lab1.cpp

```cpp
#include<iostream>
#include <unordered_set>
using namespace std;
void display(vector<int>vec) {
    for (int i = 0; i < vec.size(); i++) {
        cout << vec[i] <<" ";
    }
    cout << endl;
}
void append(int x,vector<int>&vec){
    cout << "Before Appending ";
    for (auto t:vec){
        cout << t << " ";
    }
    cout << endl;
    vec.emplace_back(x);
    cout << "After Appending ";
    for (auto t:vec){
        cout << t << " ";
    }
    cout << endl;
}
void insert(int index,int x,vector<int>&vec){
    cout << "Before inserting ";
    for (auto t:vec){
        cout << t << " ";
    }
    cout << endl;
    if (index <= vec.size())
    {
        vec.insert(vec.begin() + index, x);
    }
    else
    {
        cout << "Invalid index";
    }
    cout << "After inserting ";
    for (auto t:vec){
        cout << t << " ";
    }
    cout << endl;
}
void Delete(int x,vector<int>&vec){
    cout << "Before deleting " <<x;
    cout << endl;
    display(vec);
    int index = -1;
    for (int i = 0; i < vec.size(); ++i)
    {
        if(vec[i] == x)
            index = i;
    }
    vec.erase (vec.begin()+index);
    cout << "After deleting " <<x;
    cout << endl;
    display(vec);
}
```

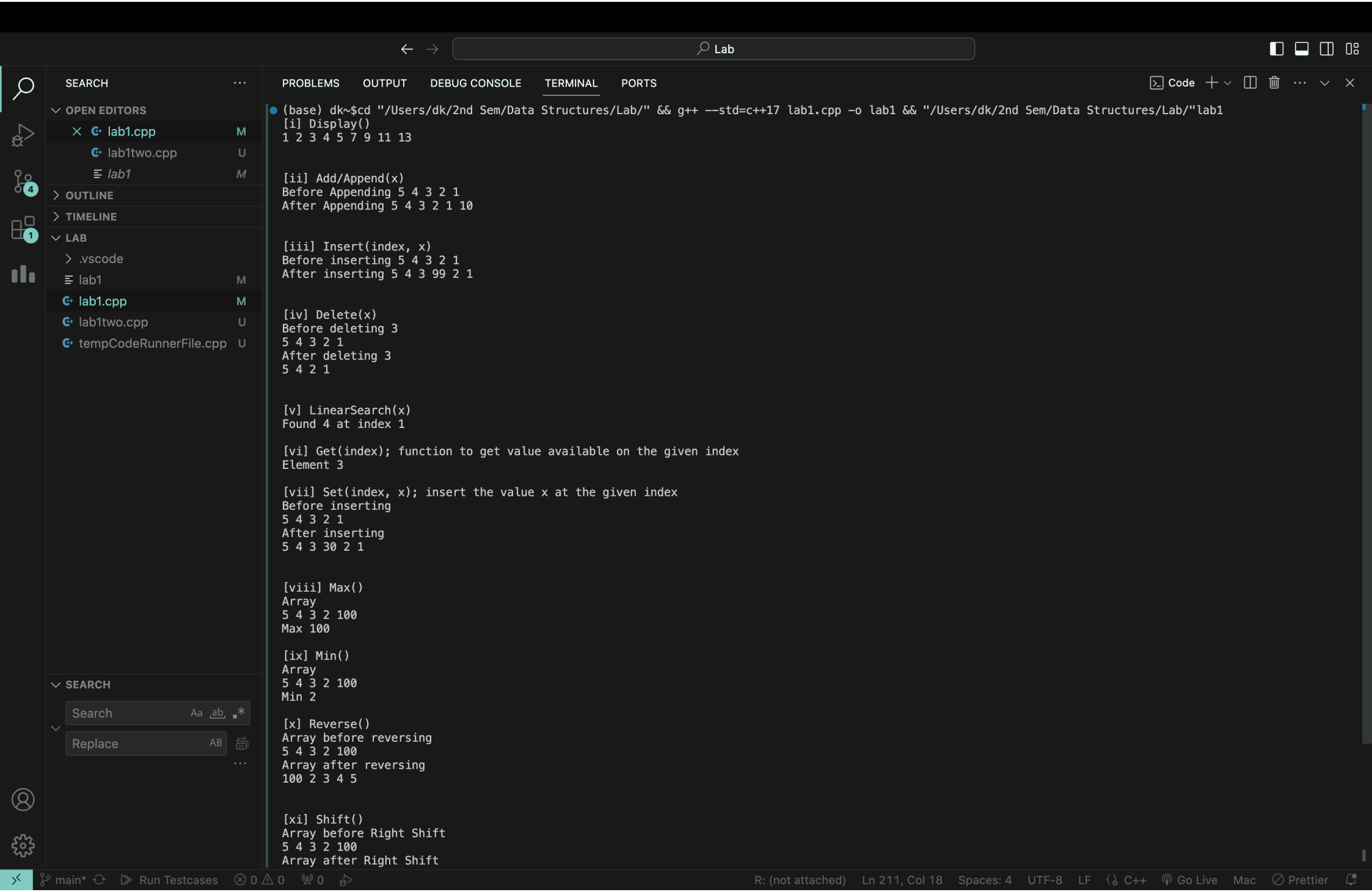
```cpp
void LinearSearch(int x,vector<int>&vec){
    int index = -1;
    for (int i = 0; i < vec.size(); ++i)
    {
        if(vec[i] == x)
            index = i;
    }
    if(index ==-1){
        cout << x << " Not found";
    }
    else
    {
        cout << "Found " << x <<" at index " << index;
    }
}
void Get(int index,vector<int>&vec){
    if(index > vec.size()){
        cout << "index out of bound ";
        return;
    }
    for (int i = 0; i < vec.size(); ++i)
    {
        if(i == index)
            cout << "Element " << vec[index];
    }
}
void Set(int index,int x,vector<int>&vec){
    cout << "Before inserting ";
    cout << endl;
    display(vec);
    if (index <= vec.size())
    {
        vec.insert(vec.begin() + index, x);
    }
    else
    {
        cout << "Invalid index";
    }
    cout << "After inserting ";
    cout << endl;
    display(vec);
}
void Max(vector<int>&vec){
    cout << "Array ";
    cout << endl;
    display(vec);
    int max = INT_MIN;
    for(auto it: vec){
        if(it > max)
            max = it;
    }
