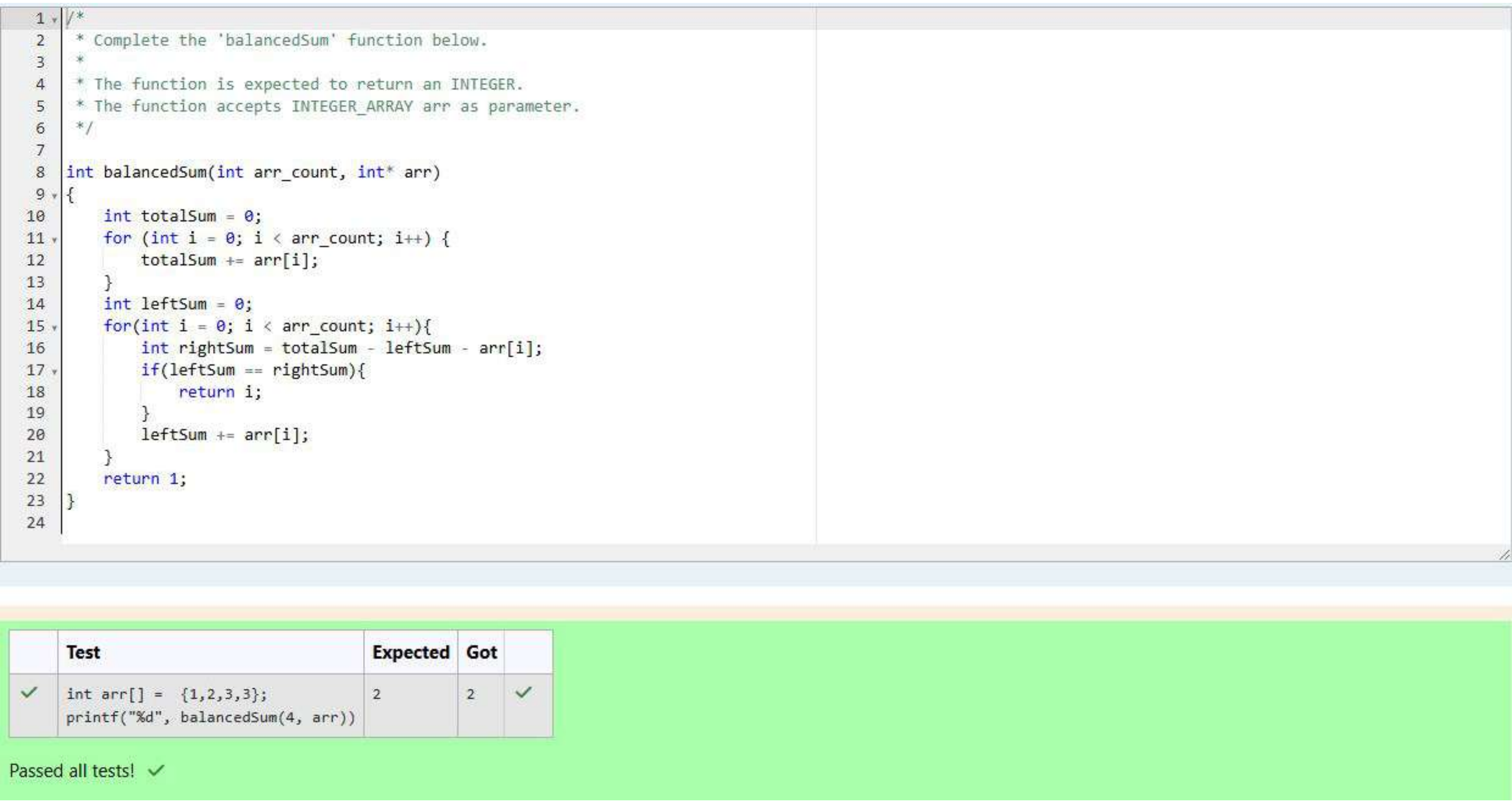
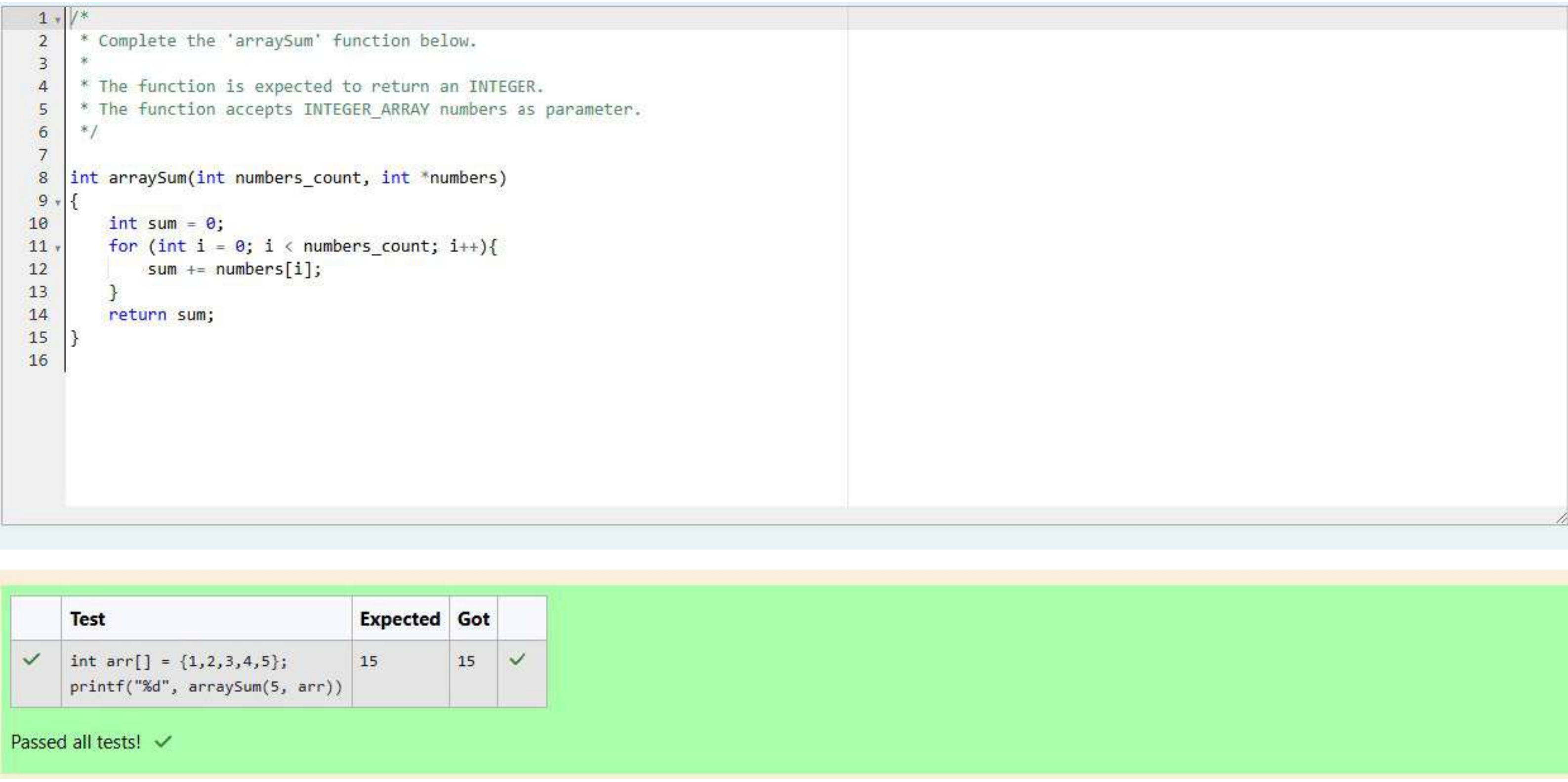


```c
/*
 * Complete the 'balancedSum' function below.
 *
 * The function is expected to return an INTEGER.
 * The function accepts INTEGER_ARRAY arr as parameter.
 */

int balancedSum(int arr_count, int* arr)
{
    int totalSum = 0;
    for (int i = 0; i < arr_count; i++) {
        totalSum += arr[i];
    }
    int leftSum = 0;
    for(int i = 0; i < arr_count; i++){
        int rightSum = totalSum - leftSum - arr[i];
        if(leftSum == rightSum){
            return i;
        }
        leftSum += arr[i];
    }
    return 1;
}
```

