All Contests > In21-CS2023-Lab5 > Closest Numbers

Closest Numbers

locked

Problem	Submissions	Leaderboard	Discussions			
Submitted 21 min	utes ago • Score: 25.00					Status: Accepted
~	Test Case #0	~	Test Case #1	~	Test Case #2	
~	Test Case #3	~	Test Case #4	~	Test Case #5	

Submitted Code

```
nguage: C++
                                                                                              P Open in editor
 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 &);
9 /*
10 * Complete the 'closestNumbers' function below.
11
12 * The function is expected to return an INTEGER_ARRAY.
   * The function accepts INTEGER_ARRAY arr as parameter.
14
   */
15
16 vector<int> closestNumbers(vector<int> arr) {
       //initializing vector
17
       vector<int> closestArr;
18
       \ensuremath{//} doing insertion sort for the vector
19
       for (int i = 1; i < (int)arr.size(); i++)</pre>
20
21
           int key = arr[i];
22
           int j = i - 1;
23
           while (j \ge 0 \& arr[j] > key)
24
25
               arr[j + 1] = arr[j];
26
27
               j--;
28
           arr[j + 1] = key;
29
30
       // initialzing min with the difference of first two variables
31
32
       int min = arr[1]-arr[0];
33
       // itaerating through the elements of the vector
34
       for (int i=0; i<(int)arr.size()-1; i++){
           // checking whether the new difference is smaller than min
```

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

```
102
103
        s.erase(
104
            find_if(s.rbegin(), s.rend(), not1(ptr_fun<int, int>(isspace))).base(),
105
            s.end()
106
        );
107
108
        return s;
109 }
110
111 vector<string> split(const string &str) {
        vector<string> tokens;
112
113
        string::size_type start = 0;
114
115
        string::size_type end = 0;
116
        while ((end = str.find(" ", start)) != string::npos) {
117
            tokens.push_back(str.substr(start, end - start));
118
119
120
            start = end + 1;
        }
121
122
        tokens.push_back(str.substr(start));
123
124
125
        return tokens;
126 }
127
```

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