    cout << "Max " << max;
}
void Min(vector<int>&vec){
    cout << "Array ";
    cout << endl;
    display(vec);
    int min = INT_MAX;
    for(auto it: vec){
        if(it < min)
```

```
            min = it;
    }
    cout << "Min " << min;
}
void Reverse(vector<int>&vec){
    cout << "Array before reversing ";
    cout << endl;
    display(vec);
    int l = 0;
    int h = vec.size()-1;
    while(l<h){
        swap(vec[l],vec[h]);
        l++;
        h--;
    }
    cout << "Array after reversing ";
    cout << endl;
    display(vec);
}
void RightShift(vector<int>& vec, int k) {
    cout << "Array before Right Shift ";
    cout << endl;
    display(vec);
    for(int i=0;i<k;i++){
        vec.pop_back();
        vec.insert(vec.begin(),0);
    }
    cout << "Array after Right Shift ";
    cout << endl;
    display(vec);
}
void _reverseArray(vector<int>& nums,int s, int e){
    while(s<e){
        swap(nums[s],nums[e]);
        s++;
        e--;
    }
}
void Rotate(vector<int>& vec, int k) {
    cout << "Array before rotate by "<<k;
    cout << endl;
    display(vec);
    int length = vec.size();
    k = k % length;
    _reverseArray(vec,0,length-1);
    _reverseArray(vec,0,k-1);
    _reverseArray(vec,k,length-1);
    cout << "Array After rotate ";
    cout << endl;
    display(vec);
}

int main()
{
    cout << "[i] Display()";
    cout << endl;
    vector<int> myvecOne = {1, 2, 3, 4, 5, 7, 9, 11, 13};
    display(myvecOne);
    cout << endl;
    cout << endl;
```

```
178
179     cout << "[ii] Add/Append(x)";
180     cout << endl;
181     vector<int> myvecTwo = {5, 4, 3, 2, 1};
182     append(10, myvecTwo);
183     cout << endl;
184     cout << endl;
185
186     cout << "[iii] Insert(index, x)";
187     cout << endl;
188     vector<int> myvecThree = {5, 4, 3, 2, 1};
189     insert(3, 99, myvecThree);
190     cout << endl;
191     cout << endl;
192
193     cout << "[iv] Delete(x)";
194     cout << endl;
195     vector<int> myvecFour = {5, 4, 3, 2, 1};
196     Delete(3, myvecFour);
197     cout << endl;
198     cout << endl;
199
200     cout << "[v] LinearSearch(x)";
201     cout << endl;
202     vector<int> myvecFive = {5, 4, 3, 2, 1};
203     LinearSearch(4, myvecFive);
204     cout << endl;
205     cout << endl;
206
207     cout << "[vi] Get(index); function to get value available on the given index";
