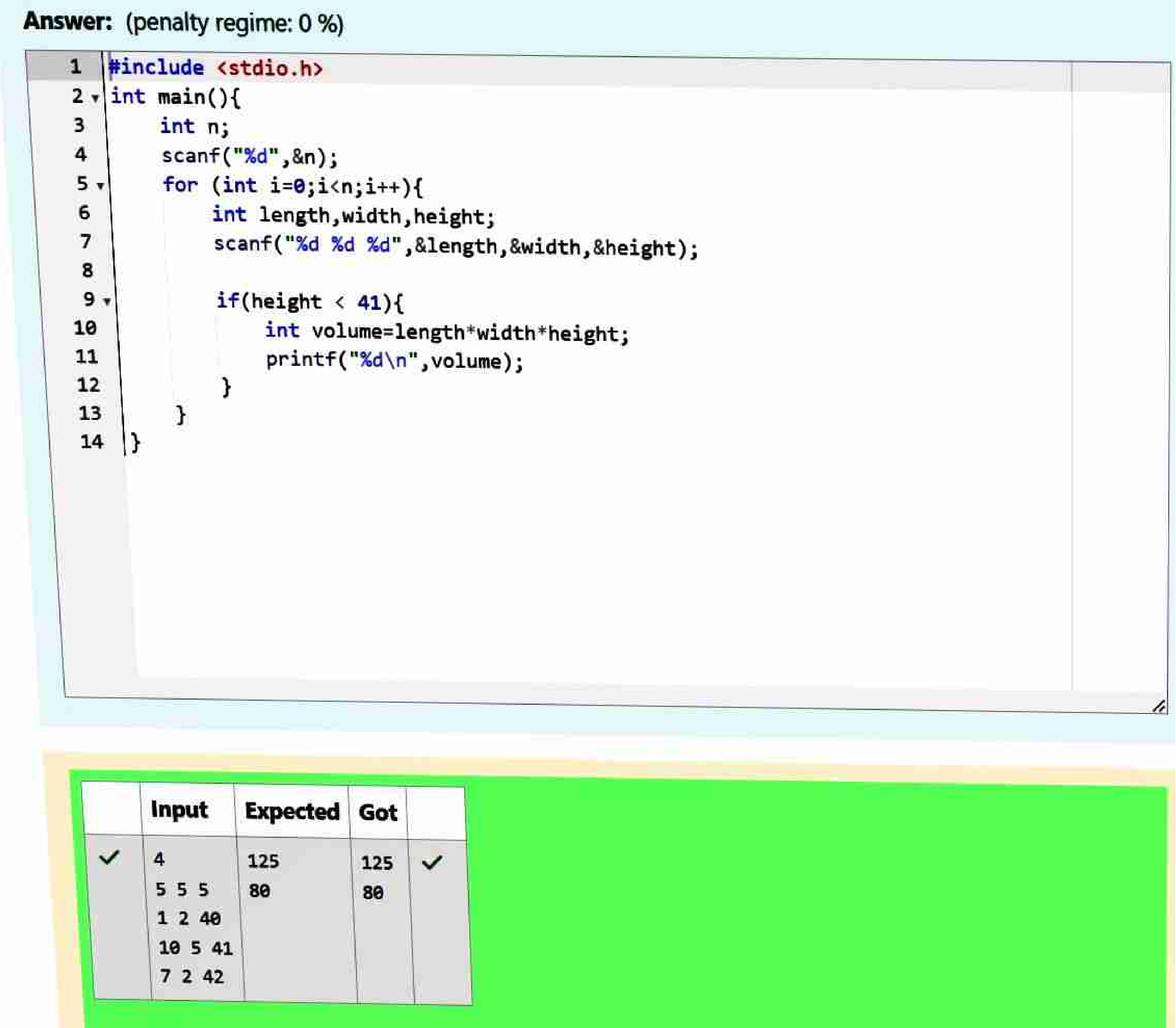
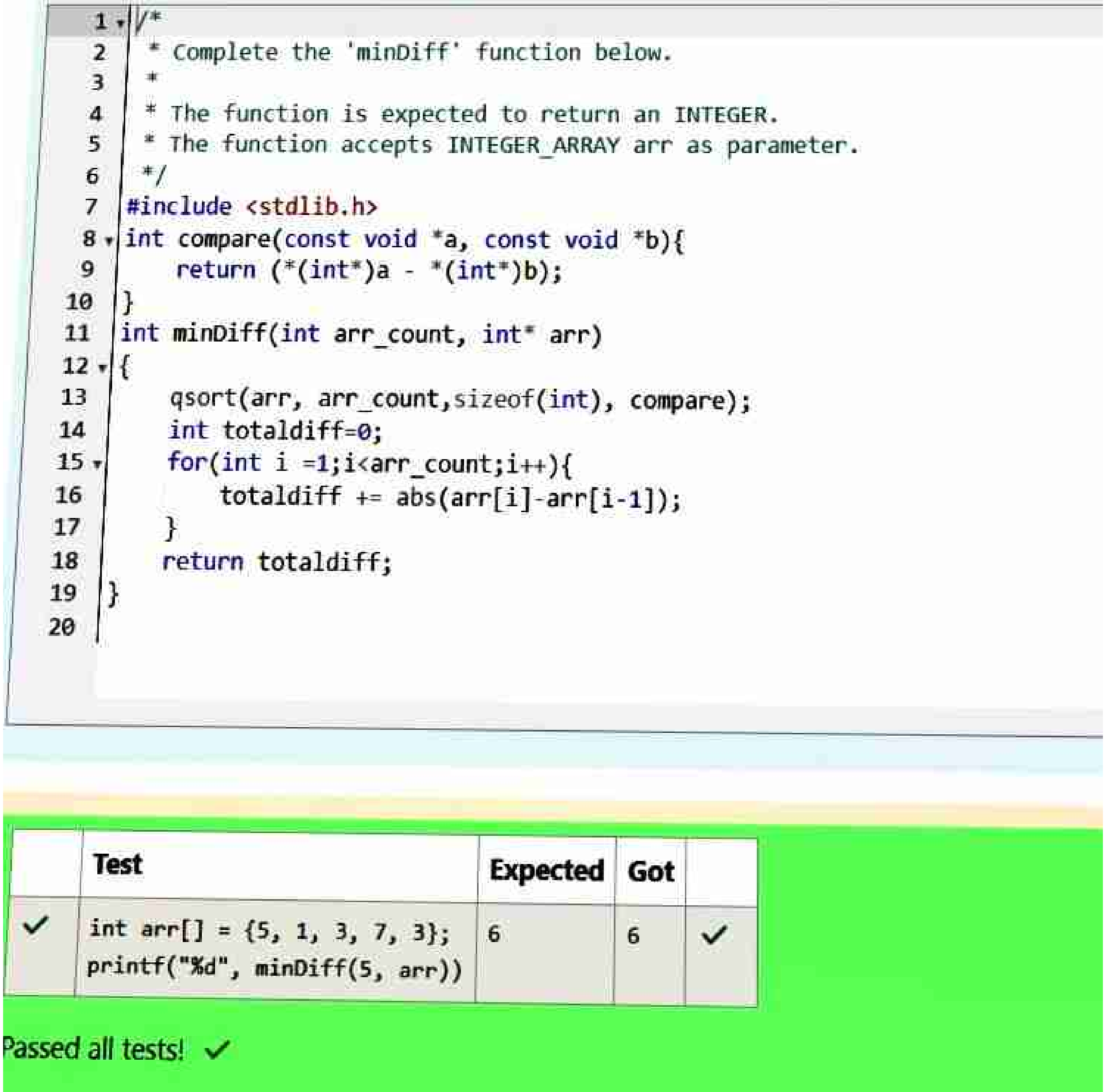


Answer: (penalty regime: 0 %)

```c
#include <stdio.h>
int main(){
    int n;
    scanf("%d",&n);
    for (int i=0;i<n;i++){
        int length,width,height;
