCompleted=true;
            cout<<"Transaction Completed"<<endl;
        }

        ~Transaction(){
            if(isCompleted!=true){
            cout<<"Transaction Rolled Back"<<endl;
            }
        }
};
int main() {
    /* Enter your code here. Read input from STDIN. Print output to STDOUT */
    int T;
    string status;
    cin>>T;
    if(1<=T&&T<=10){
        for(int x=0;  x<T; x++){
            cin>>status;
            if(status=="complete"){
                Transaction complete;
                complete.completeTransaction();
            }else if(status=="fail"){
                Transaction fail;
            }
        }
    }
    return 0;
}

```

# Nishanth P – 24BAD405

# Week 3

## 5. Dynamic Memory Cleanup using Destructor



Dynamic Memory Cleanup using Destructor

Submitted 34 minutes ago • Score: 6.00

Status: Accepted

✓ Test Case #0

✓ Test Case #1

Submitted Code

Language: C++

```cpp
#include <iostream>
using namespace std;

class DynamicArray {
private:
    int* arr;
    int size;

public:
    DynamicArray(int s) {
        size = s;
        arr = new int[size];
    }

    void inputValues() {
        for (int i = 0; i < size; ++i) {
            cin >> arr[i];
        }
    }

    void printValues() const {
        cout << "Array elements: ";
        for (int i = 0; i < size; ++i) {
            cout << arr[i] << " ";
        }
        cout << endl;
    }

    ~DynamicArray() {
        delete[] arr;
        cout << "Destructor called. Memory deallocated." << endl;
    }
};

int main() {
    int n;
    cin >> n;

    if (n < 1 || n > 100) {
        cout << "Invalid size. Must be between 1 and 100." << endl;
        return 1;
    }

    DynamicArray myArray(n);
    myArray.inputValues();
    myArray.printValues();

    return 0;
}
```