208     cout << endl;
209     vector<int> myvecSix = {5, 4, 3, 2, 1};
210     Get(2, myvecSix);
211     cout << endl;
212     cout << endl;
213
214     cout << "[vii] Set(index, x); insert the value x at the given index";
215     cout << endl;
216     vector<int> myvecSeven = {5, 4, 3, 2, 1};
217     Set(3,30, myvecSeven);
218     cout << endl;
219     cout << endl;
220
221     cout <<"[viii] Max()";
222     cout << endl;
223     vector<int> myvecEight = {5, 4, 3, 2, 100};
224     Max(myvecEight);
225     cout << endl;
226     cout << endl;
227
228     cout <<"[ix] Min()";
229     cout << endl;
230     vector<int> myvecNine = {5, 4, 3, 2, 100};
231     Min(myvecNine);
232     cout << endl;
233     cout << endl;
234
235     cout <<"[x] Reverse()";
236     cout << endl;
237     vector<int> myvecTen = {5, 4, 3, 2, 100};
```

```
238     Reverse(myvecTen);
239     cout << endl;
240     cout << endl;
241 
242     cout << "[xi] Shift()";
243     cout << endl;
244     vector<int> myvecEleven = {5, 4, 3, 2, 100};
245     RightShift(myvecEleven,3);
246     cout << endl;
247     cout << endl;
248 
249     cout << "[xii] Rotate()";
250     cout << endl;
251     vector<int> myvecTwelve = {5, 4, 3, 2, 100};
252     Rotate(myvecTwelve,2);
253     cout << endl;
254     cout << endl;
255 }
```


Lab
SEARCH
OPEN EDITORS
lab1.cpp M
lab1two.cpp U
lab1 M
OUTLINE
TIMELINE
LAB
.vscode
lab1 M
lab1.cpp M
lab1two.cpp U
tempCodeRunnerFile.cpp U
SEARCH
Search
Replace
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Code
(base) dk~$cd "/Users/dk/2nd Sem/Data Structures/Lab/" && g++ --std=c++17 lab1.cpp -o lab1 && "/Users/dk/2nd Sem/Data Structures/Lab/"lab1
[i] Display()
1 2 3 4 5 7 9 11 13
[ii] Add/Append(x)
Before Appending 5 4 3 2 1
After Appending 5 4 3 2 1 10
[iii] Insert(index, x)
Before inserting 5 4 3 2 1
After inserting 5 4 3 99 2 1
[iv] Delete(x)
Before deleting 3
5 4 3 2 1
After deleting 3
5 4 2 1
[v] LinearSearch(x)
Found 4 at index 1
[vi] Get(index); function to get value available on the given index
Element 3
[vii] Set(index, x); insert the value x at the given index
Before inserting
5 4 3 2 1
After inserting
5 4 3 30 2 1
[viii] Max()
Array
5 4 3 2 100
Max 100
[ix] Min()
Array
5 4 3 2 100
Min 2
[x] Reverse()
Array before reversing
5 4 3 2 100
Array after reversing
