

Week-13-Passing Arrays and Strings to Functions

Question 1

Correct

[Flag question](#)

Given an array of numbers, find the index of the smallest array element (the pivot), for which the sums of all elements to the left and to the right are equal. The array may not be reordered.

```
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 left =0,right=0;
11     for(int i=0;i<arr_count;i++)
12     {
13         right+=arr[i];
14     }
15     for(int i=0;i<arr_count;i++)
16     {
17         if(left==(right-arr[i]))
18         {
19             return i;
20         }
21         left+=arr[i];
22         right-=arr[i];
23     }
24     return 1;
25 }
26
```

	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.

```

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     {
13         sum+=numbers[i];
14     }
15     return sum;
16 }
17

```

	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 $2 \leq n \leq 105$ $0 \leq arr[i] \leq 109$, where $0 \leq i < n$ Input Format For Custom Testing The first line of input contains an integer, n , the size of `arr`. Each of the following n lines contains an integer that describes `arr[i]` (where $0 \leq i < n$).

Sample Case 0 Sample Input For Custom Testing STDIN Function ----- 5 \rightarrow `arr[]` size $n = 5$ 5 \rightarrow `arr[]` = [5, 1, 3, 7, 3]
1 3 7 3 Sample Output 6 Explanation $n = 5$ $arr = [5, 1, 3, 7, 3]$ If `arr` is rearranged as $arr' = [1, 3, 3, 5, 7]$, the differences are minimized. The final answer is $|1 - 3| + |3 - 3| + |3 - 5| + |5 - 7| = 6$. Sample Case 1 Sample Input For Custom Testing STDIN Function ----- 2 \rightarrow `arr[]` size $n = 2$ 3 \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 $|3 - 2| = 1$.

```

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  int minDiff(int arr_count, int* arr)
9  {
10     for(int i=0;i<arr_count-1;i++)
11     {
12         for(int j=0;j<arr_count-i-1;j++)
13         {
14             if(arr[j]>arr[j+1])
15             {
16                 int temp = arr[j];
17                 arr[j]=arr[j+1];
18                 arr[j+1]=temp;
19             }
20         }
21     }
22     int sum =0;
23     for(int i=0;i<arr_count-1;i++)
24     {
25         sum+=abs(arr[i]-arr[i+1]);
26     }
27     return sum;
28 }

```

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

Passed all tests! ✓