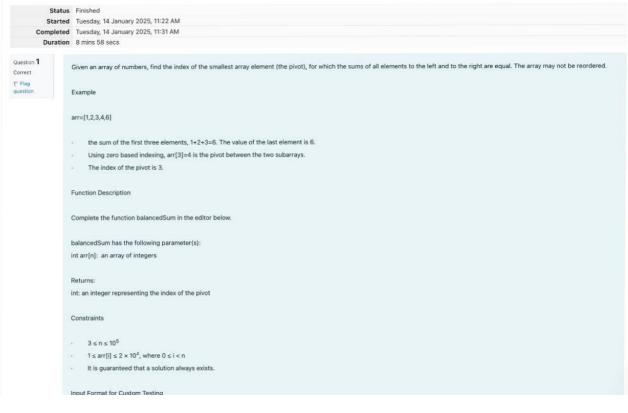
GE23131-Programming Using C-2024





```
Sample Case 0
Sample Input 0
STDIN Function Parameters
4 → arr[] size n = 4
1 → arr = [1, 2, 3, 3]
Sample Output 0
2
Explanation 0
 The sum of the first two elements, 1+2=3. The value of the last element is 3.
Using zero based indexing, \operatorname{arr}[2]=3 is the pivot between the two subarrays. The index of the pivot is 2.
Sample Case 1
Sample Input 1
 STDIN Function Parameters
 3 → arr[] size n = 3
 1 → arr = [1, 2, 1]
Sample Output 1
```

Explanation 1

- The first and last elements are equal to 1.
- Using zero based indexing, arr[1]=2 is the pivot between the two subarrays.
 The index of the pivot is 1.

Answer: (penalty regime: 0 %)

```
Test Expected Got

v int arr[] = {1,2,3,3}; 2 2 v

printf("%d", balancedSum(4, arr))
Passed all tests! 🗸
```

Question 2 Calculate the sum of an array of integers. Y Flag question Example numbers = [3, 13, 4, 11, 9] The sum is 3 + 13 + 4 + 11 + 9 = 40. Function Description Complete the function arraySum in the editor below. arraySum has the following parameter(s): int numbers[n]: an array of integers int: integer sum of the numbers array Constraints $1 \le n \le 10^4$ $1 \leq numbers[i] \leq 10^4$ Input Format for Custom Testing Input from stdin will be processed as follows and passed to the function. The first line contains an integer n, the size of the array numbers. Each of the next n lines contains an integer numbers [i] where $0 \le i < n$. Sample Case 0 Sample Input 0 STDIN Function

```
STDIN Function
5 → numbers[] size n = 5
 1 → numbers = [1, 2, 3, 4, 5]
Sample Output 0
 Explanation 0
 1 + 2 + 3 + 4 + 5 = 15.
 Sample Case 1
 Sample Input 1
 STDIN Function
 2 → numbers[] size n = 2
 12 → numbers = [12, 12]
 12
 Sample Output 1
 Explanation 1
 12 + 12 = 24.
 Answer: (penalty regime: 0 %)
Reset answer
```

Ouestion 3

Oiven an array of n integers, rearrange them so that the sum of the absolute differences of all adjacent elements is minimized. Then, compute the sum of those absolute differences. Example n = 5 arr = [1, 3, 3, 2, 4] if the list is rearranged as arr = [1, 2, 3, 3, 4], the absolute differences are [1 - 2] = 1, [2 - 3] = 1, [3 - 3] = 0, [3 - 4] = 1. The sum of those differences is 1 + 1 + 0 + 1 = 3.

Function Description Complete the function minibil in the editor below, minibil has the following parameter: arr: an integer array Returns: int: the sum of the absolute differences of adjacent elements Constraints 2 s - ns [0 5 c arrigh 1 = 00, s - arrigh 1 = 0, s - arright 1 = 0, s - arrigh 1 = 0, s - arrigh 1 = 0, s - arrigh 1 = 0,

int arr[] = {5, 1, 3, 7, 3}; 6
printf("%d", minDiff(5, arr))

Passed all tests! ✓

Finish review