100 2 3 4 5
[xi] Shift()
Array before Right Shift
5 4 3 2 100
Array after Right Shift
main*
Run Testcases
R: (not attached)
Ln 211, Col 18
Spaces: 4
UTF-8
LF
C++
Go Live
Mac
Prettier

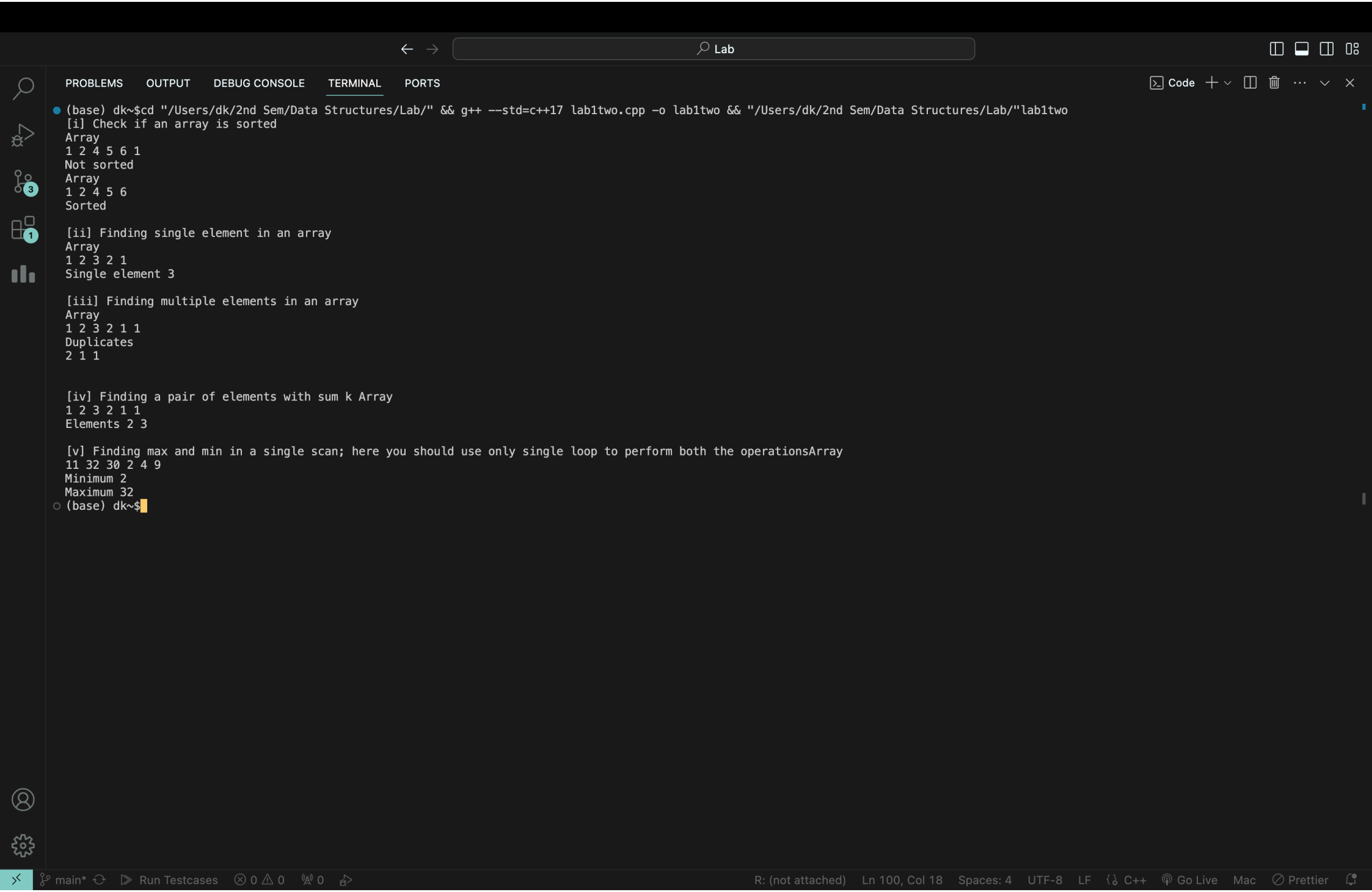

SEARCH
Search
Replace
[xi] Shift()
Array before Right Shift
5 4 3 2 100
Array after Right Shift
0 0 0 5 4
[xii] Rotate()
Array before rotate by 2
5 4 3 2 100
Array After rotate
2 100 5 4 3
(base) dk~$
main*
Run Testcases
R: (not attached)
Ln 211, Col 18
Spaces: 4
UTF-8
LF
C++
Go Live
Mac
Prettier

lab1two.cpp

```cpp
#include<iostream>
#include <unordered_set>
using namespace std;
// --------------------------
void display(vector<int>vec) {
    for (int i = 0; i < vec.size(); i++) {
        cout << vec[i] <<" ";
    }
    cout << endl;
}
void isSorted(vector<int>& vec) {
    cout << "Array ";
    cout << endl;
    display(vec);
    for (int i = 0; i < vec.size() - 1; ++i) {
        if (vec[i] > vec[i + 1]) {
            cout << "Not sorted";
            return;
        }
    }
    cout <<"Sorted ";
}
void findSingleElement(vector<int>& nums) {
    cout << "Array ";
    cout << endl;
    display(nums);
    int result = 0;
    for (int num : nums) {
        result ^= num;
    }
    cout << "Single element " << result;
}
void findMultipleElements (vector<int>&nums) {
    unordered_set<int> seen;
    vector<int> duplicates;
    cout << "Array ";
    cout << endl;
    display(nums);
    for (int num : nums) {
        // If the element is already in the set, it's a duplicate
        if (seen.find(num) != seen.end()) {
            duplicates.push_back(num);
        } else {
            seen.insert(num);
        }
    }
    cout << "Duplicates ";
    cout << endl;
    display(duplicates);
}
void twoSum(vector<int>& nums, int target) {
    cout << "Array ";
    cout << endl;
    display(nums);
    unordered_map<int,int> mpp;
    for(int i = 0; i < nums.size();++i){
        int left = target - nums[i];
```

```
58          auto it = mpp.find(nums[i]);
59          if(it != mpp.end()){
60              cout << "Elements " << nums[it->second] <<" "<< nums[i];
61              return;
62          }
63          else
64          {
65              mpp[left] = i;
66          }
67      }
68      cout << "Elements not found";
69  }
70  void findMinMax(vector<int>&nums){
71      cout << "Array ";
