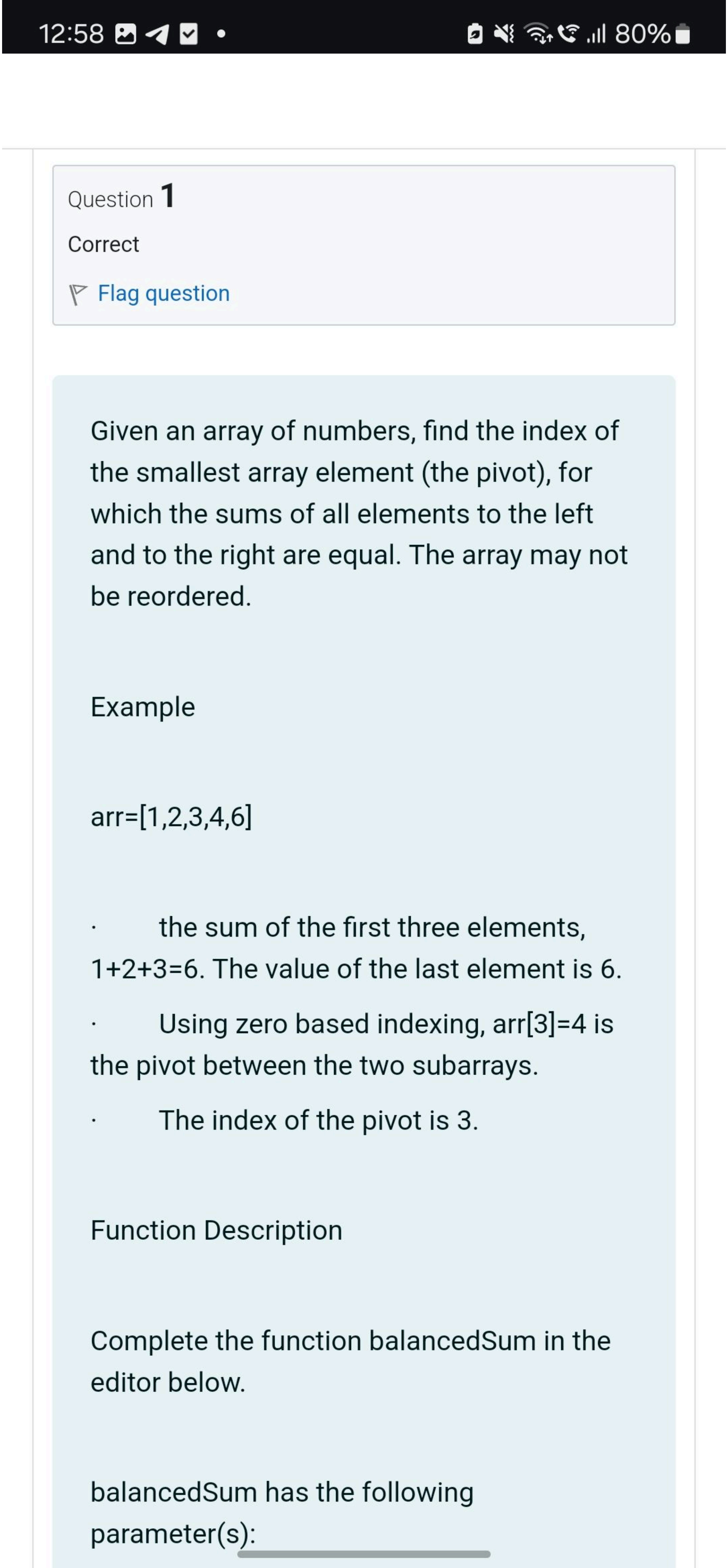


Question 1

Correct

Given an array of numbers, find the index of the smallest array element (the pivot), for which the sums of all elements to the left and to the right are equal. The array may not be reordered.

Example

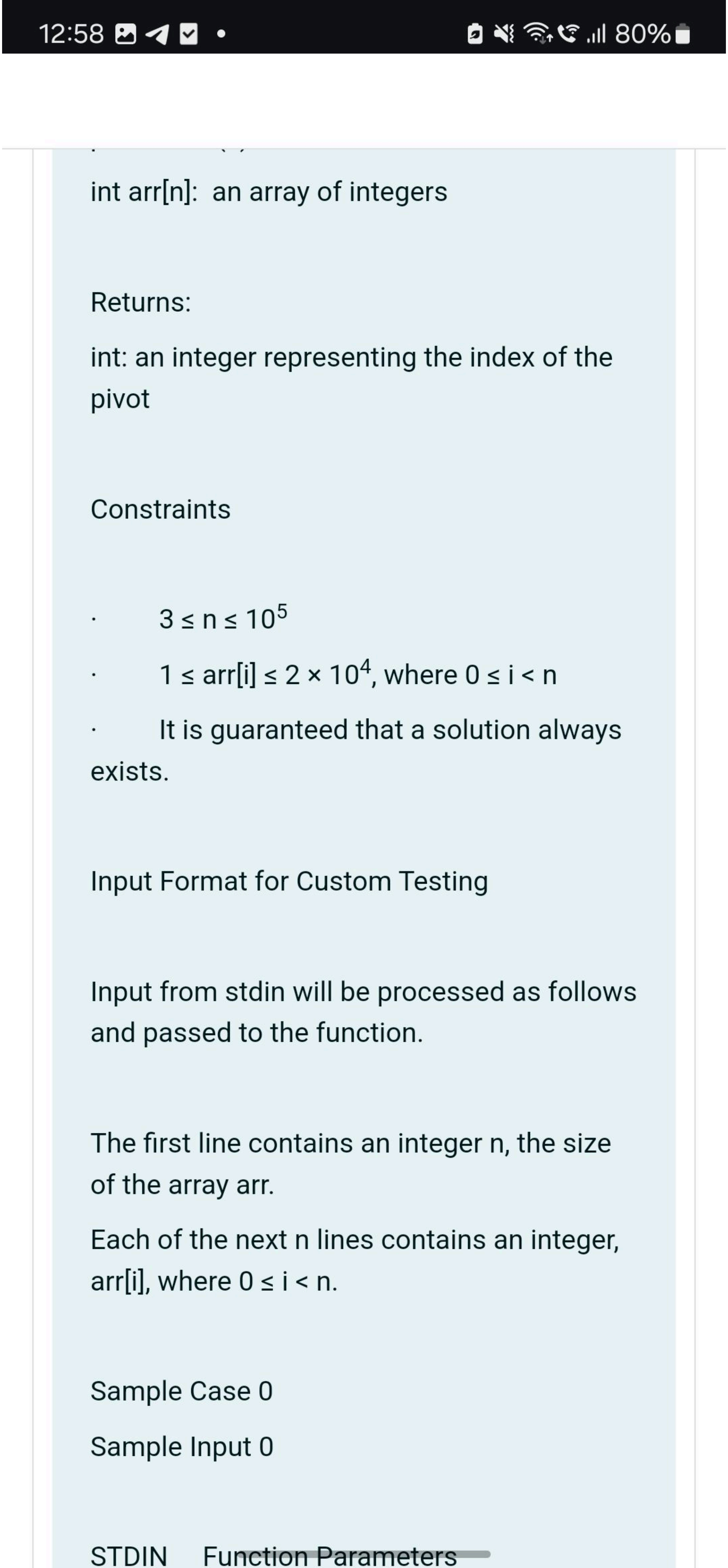
arr=[1,2,3,4,6]

