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=sum+numbers[i];
    }
    return sum;
}
```

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

Passed all tests! ✓

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

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

Passed all tests! ✓

Answer: (penalty regime: 0 %)

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

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