72      cout << endl;
73      display(nums);
74      int min = INT_MAX;
75      int max = INT_MIN;
76      for(auto it : nums){
77          if(it > max)
78              max = it;
79          if(it < min)
80              min = it;
81      }
82      cout << "Minimum " << min <<endl;
83      cout << "Maximum " << max <<endl;
84  }
85  int main()
86  {
87
88      // ------------------------------------------
89      cout << "[i] Check if an array is sorted";
90      cout << endl;
91      vector<int> nums = {1,2,4,5,6,1};
92      isSorted(nums);
93      cout << endl;
94      vector<int> numsOne = {1,2,4,5,6};
95      isSorted(numsOne);
96      cout << endl;
97      cout << endl;
98
99      cout << "[ii] Finding single element in an array";
100     cout << endl;
101     vector<int> numsTwo = {1,2,3,2,1};
102     findSingleElement(numsTwo);
103     cout << endl;
104     cout << endl;
105
106     cout << "[iii] Finding multiple elements in an array";
107     cout << endl;
108     vector<int> numsThree = {1,2,3,2,1,1};
109     findMultipleElements(numsThree);
110     cout << endl;
111     cout << endl;
112
113     cout << "[iv] Finding a pair of elements with sum k ";
114     vector<int> numsFour = {1,2,3,2,1,1};
115     twoSum(numsFour,5);
116     cout << endl;
117     cout << endl;
```

```
118
119     cout << "[v] Finding max and min in a single scan; here you should use only
    single loop to perform both the operations";
120     vector<int> numsFive = {11,32,30,2,4,9};
121     findMinMax(numsFive);
122 }
```


Lab
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Code
(base) dk~$cd "/Users/dk/2nd Sem/Data Structures/Lab/" && g++ --std=c++17 lab1two.cpp -o lab1two && "/Users/dk/2nd Sem/Data Structures/Lab/"lab1two
[i] Check if an array is sorted
Array
1 2 4 5 6 1
Not sorted
Array
1 2 4 5 6
Sorted
[ii] Finding single element in an array
Array
1 2 3 2 1
Single element 3
[iii] Finding multiple elements in an array
Array
1 2 3 2 1 1
Duplicates
2 1 1
[iv] Finding a pair of elements with sum k Array
1 2 3 2 1 1
Elements 2 3
[v] Finding max and min in a single scan; here you should use only single loop to perform both the operationsArray
11 32 30 2 4 9
Minimum 2
Maximum 32
(base) dk~$
main*
Run Testcases
R: (not attached)
Ln 100, Col 18
Spaces: 4
UTF-8
LF
C++
Go Live
Mac
Prettier