Programming Assignments 1 - Closest Numbers

HackerRank_Username = **s_tassiga**

Problem Statement

Watch Dragon Ball Z Movie ⓪ ✕ | Floating Tasks by Mik Skuza ✕ | GitHub: Where the world bui ✕ | Java Programming Mastercla ✕ | Qt for Python Tutorials — Qt ✕ | Closest Numbers | HackerRa ✕ | +

hackerrank.com/challenges/closest-numbers/problem

Apps M Gmail 30 Best Bootstrap... JavaScript fundame... Create a palette - C... Color Hunt - Color... EJS Eloquent JavaScript... Free React Tutorial... React: Getting Start... Top Free Ethical Ha...

Problem	Submissions	Leaderboard	Discussions	Editorial	Topics

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

Author	HackerRank
Difficulty	Easy
Max Score	35
Submitted By	58780

NEED HELP?

🗎 View discussions

📖 View editorial

🏆 View top submissions

RESOURCES

📖 Greedy Technique

📖 Sorting

📖 Pair

RATE THIS CHALLENGE

☆ ☆ ☆ ☆ ☆

MORE DETAILS

⤓ Download problem statement

⤓ Download sample test cases

✎ Suggest Edits

f y in

Watch Dragon Ball Z Movie ⓪ ✕ | Floating Tasks by Mik Skuza ✕ | GitHub: Where the world bui ✕ | Java Programming Mastercla ✕ | Qt for Python Tutorials — Qt ✕ | Closest Numbers | HackerRa ✕ | +

hackerrank.com/challenges/closest-numbers/problem

Apps M Gmail 30 Best Bootstrap... JavaScript fundame... Create a palette - C... Color Hunt - Color... EJS Eloquent JavaScript... Free React Tutorial... React: Getting Start... Top Free Ethical Ha...

- $2 \leq n \leq 200000$
- $-10^7 \leq arr[i] \leq 10^7$
- 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
```
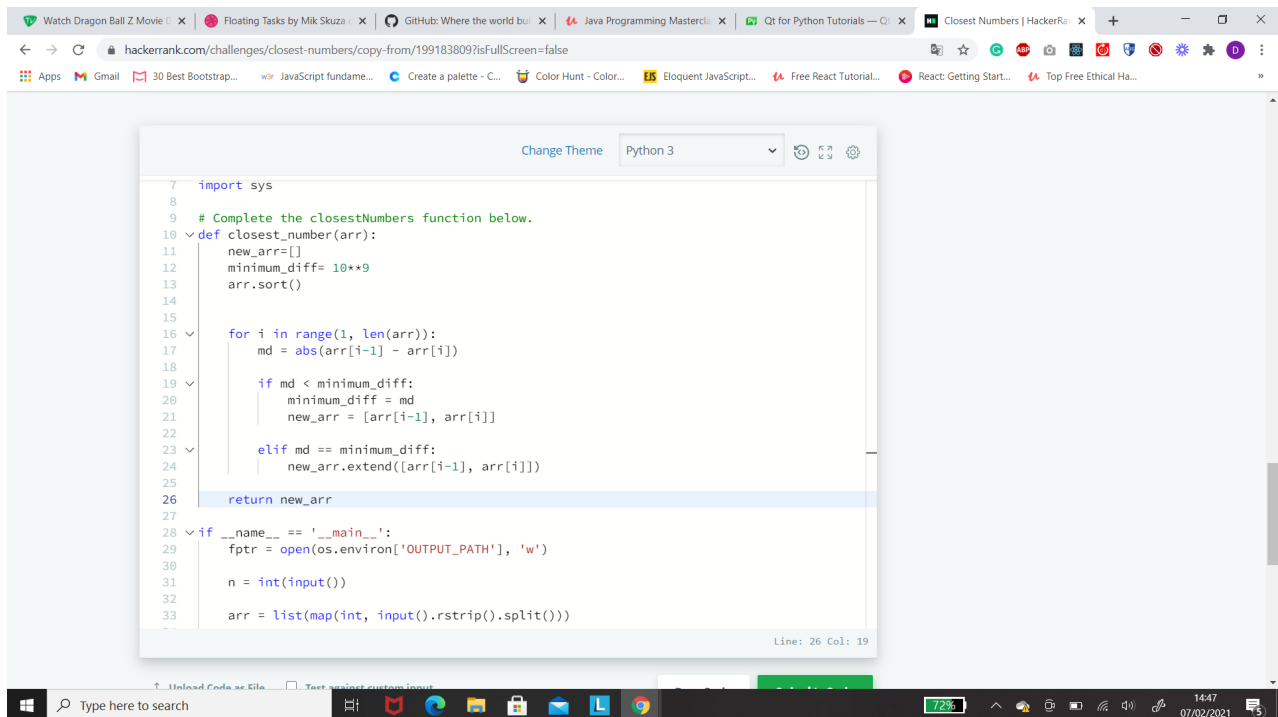
Sample Output 1

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

Explanation 1

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

Code Submitted


