Sample Output 1

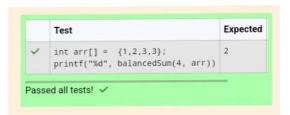
2

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 %)

```
Reset answer
         Complete the 'balancedSum' function be
  3
      * The function is expected to return an
* The function accepts INTEGER_ARRAY arr
 5
  6
     int balancedSum(int arr_count, int* arr)
  8
  9 +
          int totalSum=0,leftSum=0;
 10
          for(int i=0;i<arr_count;i++)
 11
 12
               totalSum+=arr[i];
 13
 14
 15
          for(int i=0;i<arr_count;i++)</pre>
 16
               totalSum-=arr[i];
 17
              if(leftSum==totalSum)
 18
 19
                   return i;
 20
 21
              leftSum+=arr[i];
 22
 23
 24
          return 1;
 25
 26
27
     }
```



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): int numbers[n]: an array of integers

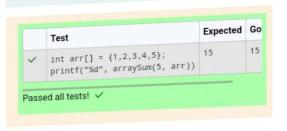
ant number spin, an array of integers

Returns

int: integer sum of the numbers array

REC-CIS

```
IZ → numbers = [IZ, IZ]
12
Sample Output 1
24
Explanation 1
12 + 12 = 24.
Answer: (penalty regime: 0 %)
 Reset answer
        * Complete the 'arraySum' function below
       * The function is expected to return an
   3
       * The function accepts INTEGER_ARRAY num
   4
   6
      int arraySum(int numbers_count, int *numb
   8
           for(int i=0;i<numbers_count;i++)
           int sum=0;
   10
   11
   12
               sum+=numbers[i];
  13
           return sum;
   15
   16
   17
       }
   18
```



Question 3
Correct

P 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 $2 \le n \le 105$ $0 \le arr[i] \le 109$, where $0 \le i < n$ 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 $0 \le i < n$) . Sample Case 0 Sample Input For Custom Testing STDIN --- 5 → arr[] size n = 5 5 → arr[] = [5, 1, 3, 7, 3] Function ---- --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.

Answer: (penalty regime: 0 %)

```
Reset answer
```

```
1 * /*
2     * Complete the 'minDiff' function below.
3     *
4     * The function is expected to return an
5     * The function accepts INTEGER_ARRAY arr
6     */
```

```
*/
6
    int arraySum(int numbers_count, int *numb
9 .
10
        int sum=0;
        for(int i=0;i<numbers_count;i++)</pre>
11
12
13
            sum+=numbers[i]:
15
        return sum;
16
17
18
```

	Test	Expected	Go
~	<pre>int arr[] = {1,2,3,4,5}; printf("%d", arraySum(5, arr))</pre>	15	15
	printf("%d", arraySum(5, arr)) ed all tests! ✓		

Question 3 Correct

P 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 $2 \le n \le 105$ $0 \le arr[i] \le 109$, where $0 \le i < 100$ 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 $0 \le i \le n$) . 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.

Answer: (penalty regime: 0 %)

```
Reset answer
      * Complete the 'minDiff' function below.
     * The function is expected to return an
  4
     * The function accepts INTEGER_ARRAY arr
  6
     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),compa
 15
 16
         int min_sum=0;
         for(int i=1;i<arr_count;i++)
 17
 18
 19
            min_sum+=abs(arr[i]-arr[i-1]);
 20
 21
         return min_sum;
 22
```

```
Test Expected Got

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

Passed all tests! ✓
```