

```

1  /*
2   * Complete the 'balancedSum' function
3   *
4   * The function is expected to return a boolean.
5   * The function accepts INTEGER_ARRAY arr and INTEGER n as parameters.
6   */
7
8  int balancedSum(int arr_count, int n)
9  {
10     int totalsum = 0;
11     for (int i = 0; i < arr_count; i++)
12         totalsum += arr[i];
13
14     int leftsum = 0;
15     for(int i = 0; i < arr_count; i++)
16     {
17         int rightsum = totalsum - leftsum;
18         if(leftsum == rightsum)
19             return true;
20         leftsum += arr[i];
21     }
22     return false;
23 }
24

```

	Test
✓	<pre>int arr[] = {1,2,3,3}; printf("%d", balancedSum(4, arr));</pre>

Passed all tests! ✓

```

1  /*
2   * Complete the 'arraySum' f
3   *
4   * The function is expected
5   * The function accepts INTE
6   */
7
8  int arraySum(int numbers_cou
9  {
10     int sum =0;
11     for (int i =0;i<numbers_c
12         sum = sum+numbers[i];
13     }
14     return sum;
15 }
16

```

	Test
✓	<pre>int arr[] = {1,2,3,4,5}; printf("%d", arraySum(5, arr))</pre>

Passed all tests! ✓

**Answer:** (penalty regime: 0 %)

Reset answer

```
1  /*
2  * Complete the 'minDiff' fu
3  *
4  * The function is expected
5  * The function accepts INTE
6  */
7  #include <stdlib.h>
8  int compare(const void *a, c
9      return (*(int*)a - *(int
10 }
11 int minDiff(int arr_count, i
12 {
13     qsort(arr, arr_count, siz
14     int totaldiff=0;
15     for(int i =1;i<arr_count
16         totaldiff += abs(arr
17     }
18     return totaldiff;
19 }
20
```

	Test
✓	<pre>int arr[] = {5, 1, 3, 7, 3}; printf("%d", minDiff(5, arr))</pre>

Passed all tests! ✓