

PREPARE^{NEW}

CERTIFY

COMPETE

Search



210554M_CSE_21 ▾

[All Contests](#) > [In21-CS2023-Lab5](#) > [Closest Numbers](#)

Closest Numbers

locked

Problem

Submissions

Leaderboard

Discussions

Sorting is useful as the first step in many different tasks. The most common task is to make finding things easier, but there are other uses as well. In this case, it will make it easier to determine which pair or pairs of elements have the smallest absolute difference between them.

Example

 $arr = [5, 2, 3, 4, 1]$

Sorted, **$arr' = [1, 2, 3, 4, 5]$** . Several pairs have the minimum difference of 1: **$[(1, 2), (2, 3), (3, 4), (4, 5)]$** . Return the array **$[1, 2, 2, 3, 3, 4, 4, 5]$** .

Note

As shown in the example, pairs may overlap.

Given a list of unsorted integers, **arr** , find the pair of elements that have the smallest absolute difference between them. If there are multiple pairs, find them all.

Function Description

Complete the *closestNumbers* function in the editor below.

closestNumbers has the following parameter(s):

- *int arr[n]*: an array of integers

Returns

- *int[]*: an array of integers as described

Input Format

The first line contains a single integer **n** , the length of **arr** .
The second line contains **n** space-separated integers, **$arr[i]$** .

Constraints

- $2 \leq n \leq 200000$
- $-10^7 \leq arr[i] \leq 10^7$
- All $a[i]$ are unique in **arr** .

Output Format

Sample Input 0

```
10
-20 -3916237 -357920 -3620601 7374819 -7330761 30 6246457 -6461594 266854
```

Sample Output 0

```
-20 30
```

Explanation 0

$(30) - (-20) = 50$, which is the smallest difference.

Sample Input 1

```
12
-20 -3916237 -357920 -3620601 7374819 -7330761 30 6246457 -6461594 266854 -520 -470
```

Sample Output 1

```
-520 -470 -20 30
```

Explanation 1

$(-470) - (-520) = 30 - (-20) = 50$, which is the smallest difference.

Sample Input 2

```
4
5 4 3 2
```

Sample Output 2

```
2 3 3 4 4 5
```

Explanation 2

Here, the minimum difference is 1. Valid pairs are (2, 3), (3, 4), and (4, 5).

[f](#) [t](#) [in](#)

Submissions: 196

Max Score: 25

Difficulty: Easy

Rate This Challenge:

☆☆☆☆☆

[More](#)

C++

```

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 'closestNumbers' function below.
11  *
12  * The function is expected to return an INTEGER_ARRAY.
13  * The function accepts INTEGER_ARRAY arr as parameter.
14  */

```

```

15
16 vector<int> closestNumbers(vector<int> arr) {
17
18 }
19
20 int main()
21 {
22     ofstream fout(getenv("OUTPUT_PATH"));
23
24     string n_temp;
25     getline(cin, n_temp);
26
27     int n = stoi(ltrim(rtrim(n_temp)));
28
29     string arr_temp_temp;
30     getline(cin, arr_temp_temp);
31
32     vector<string> arr_temp = split(rtrim(arr_temp_temp));
33
34     vector<int> arr(n);
35
36     for (int i = 0; i < n; i++) {
37         int arr_item = stoi(arr_temp[i]);
38
39         arr[i] = arr_item;
40     }
41
42     vector<int> result = closestNumbers(arr);
43
44     for (size_t i = 0; i < result.size(); i++) {
45         fout << result[i];
46
47         if (i != result.size() - 1) {
48             fout << " ";
49         }
50     }
51
52     fout << "\n";
53
54     fout.close();
55
56     return 0;
57 }
58
59 string ltrim(const string &str) {
60     string s(str);
61
62     s.erase(
63         s.begin(),
64         find_if(s.begin(), s.end(), not1(ptr_fun<int, int>(isspace)))
65     );
66
67     return s;
68 }
69
70 string rtrim(const string &str) {
71     string s(str);
72
73     s.erase(
74         find_if(s.rbegin(), s.rend(), not1(ptr_fun<int, int>(isspace))).base(),
75         s.end()
76     );
77
78     return s;
79 }
80

```

```
81 vector<string> split(const string &str) {  
82     vector<string> tokens;  
83  
84     string::size_type start = 0;  
85     string::size_type end = 0;  
86  
87     while ((end = str.find(" ", start)) != string::npos) {  
88         tokens.push_back(str.substr(start, end - start));  
89  
90         start = end + 1;  
91     }  
92  
93     tokens.push_back(str.substr(start));  
94  
95     return tokens;  
96 }  
97
```

Line: 1 Col: 1

 [Upload Code as File](#) ☐ [Test against custom input](#)

[Run Code](#)[Submit Code](#)

[Interview Prep](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) |