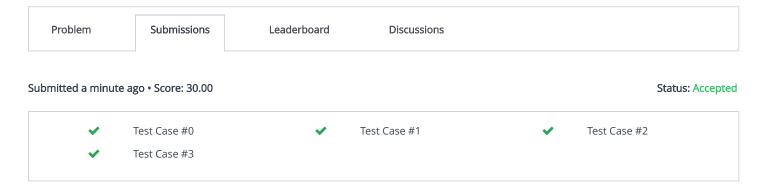
All Contests > In21-CS2023-Lab3 > Insertion Sort - Part 1

Insertion Sort - Part 1



Submitted Code

```
nguage: 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 'insertionSort1' function below.
11
12 * The function accepts following parameters:
   * 1. INTEGER n
13
14
   * 2. INTEGER_ARRAY arr
15 */
16
17 void insertionSort1(int n, vector<int> arr) {
       int key = arr[n-1]; // initializing right most number as key
18
19
       int x = n-2;
       while(x >= 0 && arr[x]>key) { // check whether left elements are larger than key value
20
           arr[x+1]= arr[x]; // shifting elements
21
22
           // printing array after each shifting
23
           for (int j=0; j<n; j++){
24
               cout << arr[j] << " ";</pre>
25
26
27
           cout << endl;</pre>
28
29
           x--;
       }
30
       arr[x+1] = key; // placing key value at its correct position
31
32
33
       // printing array after correct placing
       for (int j=0; j<n; j++){
34
35
           cout << arr[j] << " ";</pre>
```

```
36
        cout << endl;</pre>
37
38 }
39
 40 int main()
41 {
42
        string n_temp;
 43
        getline(cin, n_temp);
 44
        int n = stoi(ltrim(rtrim(n_temp)));
 45
 46
 47
        string arr_temp_temp;
        getline(cin, arr_temp_temp);
 48
49
 50
        vector<string> arr_temp = split(rtrim(arr_temp_temp));
 51
        vector<int> arr(n);
 52
 53
 54
        for (int i = 0; i < n; i++) {
 55
            int arr_item = stoi(arr_temp[i]);
 56
 57
            arr[i] = arr_item;
        }
 58
59
        insertionSort1(n, arr);
 60
61
62
        return 0;
63 }
64
 65 string ltrim(const string &str) {
 66
        string s(str);
67
68
        s.erase(
 69
            s.begin(),
            find_if(s.begin(), s.end(), not1(ptr_fun<int, int>(isspace)))
 70
 71
        );
72
73
        return s;
74 }
75
76 string rtrim(const string &str) {
        string s(str);
77
78
 79
            find_if(s.rbegin(), s.rend(), not1(ptr_fun<int, int>(isspace))).base(),
 80
            s.end()
 81
 82
        );
 83
 84
        return s;
85 }
86
 87 vector<string> split(const string &str) {
        vector<string> tokens;
88
89
        string::size_type start = 0;
90
 91
        string::size_type end = 0;
 92
        while ((end = str.find(" ", start)) != string::npos) {
93
 94
            tokens.push_back(str.substr(start, end - start));
 95
96
            start = end + 1;
97
        }
98
99
        tokens.push_back(str.substr(start));
100
        return tokens;
101
```

102 } 103

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