- the sum of the first three elements, 1+2+3=6. The value of the last element is 6.
- Using zero based indexing, arr[3]=4 is the pivot between the two subarrays.
- The index of the pivot is 3.

Function Description

Complete the function balancedSum in the editor below.

balancedSum has the following parameter(s):



int arr[n]: an array of integers

Returns:

int: an integer representing the index of the pivot

Constraints

- $3 \le n \le 10^5$
- $1 \le arr[i] \le 2 \times 10^4$, where $0 \le i < n$
- It is guaranteed that a solution always exists.

Input Format for Custom Testing

Input from stdin will be processed as follows and passed to the function.

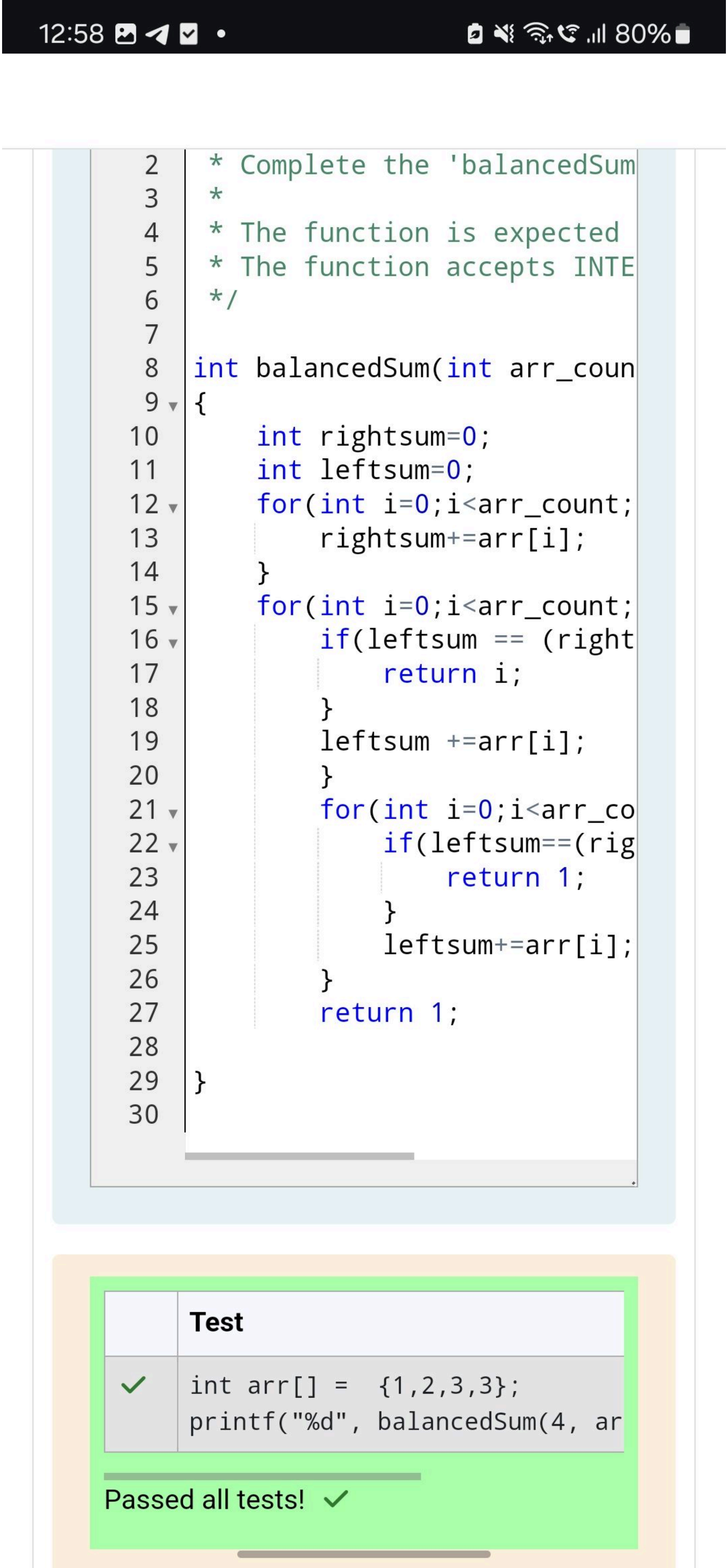
The first line contains an integer n, the size of the array arr.