```python
import sys

# Complete the closestNumbers function below.
def closest_number(arr):
    new_arr=[]
    minimum_diff= 10**9
    arr.sort()


    for i in range(1, len(arr)):
        md = abs(arr[i-1] - arr[i])

        if md < minimum_diff:
            minimum_diff = md
            new_arr = [arr[i-1], arr[i]]

        elif md == minimum_diff:
            new_arr.extend([arr[i-1], arr[i]])

    return new_arr

if __name__ == '__main__':
    fptr = open(os.environ['OUTPUT_PATH'], 'w')

    n = int(input())

    arr = list(map(int, input().rstrip().split()))
```
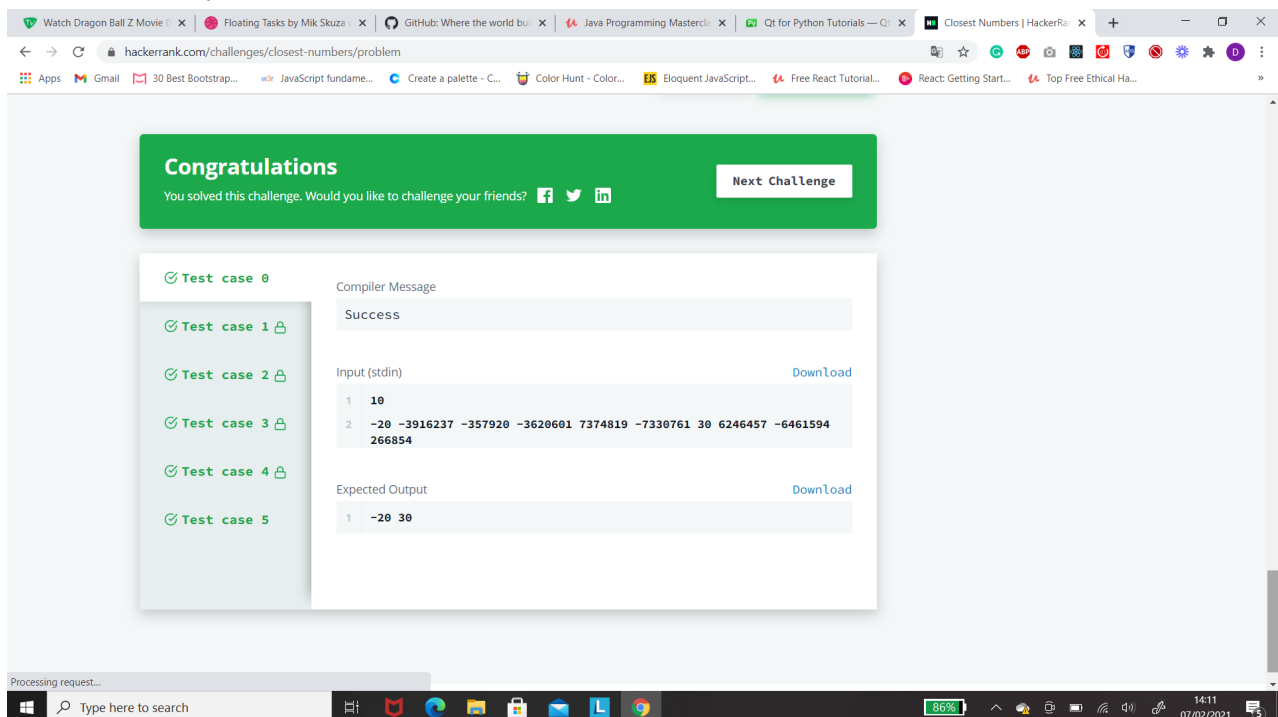
Code Successfully

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

Test case 0
Test case 1
Test case 2
Test case 3
Test case 4
Test case 5

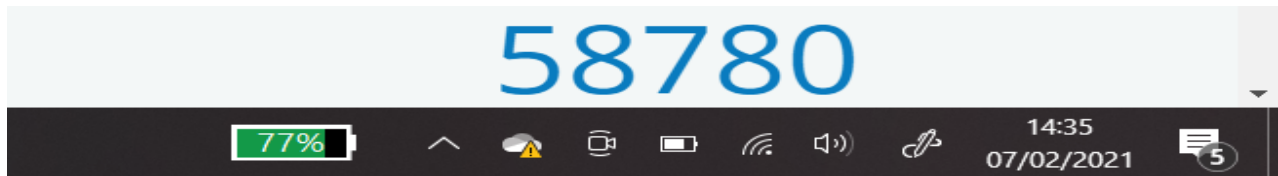Compiler Message

Success

Input (stdin)   Download

```
1   10
2   -20 -3916237 -357920 -3620601 7374819 -7330761 30 6246457 -6461594
    266854
```

Expected Output   Download

```
1   -20 30
```

Programming Language Used: **PYTHON 3**

Time Submitted:

Code User

```python
def closest_number(arr):
    new_arr=[]
    minimum_diff= 10**9
    arr.sort()


    for i in range(1, len(arr)):
        md = abs(arr[i-1] - arr[i])

        if md < minimum_diff:
            minimum_diff = md
            new_arr = [arr[i-1], arr[i]]

        elif md == minimum_diff:
            new_arr.extend([arr[i-1], arr[i]])

    return new_arr
```