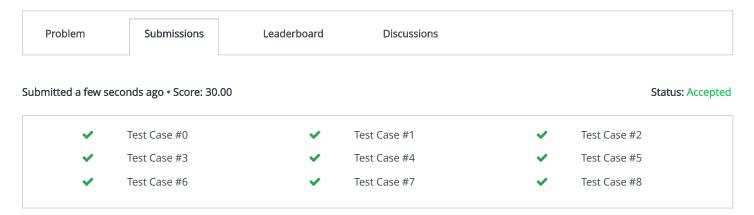
All Contests > In21-CS2023-Lab2 > Arrays - DS

Arrays - DS



Submitted Code

```
P Open in editor
  nguage: C++
 1 #include <bits/stdc++.h>
 3 using namespace std;
 5 string ltrim(const string &);
 6 string rtrim(const string &);
 7 vector<string> split(const string &);
8
9 /*
10 * Complete the 'reverseArray' function below.
11 *
12 * The function is expected to return an INTEGER_ARRAY.
13 * The function accepts INTEGER_ARRAY a as parameter.
14
15
16 vector<int> reverseArray(vector<int> a) {
17
       int arr_length = a.size(); // getting the size of array a
       vector<int> reverse_arr(arr_length); // initializing the reverse_arr vector<int>
18
19
20
       // looping through array a
21
       for (int i=0; i<arr_length; i++){</pre>
           reverse_arr[i] = a[arr_length-i-1]; // assigning the last element for first element i
22
   reverse_arr
23
24
       return reverse_arr;
25 }
26
27 int main()
28 {
29
       ofstream fout(getenv("OUTPUT_PATH"));
30
       string arr_count_temp;
31
32
       getline(cin, arr_count_temp);
```

```
33
       int arr_count = stoi(ltrim(rtrim(arr_count_temp)));
34
35
36
       string arr_temp_temp;
       getline(cin, arr_temp_temp);
37
38
39
       vector<string> arr_temp = split(rtrim(arr_temp_temp));
40
41
       vector<int> arr(arr_count);
42
       for (int i = 0; i < arr_count; i++) {</pre>
43
           int arr_item = stoi(arr_temp[i]);
44
45
           arr[i] = arr_item;
46
47
       }
48
       vector<int> res = reverseArray(arr);
49
50
51
       for (size_t i = 0; i < res.size(); i++) {
52
           fout << res[i];
53
           if (i != res.size() - 1) {
54
               fout << " ";
55
56
           }
       }
57
58
       fout << "\n";
59
60
       fout.close();
61
62
63
       return 0;
64 }
65
66 string ltrim(const string &str) {
67
       string s(str);
68
69
       s.erase(
70
           s.begin(),
71
           find_if(s.begin(), s.end(), not1(ptr_fun<int, int>(isspace)))
72
       );
73
74
       return s;
75 }
76
77 string rtrim(const string &str) {
78
       string s(str);
79
80
       s.erase(
81
           find_if(s.rbegin(), s.rend(), not1(ptr_fun<int, int>(isspace))).base(),
82
           s.end()
       );
83
84
85
       return s;
86 }
87
88 vector<string> split(const string &str) {
       vector<string> tokens;
89
90
91
       string::size_type start = 0;
92
       string::size_type end = 0;
93
       while ((end = str.find(" ", start)) != string::npos) {
94
95
           tokens.push_back(str.substr(start, end - start));
96
97
           start = end + 1;
       }
98
```

```
tokens.push_back(str.substr(start));
tokens.push_back(str.substr(start));
return tokens;

103 }
104
```

Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy |