

```

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! ✓

```

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
8  int compare(const void* a,const void*b)
9  {
10     return(*(int*)a-*(int*)b);
11 }
12 int minDiff(int arr_count,int* arr)
13 {
14     qsort(arr,arr_count,sizeof(int),compare);
15     int min_sum=0;
16     for(int i=1;i<arr_count;i++)
17     {
18         min_sum+=abs(arr[i]-arr[i-1]);
19     }
20     return min_sum;
21 }
22

```

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

```

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  int balancedSum(int arr_count, int* arr)
8  {
9      int totalSum=0,leftSum=0;
10     for(int i=0;i<arr_count;i++)
11     {
12         totalSum+=arr[i];
13     }
14     for(int i=0;i<arr_count;i++)
15     {
16         totalSum-=arr[i];
17         if(leftSum==totalSum)
18         {
19             return i;
20         }
21         leftSum+=arr[i];
22     }
23     return 1;
24 }
25
26

```

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

Passed all tests! ✓