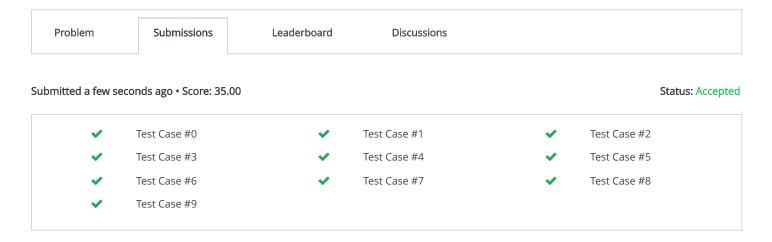
All Contests > In21-CS2023-Lab2 > Left Rotation

Left Rotation



Submitted Code

```
nguage: C++
                                                                                           Open in editor
 1 #include <bits/stdc++.h>
 2
 3 using namespace std;
 4
 5 string ltrim(const string &);
 6 string rtrim(const string &);
 7 vector<string> split(const string &);
8
9 /*
10 * Complete the 'rotateLeft' function below.
11
12
   * The function is expected to return an INTEGER_ARRAY.
   * The function accepts following parameters:
13
   * 1. INTEGER d
14
15 * 2. INTEGER_ARRAY arr
16 */
17
18 vector<int> rotateLeft(int d, vector<int> arr) {
      vector<int> rotatedArr = arr; // initializing rotated array and assigning it to arr
20
       for (int i=0; i<d; i++){ // looping d times</pre>
           int first_elem = rotatedArr[0]; // store the value of first element of arr in a variable
21
           rotatedArr.erase(rotatedArr.begin()); // removing first element from rotatedArr
22
           rotatedArr.push_back(first_elem); // pushing the first element to the back of rotatedArr
23
24
25
      return rotatedArr;
26 }
27
28 int main()
29 {
30
      ofstream fout(getenv("OUTPUT_PATH"));
```

```
32
       string first_multiple_input_temp;
       getline(cin, first_multiple_input_temp);
33
34
       vector<string> first_multiple_input = split(rtrim(first_multiple_input_temp));
35
36
37
       int n = stoi(first_multiple_input[0]);
38
       int d = stoi(first_multiple_input[1]);
39
40
       string arr_temp_temp;
41
       getline(cin, arr_temp_temp);
42
43
44
       vector<string> arr_temp = split(rtrim(arr_temp_temp));
45
       vector<int> arr(n);
46
47
       for (int i = 0; i < n; i++) {
48
           int arr_item = stoi(arr_temp[i]);
49
50
51
           arr[i] = arr_item;
       }
52
53
54
       vector<int> result = rotateLeft(d, arr);
55
       for (size_t i = 0; i < result.size(); i++) {</pre>
56
           fout << result[i];</pre>
57
58
59
           if (i != result.size() - 1) {
               fout << " ";
60
61
62
       }
63
64
       fout << "\n";
65
66
       fout.close();
67
68
       return 0;
69 }
70
71 string ltrim(const string &str) {
       string s(str);
72
73
74
       s.erase(
75
           s.begin(),
           find_if(s.begin(), s.end(), not1(ptr_fun<int, int>(isspace)))
76
77
       );
78
79
       return s;
80 }
81
82 string rtrim(const string &str) {
       string s(str);
83
84
85
       s.erase(
           find_if(s.rbegin(), s.rend(), not1(ptr_fun<int, int>(isspace))).base(),
86
87
       );
88
89
90
       return s;
91 }
92
93 vector<string> split(const string &str) {
94
       vector<string> tokens;
95
96
       string::size_type start = 0;
97
       string::size_type end = 0;
```

```
98
        while ((end = str.find(" ", start)) != string::npos) {
99
100
            tokens.push_back(str.substr(start, end - start));
101
            start = end + 1;
102
        }
103
104
        tokens.push_back(str.substr(start));
105
106
        return tokens;
107
108 }
109
```

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