

Answer. (penalty regime: 0 %)

Reset answer

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

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

Passed all tests! ✓

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

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

Passed all tests! ✓

Reset answer

```
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! ✓