        scanf("%d %d %d",&length,&width,&height);

        if(height < 41){
            int volume=length*width*height;
            printf("%d\n",volume);
        }
    }
}
```

	Input	Expected	Got	
✓	4 5 5 5 1 2 40 10 5 41 7 2 42	125 80	125 80	✓

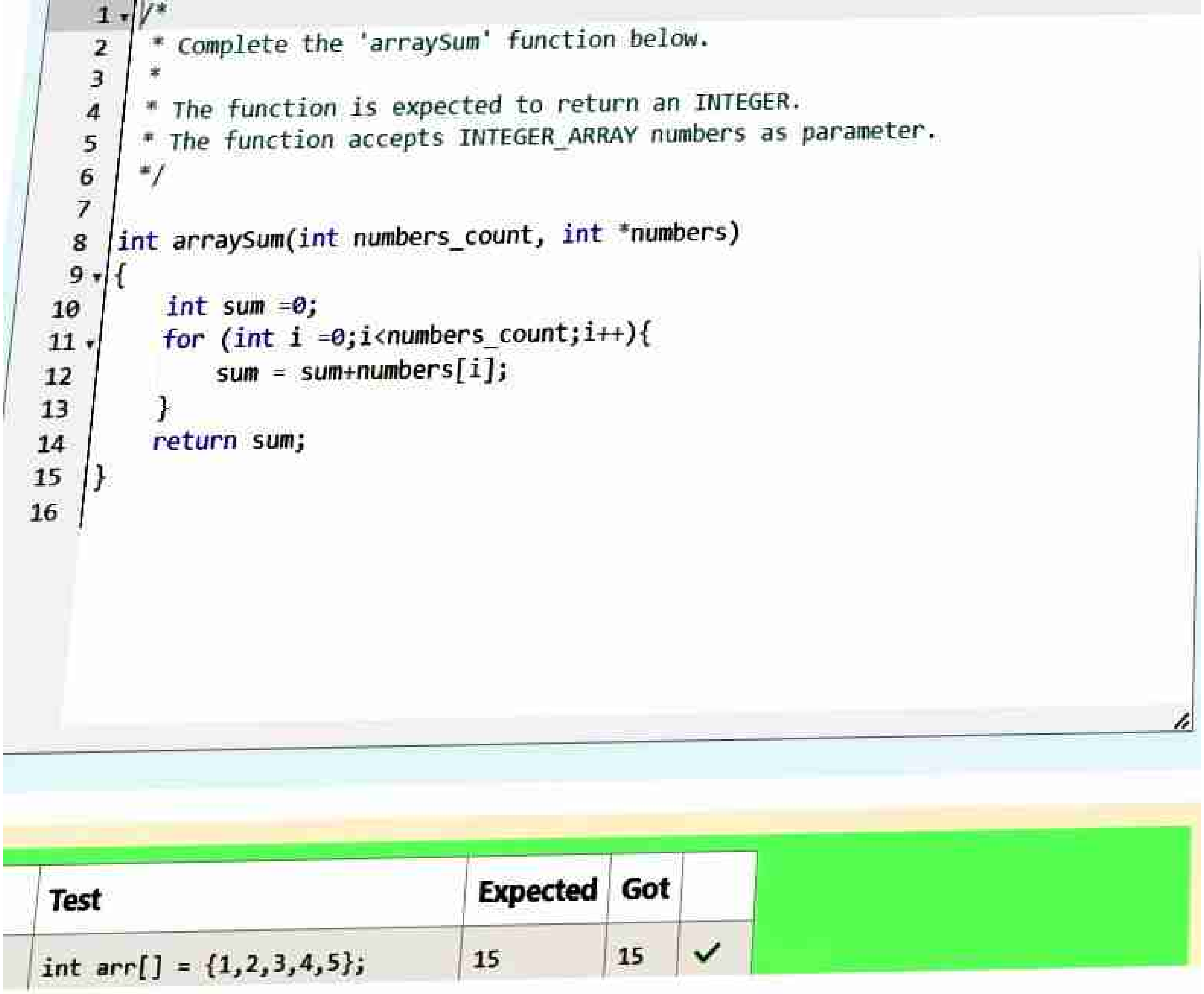


```c
/*
 * Complete the 'minDiff' function below.
 *
 * The function is expected to return an INTEGER.
 * The function accepts INTEGER_ARRAY arr as parameter.