Each of the next n lines contains an integer, arr[i], where $0 \le i < n$.

Sample Case 0

Sample Input 0

STDIN Function Parameters

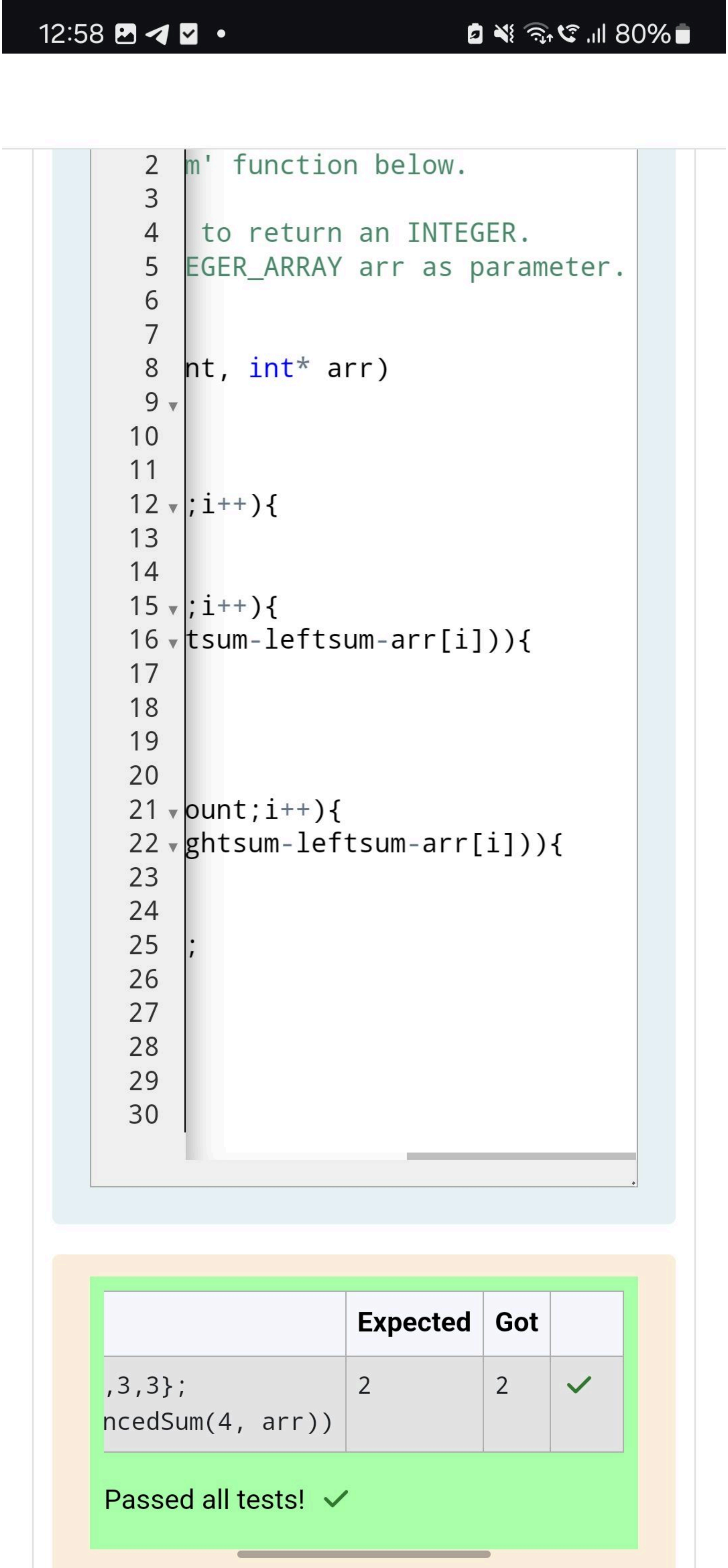


```
 * Complete the 'balancedSum
 *
 * The function is expected
 * The function accepts INTE
 */

int balancedSum(int arr_coun
{
    int rightsum=0;
    int leftsum=0;
    for(int i=0;i<arr_count;
        rightsum+=arr[i];
    }
    for(int i=0;i<arr_count;
        if(leftsum == (right
            return i;
        }
        leftsum +=arr[i];
        }
        for(int i=0;i<arr_co
            if(leftsum==(rig
                return 1;
            }
            leftsum+=arr[i];
        }
        return 1;

}

