Insertion Sort - Part 2

Problem | Submissions | Leaderboard | Discussions

Submitted a few seconds ago • Score: 30.00 Status: Accepted

✔ Test Case #0	✔ Test Case #1	✔ Test Case #2
✔ Test Case #3	✔ Test Case #4	✔ Test Case #5

Submitted Code

Language: C++ Open in editor

```cpp
#include <bits/stdc++.h>

using namespace std;

string ltrim(const string &);
string rtrim(const string &);
vector<string> split(const string &);

/*
 * Complete the 'insertionSort2' function below.
 *
 * The function accepts following parameters:
 *  1. INTEGER n
 *  2. INTEGER_ARRAY arr
 */

void insertionSort2(int n, vector<int> arr) {
    // looping from right side
    for (int i=1; i<n; i++){
        int key = arr[i]; // assigining key value
        int x = i-1;

        while(x>=0 && arr[x]>key){ // checking whether the left element is larger than key
            arr[x+1] = arr[x]; // shifting elements
            x--;
        }
        arr[x+1] = key; // placing key value at correct place

        // printing array for each iteration
        for (int j=0; j<n; j++){
            cout << arr[j] << " ";
        }
        cout << endl;
    }
}
```

```
36
37 int main()
38 {
39     string n_temp;
40     getline(cin, n_temp);
41
42     int n = stoi(ltrim(rtrim(n_temp)));
43
44     string arr_temp_temp;
45     getline(cin, arr_temp_temp);
46
47     vector<string> arr_temp = split(rtrim(arr_temp_temp));
48
49     vector<int> arr(n);
50
51     for (int i = 0; i < n; i++) {
52         int arr_item = stoi(arr_temp[i]);
53
54         arr[i] = arr_item;
55     }
56
57     insertionSort2(n, arr);
58
59     return 0;
60 }
61
62 string ltrim(const string &str) {
63     string s(str);
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(
77         find_if(s.rbegin(), s.rend(), not1(ptr_fun<int, int>(isspace))).base(),
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;
88     string::size_type end = 0;
89
90     while ((end = str.find(" ", start)) != string::npos) {
91         tokens.push_back(str.substr(start, end - start));
92
93         start = end + 1;
94     }
95
96     tokens.push_back(str.substr(start));
97
98     return tokens;
99 }
100
```