WEEK 13

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 below.
 *
 * The function is expected to return an INTEGER.
 * The function accepts INTEGER_ARRAY arr as parameter.
 */

int balancedSum(int arr_count, int* arr)
{
    int ls=0;
    for(int i=0;i<arr_count;i++){
        ls+=arr[i];
        for(int j=arr_count;j>arr_count/2;j--){
            if(ls==arr[j]) return arr_count-(i+1);
        }
    }
    return -1;
}

```

	Test	Expected	Got	
✓	int arr[] = {1,2,3,3};	2	2	✓

Sample Output 1

24

Explanation 1

12 + 12 = 24.

Answer: (penalty regime: 0 %)

Reset answer

```c
/*
 * Complete the 'arraySum' function below.
 *
 * The function is expected to return an INTEGER.
 * The function accepts INTEGER_ARRAY numbers as parameter.
 */

int arraySum(int numbers_count, int *numbers)
{
    int sum=0;
    for(int i=0;i<numbers_count;i++){
        sum+=numbers[i];
    }
    return sum;
}

```

	Test	Expected	Got	
✓	int arr[] = {1 2 3 4 5};	15	15	✓

differences of adjacent elements Constraints 2 ≤ n ≤105 0 ≤ arr[i] ≤ 109, where 0 ≤ 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 ≤ i < n) . Sample Case 0 Sample Input For Custom Testing STDIN Function ----- -------- 5 → arr[] size n = 5 5 → 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 → arr[] size n = 2 3 → 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

```c
/*
 * Complete the 'minDiff' function below.
 *
 * The function is expected to return an INTEGER.
 * The function accepts INTEGER_ARRAY arr as parameter.
 */

int minDiff(int arr_count, int* arr)
{
    int sum=0;
    for(int i=0;i<arr_count;i++){
        for(int j=i+1;j<arr_count;j++){
            if(arr[i]>arr[j]){
                int temp=arr[i];
                arr[i]=arr[j];
                arr[j]=temp;
            }
        }
    }
    for(int i=0;i<arr_count-1;i++){
        if(arr[i]<arr[i+1]) sum+=(arr[i+1]-arr[i]);
        else sum+=(arr[i]-arr[i+1]);
    }
    return sum;
}
```

	Test	Expected	Got	
✓	int arr[] = {5, 1, 3, 7, 3};	6	6	✓