```

	Test

✓	`int arr[] =  {1,2,3,3};`
	`printf("%d", balancedSum(4, ar`

Passed all tests! ✓

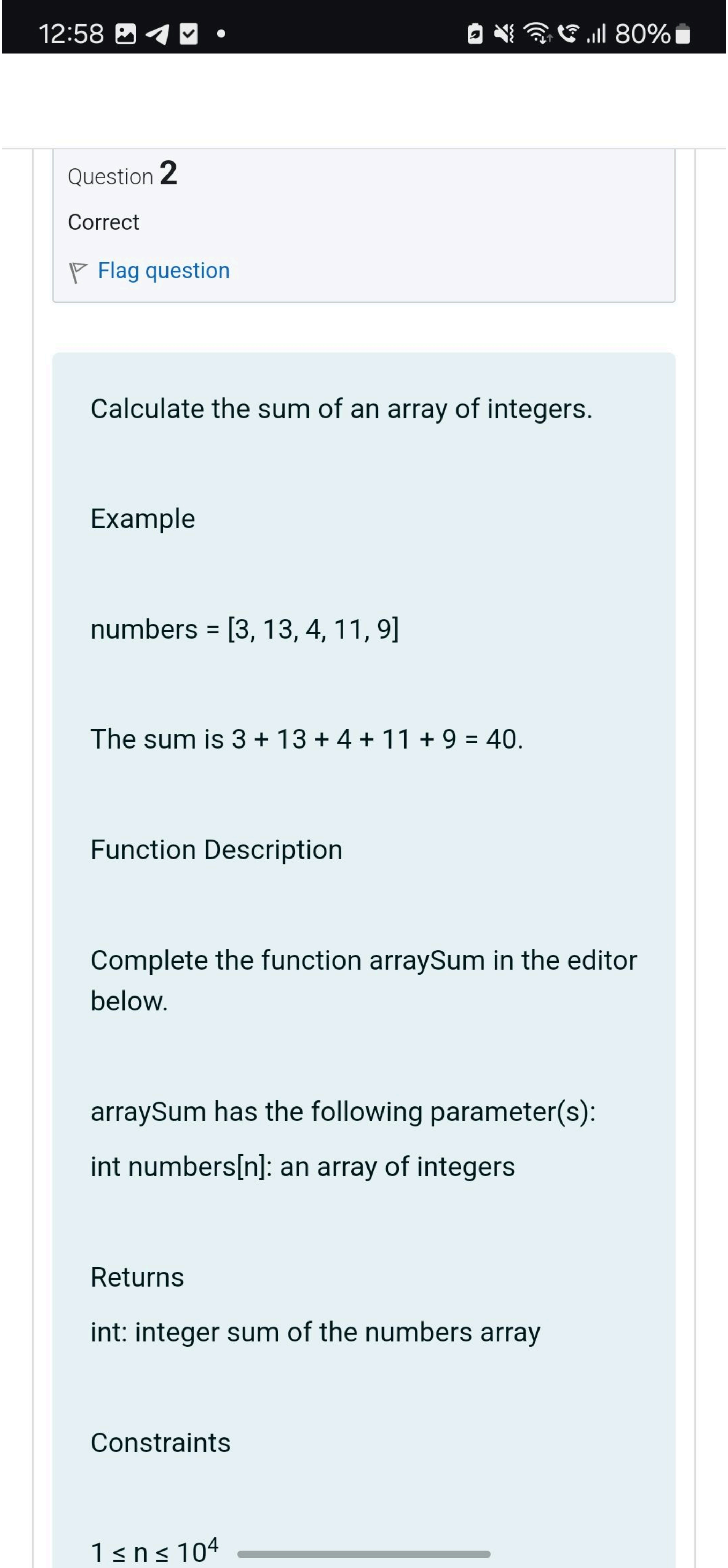


```
m' function below.

 to return an INTEGER.
EGER_ARRAY arr as parameter.


nt, int* arr)



;i++){


;i++){
tsum-leftsum-arr[i])){




ount;i++){
ghtsum-leftsum-arr[i])){


;
```

	Expected	Got	

,3,3};
ncedSum(4, arr))
	2	2	✓

Passed all tests! ✓



Question 2

Correct

Calculate the sum of an array of integers.

Example

numbers = [3, 13, 4, 11, 9]

The sum is 3 + 13 + 4 + 11 + 9 = 40.

Function Description

Complete the function arraySum in the editor below.

arraySum has the following parameter(s):

