

Week 3

1. Pointer Swap Function

Pointer Swap Function

locked

Problem

Submissions

Leaderboard

Discussions

Submitted 2 days ago • Score: 6.00

Status: Accepted



Test Case #0



Test Case #1

Submitted Code

Language: C++

[Open in editor](#)

```
1 #include <cmath>
2 #include <cstdio>
3 #include <vector>
4 #include <iostream>
5 #include <algorithm>
6 using namespace std;
7
8 class Swaper{
9     public:
10     void swap(int *c, int *d){
11         *c=*c^*d;
12         *d=*c^*d;
13         *c=*c^*d;
14         cout<<*c<<" "<<*d<<endl;
15     }
16 };
17 int main() {
18     /* Enter your code here. Read input from STDIN. Print output to STDOUT */
19     int a,b;
20     cin>>a;
21     cin>>b;
22     Swaper swaper;
23     swaper.swap(&a,&b);
24     return 0;
25 }
```

Week 3

2. Find Duplicate Elements using Pointers

Find Duplicate Elements using Pointers

locked

Problem	Submissions	Leaderboard	Discussions
---------	-------------	-------------	-------------

Submitted an hour ago • Score: 6.00

Status: Accepted

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

Submitted Code

Language: C++

[Open in editor](#)

```
1 #include <iostream>
2 #include <unordered_map>
3 #include <vector>
4
5 using namespace std;
6
7 int main() {
8     int n;
9     cin >> n;
10    vector<int> arr(n);
11
12    for (int i = 0; i < n; i++) {
13        cin >> arr[i];
14    }
15
16    unordered_map<int, int> freq;
17    vector<int> order;
18
19    for (int i = 0; i < n; i++) {
20        freq[arr[i]]++;
21        if (freq[arr[i]] == 1) {
22            order.push_back(arr[i]);
23        }
24    }
25
26    bool found = false;
27    for (int i = 0; i < order.size(); i++) {
28        if (freq[order[i]] > 1) {
29            cout << order[i] << " ";
30            found = true;
31        }
32    }
33
34    if (!found) {
35        cout << -1;
36    }
37
38    cout << endl;
39
40    return 0;
41 }
42
```

Week 3

3. Sum of Elements in 2D Array using Pointers

Sum of Elements in 2D Array using Pointers

Locked

Problem	Submissions	Leaderboard	Discussions
---------	-------------	-------------	-------------

Submitted an hour ago • Score: 6.00

Status: Accepted

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

Submitted Code

```
Language: C++  
1 #include <iostream>  
2 using namespace std;  
3  
4 int main() {  
5     int m, n;  
6     cin >> m >> n;  
7  
8     if (m < 1 || m > 100 || n < 1 || n > 100) {  
9         cout << "Invalid dimensions" << endl;  
10        return 1;  
11    }  
12  
13    int** arr = new int*[m];  
14    for (int i = 0; i < m; i++) {  
15        arr[i] = new int[n];  
16    }  
17  
18    for (int i = 0; i < m; i++) {  
19        for (int j = 0; j < n; j++) {  
20            cin >> *(arr + i) + j);  
21        }  
22    }  
23  
24    long long sum = 0;  
25    int** row_ptr = arr;  
26    for (int i = 0; i < m; i++) {  
27        int* col_ptr = *row_ptr;  
28        for (int j = 0; j < n; j++) {  
29            sum += *col_ptr;  
30            col_ptr++;  
31        }  
32        row_ptr++;  
33    }  
34  
35    cout << sum << endl;  
36  
37    for (int i = 0; i < m; i++) {  
38        delete[] arr[i];  
39    }  
40    delete[] arr;  
41  
42    return 0;  
43 }
```

Week 3

4. Transaction Rollback System

Transaction Rollback System

locked

Problem

Submissions

Leaderboard

Discussions

Submitted 38 minutes ago • Score: 6.00

Status: Accepted



Test Case #0



Test Case #1

Submitted Code

Language: C++

[Open in editor](#)

```
1 #include <cmath>
2 #include <cstdio>
3 #include <vector>
4 #include <iostream>
5 #include <algorithm>
6 using namespace std;
7
8 class Transaction{
9     public:
10         bool isCompleted;
11         Transaction(){
12             isCompleted=false;
13             cout<<"Transaction Started"<<endl;
14         }
15
16         void completeTransaction(){
17             isCompleted=true;
18             cout<<"Transaction Completed"<<endl;
19         }
20
21         ~Transaction(){
22             if(isCompleted!=true){
23                 cout<<"Transaction Rolled Back"<<endl;
24             }
25         }
26 };
27 int main() {
28     /* Enter your code here. Read input from STDIN. Print output to STDOUT */
29     int T;
30     string status;
31     cin>>T;
32     if(1<=T&&T<=10){
33         for(int x=0; x<T; x++){
34             cin>>status;
35             if(status=="complete"){
36                 Transaction complete;
37                 complete.completeTransaction();
38             }else if(status=="fail"){
39                 Transaction fail;
40             }
41         }
42     }
43     return 0;
44 }
```

Week 3

5. Dynamic Memory Cleanup using Destructor

Dynamic Memory Cleanup using Destructor

Locked

Problem	Submissions	Leaderboard	Discussions
---------	-------------	-------------	-------------

Submitted 34 minutes ago • Score: 6.00

Status: Accepted

✓	Test Case #0	✓	Test Case #1
---	--------------	---	--------------

Submitted Code

```
Language: C++P Open in editor

1 #include <iostream>
2 using namespace std;
3
4 class DynamicArray {
5 private:
6     int* arr;
7     int size;
8
9 public:
10    DynamicArray(int s) {
11        size = s;
12        arr = new int[size];
13    }
14
15    void inputValues() {
16        for (int i = 0; i < size; ++i) {
17            cin >> arr[i];
18        }
19    }
20
21    void printValues() const {
22        cout << "Array elements: ";
23        for (int i = 0; i < size; ++i) {
24            cout << arr[i] << " ";
25        }
26        cout << endl;
27    }
28
29    ~DynamicArray() {
30        delete[] arr;
31        cout << "Destructor called. Memory deallocated." << endl;
32    }
33 };
34
35 int main() {
36     int n;
37     cin >> n;
38
39     if (n < 1 || n > 100) {
40         cout << "Invalid size. Must be between 1 and 100." << endl;
41         return 1;
42     }
43
44     DynamicArray myArray(n);
45     myArray.inputValues();
46     myArray.printValues();
47
48     return 0;
49 }
50
```