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

Passed all tests! ✓



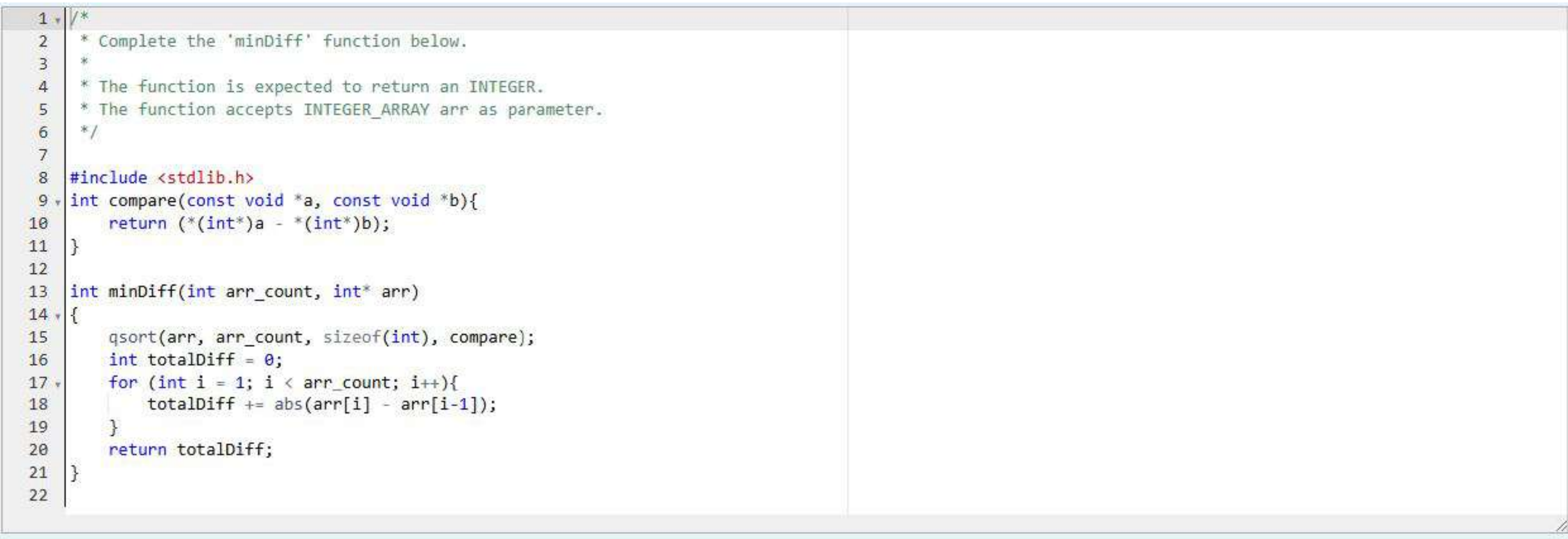
```c
/*
 * Complete the 'arraySum' function below.
 *
 * The function is expected to return an INTEGER.
 * The function accepts INTEGER_ARRAY numbers as parameter.
 */

int arraySum(int numbers_count, int *numbers)
{
    int sum = 0;
    for (int i = 0; i < numbers_count; i++){
        sum += numbers[i];
    }
    return sum;
}

```

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

Passed all tests! ✓

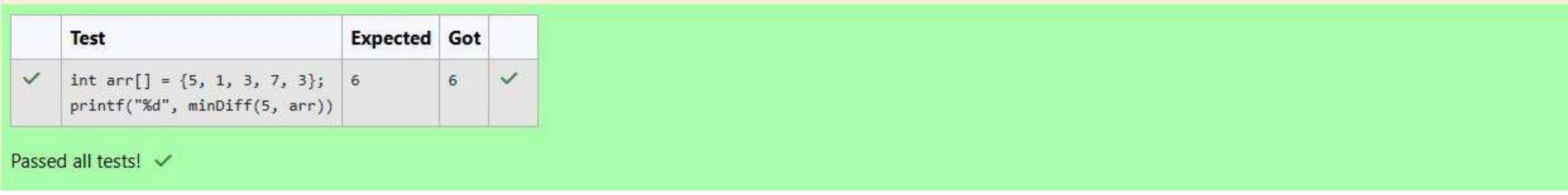


```c
/*
 * Complete the 'minDiff' function below.
 *
 * The function is expected to return an INTEGER.
 * The function accepts INTEGER_ARRAY arr as parameter.
 */

#include <stdlib.h>
int compare(const void *a, const void *b){
    return (*(int*)a - *(int*)b);
}

int minDiff(int arr_count, int* arr)
{
    qsort(arr, arr_count, sizeof(int), compare);
    int totalDiff = 0;
    for (int i = 1; i < arr_count; i++){
        totalDiff += abs(arr[i] - arr[i-1]);
    }
    return totalDiff;
}

```



	Test	Expected	Got	

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

Passed all tests! ✓