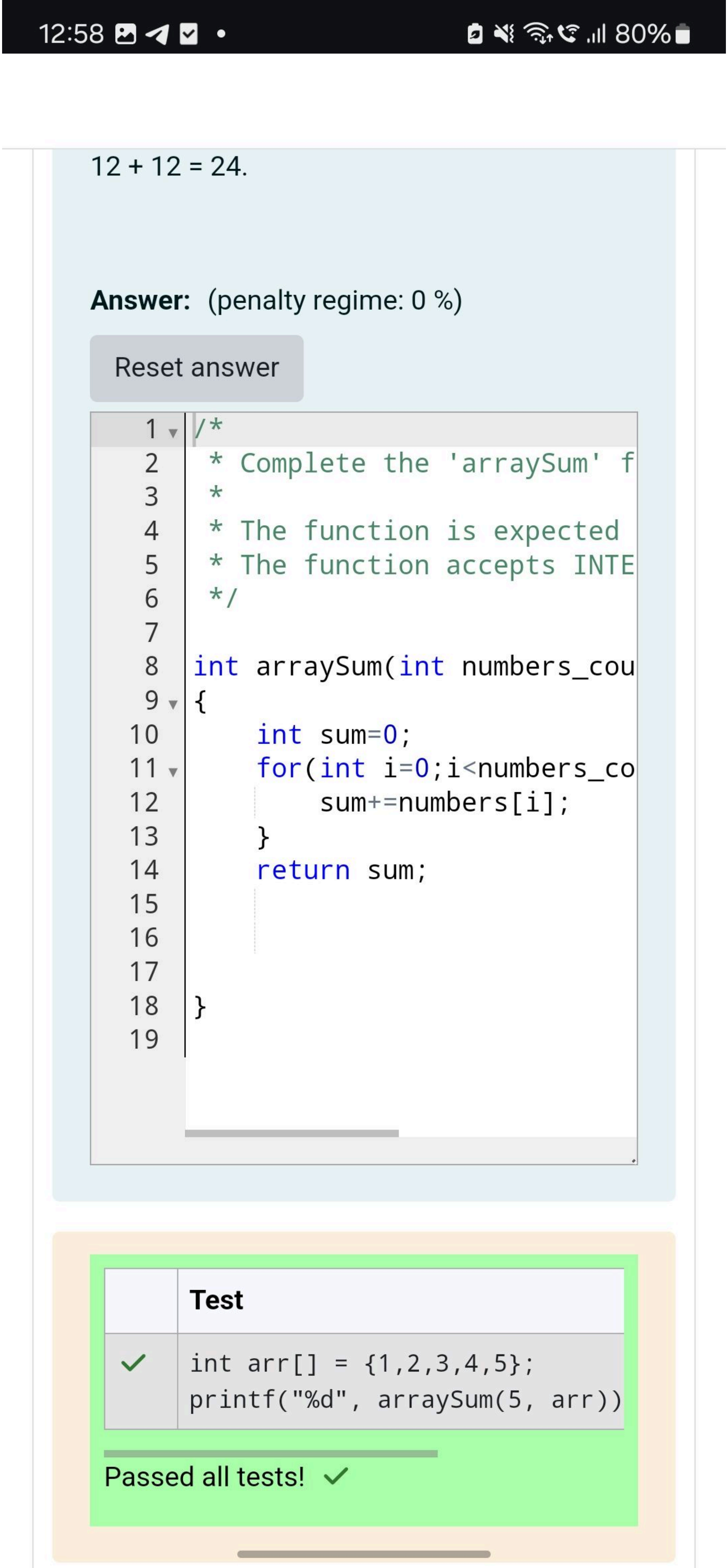
int numbers[n]: an array of integers

Returns

int: integer sum of the numbers array

Constraints

$1 \leq n \leq 10^4$



12 + 12 = 24.

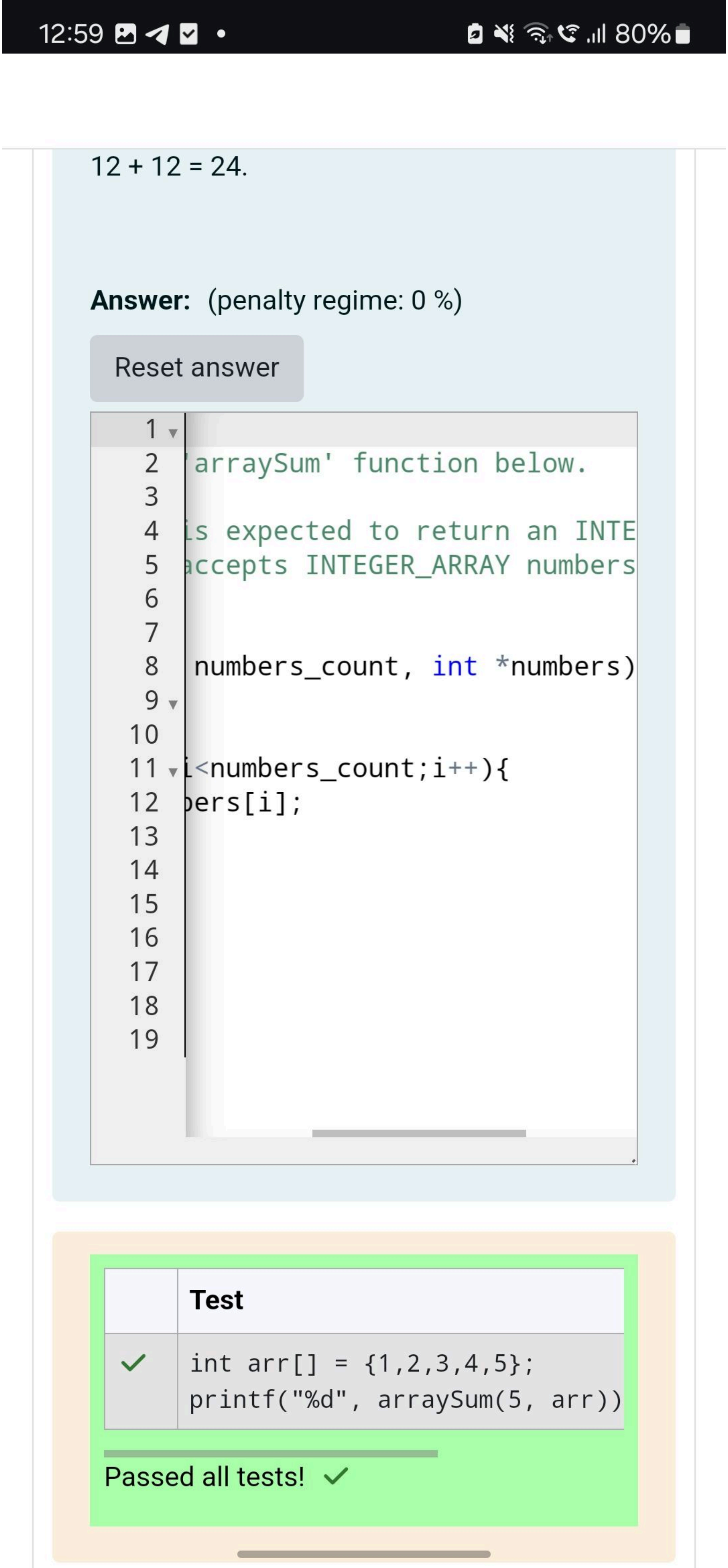
Answer: (penalty regime: 0 %)

```
/*
 * Complete the 'arraySum' f
 *
 * The function is expected
 * The function accepts INTE
 */

int arraySum(int numbers_cou
{
    int sum=0;
    for(int i=0;i<numbers_co
        sum+=numbers[i];
    }
    return sum;



}

```

	Test

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

Passed all tests! ✓



12 + 12 = 24.

