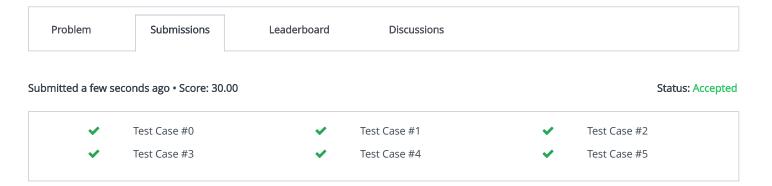
All Contests > In21-CS2023-Lab3 > Insertion Sort - Part 2

## **Insertion Sort - Part 2**



## **Submitted Code**

```
Language: C++
                                                                                            P Open in editor
1 #include <bits/stdc++.h>
 2
3 using namespace std;
 5 string ltrim(const string &);
 6 string rtrim(const string &);
 7 vector<string> split(const string &);
 9 /*
10 * Complete the 'insertionSort2' function below.
11
12 * The function accepts following parameters:
   * 1. INTEGER n
13
14
   * 2. INTEGER_ARRAY arr
15 */
16
17 void insertionSort2(int n, vector<int> arr) {
       // looping from right side
18
       for (int i=1; i<n; i++){
19
20
           int key = arr[i]; // assigining key value
21
22
           while(x>=0 && arr[x]>key){ // checking whether the left element is larger than key
23
               arr[x+1] = arr[x]; // shifting elements
24
25
26
27
           arr[x+1] = key; // placing key value at correct place
28
           // printing array for each iteration
29
           for (int j=0; j<n; j++){
30
               cout << arr[j] << " ";
31
32
33
           cout << endl;</pre>
34
       }
35 }
```

```
37 int main()
38 {
       string n_temp;
39
       getline(cin, n_temp);
40
41
42
       int n = stoi(ltrim(rtrim(n_temp)));
43
       string arr_temp_temp;
44
       getline(cin, arr_temp_temp);
45
46
       vector<string> arr_temp = split(rtrim(arr_temp_temp));
47
48
       vector<int> arr(n);
49
50
       for (int i = 0; i < n; i++) {
51
            int arr_item = stoi(arr_temp[i]);
52
53
54
            arr[i] = arr_item;
55
       }
56
57
       insertionSort2(n, arr);
58
59
        return 0;
60 }
61
62 string ltrim(const string &str) {
       string s(str);
63
64
65
       s.erase(
66
            s.begin(),
67
            find_if(s.begin(), s.end(), not1(ptr_fun<int, int>(isspace)))
68
       );
69
70
       return s;
71 }
72
73 string rtrim(const string &str) {
74
       string s(str);
75
76
       s.erase(
            find_if(s.rbegin(), s.rend(), not1(ptr_fun<int, int>(isspace))).base(),
77
78
            s.end()
79
       );
80
81
       return s;
82 }
83
84 vector<string> split(const string &str) {
85
       vector<string> tokens;
86
87
       string::size_type start = 0;
       string::size_type end = 0;
88
89
       while ((end = str.find(" ", start)) != string::npos) {
90
91
            tokens.push_back(str.substr(start, end - start));
92
            start = end + 1;
93
94
       }
95
       tokens.push_back(str.substr(start));
96
97
98
       return tokens;
99 }
100
```

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