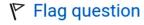
Question 1

Correct



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.

Example

- the sum of the first three elements, 1+2+3=6. The value of the last element is 6.
- Using zero based indexing, arr[3]=4 is the pivot between the two subarrays.

The index of the pivot is 3.

Function Description

Complete the function balancedSum in the editor below.

balancedSum has the following parameter(s):

int arr[n]: an array of integers

Returns:

int: an integer representing the index of the pivot

Constraints

- $3 \le n \le 10^5$
- . $1 \le arr[i] \le 2 \times 10^4$, where $0 \le i$

< n

 It is guaranteed that a solution always exists.

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 arr.

Each of the next n lines contains an integer, arr[i], where $0 \le i < n$.

Sample Case 0

Sample Input 0

STDIN Function Parameters

4 \rightarrow arr[] size n = 4

1 \rightarrow arr = [1 2 3 3]

 \equiv

<

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, 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

 $3 \rightarrow arr[] size n = 3$

1 \rightarrow arr = [1, 2, 1]

2

1

Sample Output 1

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.

recet anower

```
1
        Complete the 'bala
      *
 2
 3
      *
        The function is ex
 4
      *
        The function acce
 5
 6
      */
 7
     int balancedSum(int
 8
 9
     {
     int t=0;
10
     for(int i=0;i<arr_col
11
     t+=arr[i];
12
     int 1=0;
     for(int i=0;i<arr_col
13
          int r=t-l-arr[i]
14
15
          if(1==r){
16 ▼
              return i;
17
18
          1+=arr[i];
19
20
     return 1;
21
     }
22
23
```

Question **2**

Correct



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

Returns

int: integer sum of the numbers array

Constraints

 $1 \le n \le 10^4$

 $1 \le \text{numbers}[i] \le 10^4$

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 ≤ i < n.

Sample Case 0 Sample Input 0

STDIN Function

5 → numbers[] size n = 5

 \rightarrow numbers = [1, 2, 3, 4, 5] 1

2 3

4 5

Sample Output 0

15

Explanation 0

$$1 + 2 + 3 + 4 + 5 = 15$$
.

Sample Case 1

Sample Input 1

2 \rightarrow numbers [] size n = 2

12 \rightarrow numbers = [12, 12]

12

Sample Output 1

24

Explanation 1

12 + 12 = 24.

Reset answer

```
/*
 1
 2
        Complete the 'arra
 3
      *
 4
      * The function is ex
 5
      * The function acce
 6
      */
 7
 8
     int arraySum(int numl
 9
     {
10
        int sum=0;
11
        for(int i=0;i<num/)</pre>
12
        sum=sum+numbers[i]
13
        return sum;
14
    }
15
```

Question ${f 3}$

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 ≤ n ≤105 0 ≤ arr[i] ≤ 109, where 0 ≤ i < n Input Format For Custom Testing Reset answer

```
*
 1
 2
       Complete the 'min
 3
     *
 4
     * The function is ex
 5
     * The function acce
 6
     */
 7
    #include<stdlib.h>
 8
    int compare(const vo:
 9
    return (*(int *)a-*(:
10
    int minDiff(int arr_
11
    {
12
         qsort(arr,arr_col
13
         int t=0;
14
         for(int i=1;i<ar
15
             t+=abs(arr[i
16
17
         return t;
18
19
20
```