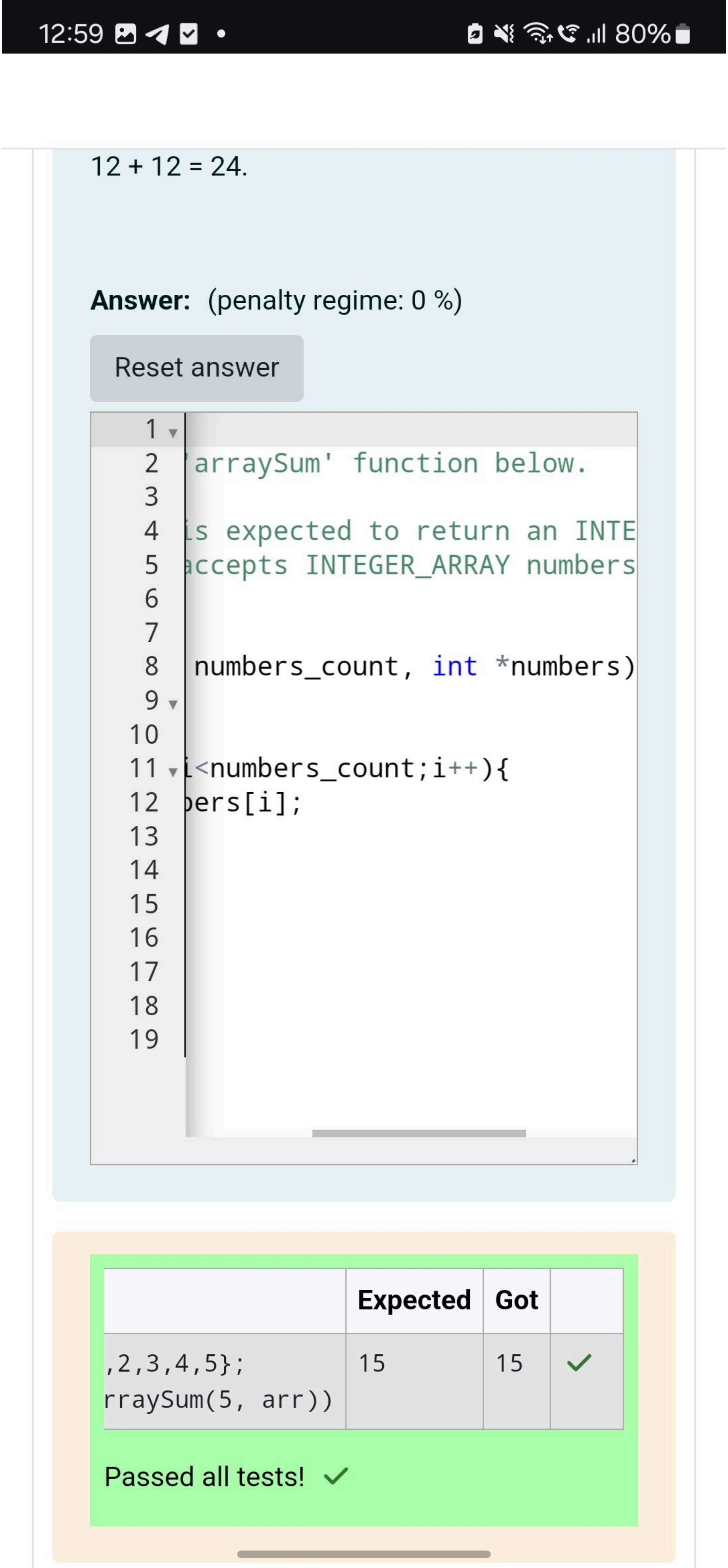
Answer: (penalty regime: 0 %)

```

'arraySum' function below.

is expected to return an INTE
accepts INTEGER_ARRAY numbers


 numbers_count, int *numbers)


i<numbers_count;i++){
pers[i];
```

	Test

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

Passed all tests! ✓



12 + 12 = 24.

