

REC-CIS

Reset answer

```

1  /*
2  * Complete the 'balancedSum' function below.
3  *
4  * The function is expected to return an INTEGER.
5  * The function accepts INTEGER_ARRAY arr as parameter.
6  */
7
8  int balancedSum(int arr_count, int* arr)
9  {
10     int totalsum=0;
11     for (int i=0;i<arr_count;i++){
12         totalsum+=arr[i];
13     }
14     int leftsum=0;
15     for(int i=0;i<arr_count;i++){
16         int rightsum = totalsum - leftsum - arr[i];
17         if(leftsum==rightsum){
18             return i;
19         }
20         leftsum+=arr[i];
21     }
22     return 1;
23 }
24
25

```

Test	Expected	Got
✓ int arr[] = {1, 2, 2, 1};	2	2 ✓

# REC-CIS

	Test	Expected	Got	
✓	int arr[] = {1,2,3,3}; printf("%d", balancedSum(4, arr))	2	2	✓

Passed all tests! ✓

## Question 2

Correct

Flag question

Calculate the sum of an array of integers.

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):



## REC-CIS

24

Explanation 1

 $12 + 12 = 24.$ **Answer:** (penalty regime: 0 %)

Reset answer

```
1 */  
2 * Complete the 'arraySum' function below.  
3 *  
4 * The function is expected to return an INTEGER.  
5 * The function accepts INTEGER_ARRAY numbers as parameter.  
6 */  
7  
8 int arraySum(int numbers_count, int *numbers)  
9 {  
10     int sum=0;  
11     for(int i=0;i<numbers_count;i++){  
12         sum=sum+numbers[i];  
13     }  
14     return sum;  
15 }  
16
```



# REC-CIS

```

9 | {
10 |     int sum=0;
11 |     for(int i=0;i<numbers_count;i++){
12 |         sum=sum+numbers[i];
13 |     }
14 |     return sum;
15 | }
16 |

```

Test	Expected	Got	
✓ int arr[] = {1,2,3,4,5}; printf("%d", arraySum(5, arr))	15	15	✓

Passed all tests! ✓

Question 3

Correct

Flag question

Given 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 minDiff in the editor below. minDiff has the following parameter: arr: an integer array Returns: int: the sum of the absolute differences of adjacent elements Constraints  $2 \leq n \leq 105$

## REC-CIS

$-5|+3|-5|+3|-7| = 6$ . Sample Case 1 Sample input For Custom testing STDIN Function ----- 2  $\rightarrow$  arr size  $n = 2$  5  
 $\rightarrow$  arr[] = [3, 2] 2 Sample Output 1 Explanation  $n = 2$  arr = [3, 2] There is no need to rearrange because there are only two elements. The final answer is  $|3 - 2| = 1$ .

**Answer:** (penalty regime: 0 %)

Reset answer

```
1 */
2 * Complete the 'minDiff' function below.
3 *
4 * The function is expected to return an INTEGER.
5 * The function accepts INTEGER_ARRAY arr as parameter.
6 */
7
8 #include<stdlib.h>
9 int compare (const void*a,const void*b){
10     return (*(int*)a)-*(int *)b);
11 }
12 int minDiff(int arr_count, int* arr)
13 {
14     qsort(arr,arr_count,sizeof(int),compare);
15     int totaldiff=0;
16     for(int i=1;i<arr_count;i++){
17         totaldiff+=abs(arr[i]-arr[i-1]);
18     }
19     return totaldiff;
20 }
21 }
22 }
```

## REC-CIS

```
7
8 #include<stdlib.h>
9 int compare (const void*a,const void*b){
10     return (*(int*)a-*(int *)b);
11 }
12 int minDiff(int arr_count, int* arr)
13 {
14     qsort(arr,arr_count,sizeof(int),compare);
15     int totaldiff=0;
16     for(int i=1;i<arr_count;i++){
17         totaldiff+=abs(arr[i]-arr[i-1]);
18     }
19     return totaldiff;
20 }
21 }
22
```

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

Passed all tests! ✓

Finish review