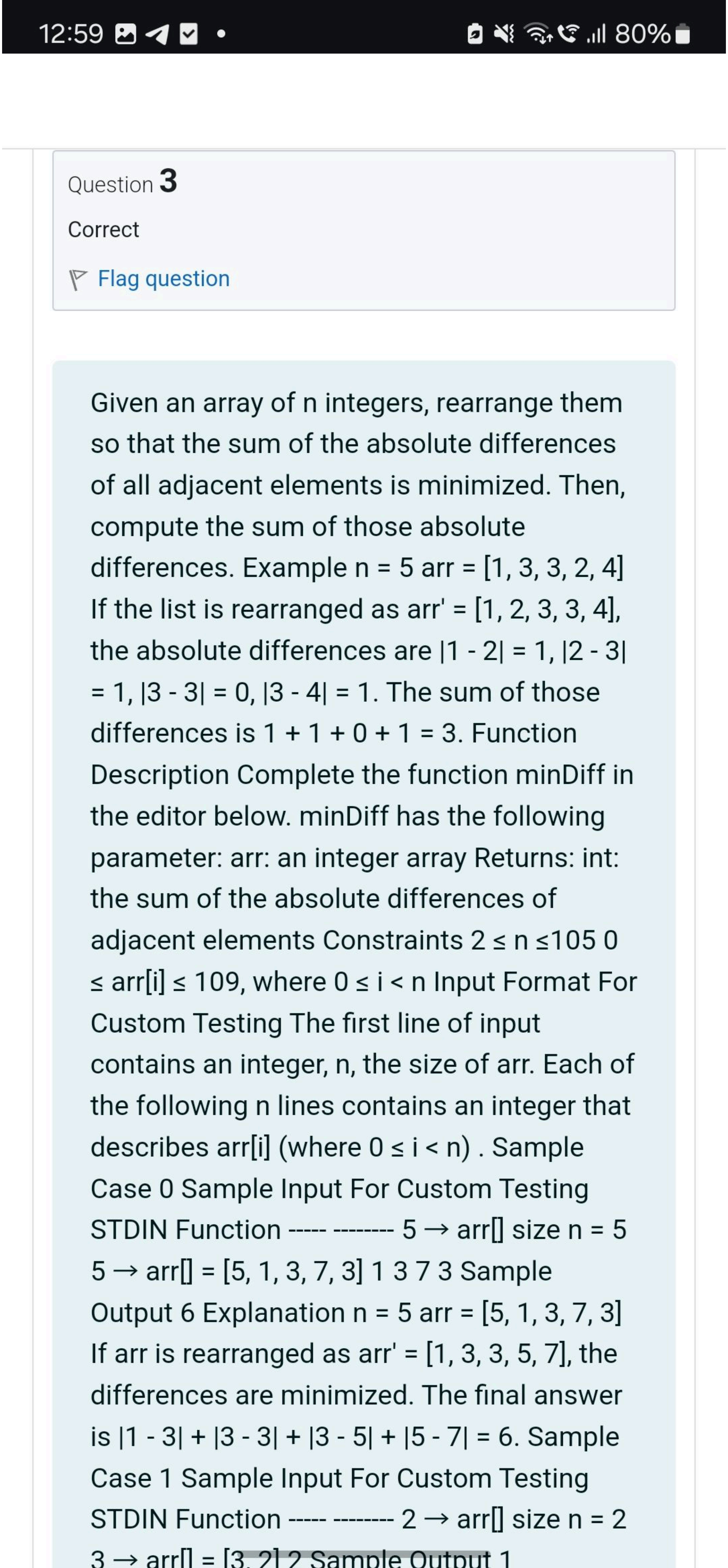
Answer: (penalty regime: 0 %)

```

'arraySum' function below.

is expected to return an INTE
accepts INTEGER_ARRAY numbers


 numbers_count, int *numbers)


i<numbers_count;i++){
bers[i];







```

	Expected	Got	

,2,3,4,5};
rraySum(5, arr))
	15	15	✓

Passed all tests! ✓

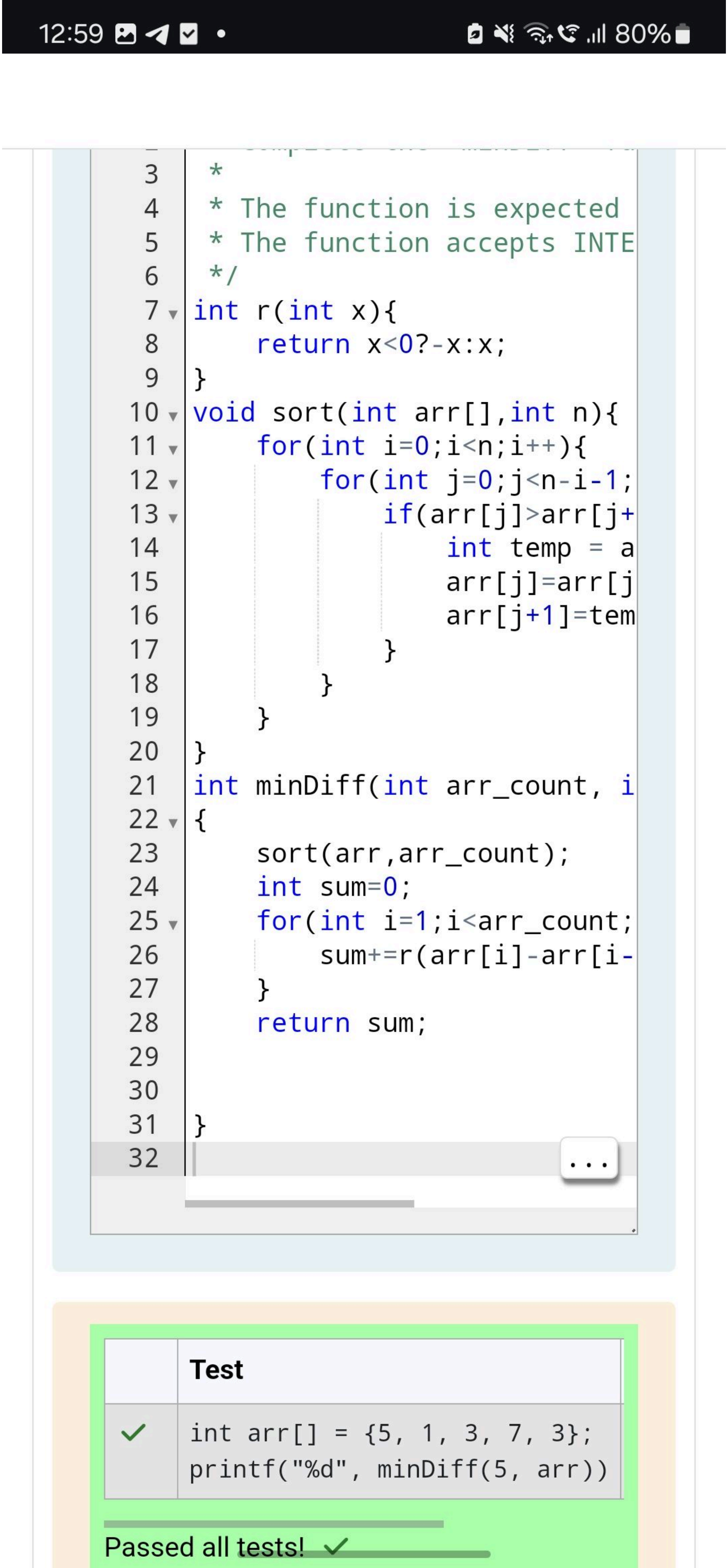


Question 3

Correct

Flag question

Given an array of n integers, rearrange them so that the sum of the absolute differences of all adjacent elements is minimized. Then, compute the sum of those absolute differences. Example n = 5 arr = [1, 3, 3, 2, 4] If the list is rearranged as arr' = [1, 2, 3, 3, 4], the absolute differences are |1 - 2| = 1, |2 - 3| = 1, |3 - 3| = 0, |3 - 4| = 1. The sum of those differences is 1 + 1 + 0 + 1 = 3. Function Description Complete the function minDiff in the editor below. minDiff has the following parameter: arr: an integer array Returns: int: the sum of the absolute 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

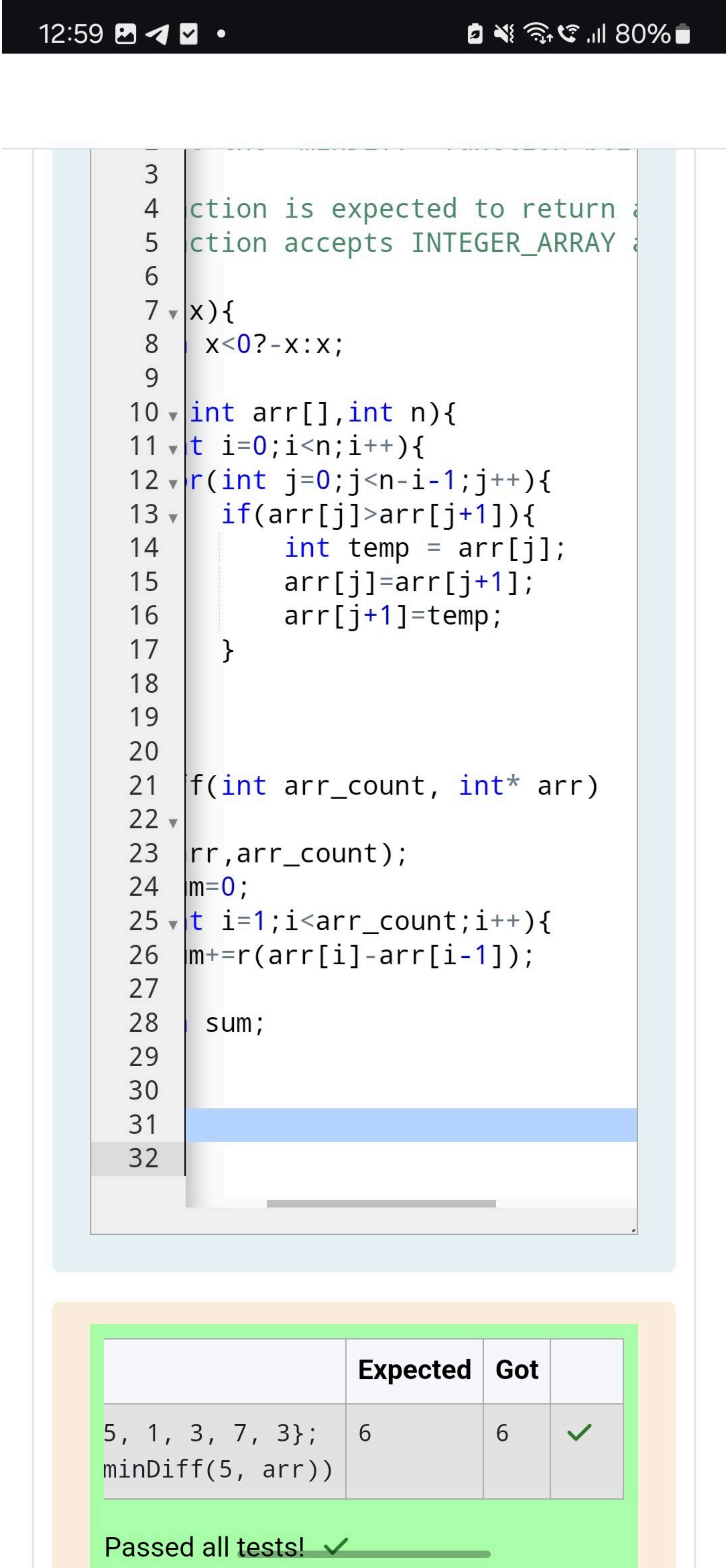


```
 *
 * The function is expected
 * The function accepts INTE
 */
int r(int x){
    return x<0?-x:x;
}
void sort(int arr[],int n){
    for(int i=0;i<n;i++){
        for(int j=0;j<n-i-1;
            if(arr[j]>arr[j+
                int temp = a
                arr[j]=arr[j
                arr[j+1]=tem
            }
        }
    }
}
int minDiff(int arr_count, i
{
    sort(arr,arr_count);
    int sum=0;
    for(int i=1;i<arr_count;
        sum+=r(arr[i]-arr[i-
    }
    return sum;


}

```

	Test

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

Passed all tests! ✓



12:59 80%

```
ction is expected to return a
ction accepts INTEGER_ARRAY a

x){
 x<0?-x:x;

int arr[],int n){
t i=0;i<n;i++){
r(int j=0;j<n-i-1;j++){
  if(arr[j]>arr[j+1]){
      int temp = arr[j];
      arr[j]=arr[j+1];
      arr[j+1]=temp;
  }



f(int arr_count, int* arr)

rr,arr_count);
m=0;
t i=1;i<arr_count;i++){
m+=r(arr[i]-arr[i-1]);

 sum;
```

	Expected	Got	

5, 1, 3, 7, 3};
minDiff(5, arr))
	6	6	✓

Passed all tests! ✓