 */
#include <stdlib.h>
int compare(const void *a, const void *b){
    return (*(int*)a - *(int*)b);
}
int minDiff(int arr_count, int* arr)
{
    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	

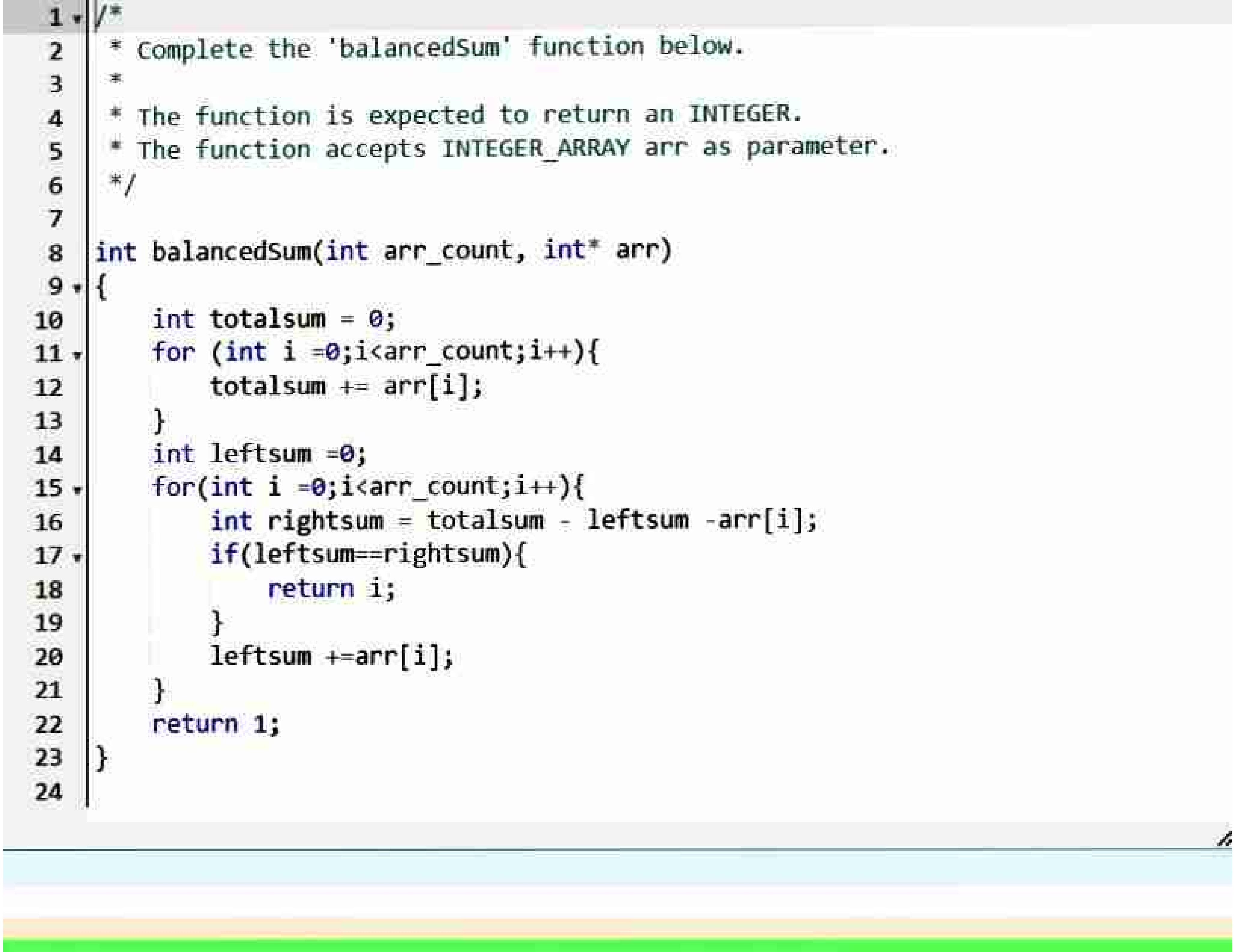
✓	`int arr[] = {5, 1, 3, 7, 3};`
	`printf("%d", minDiff(5, arr))`
		6	6	✓

Passed all tests! ✓

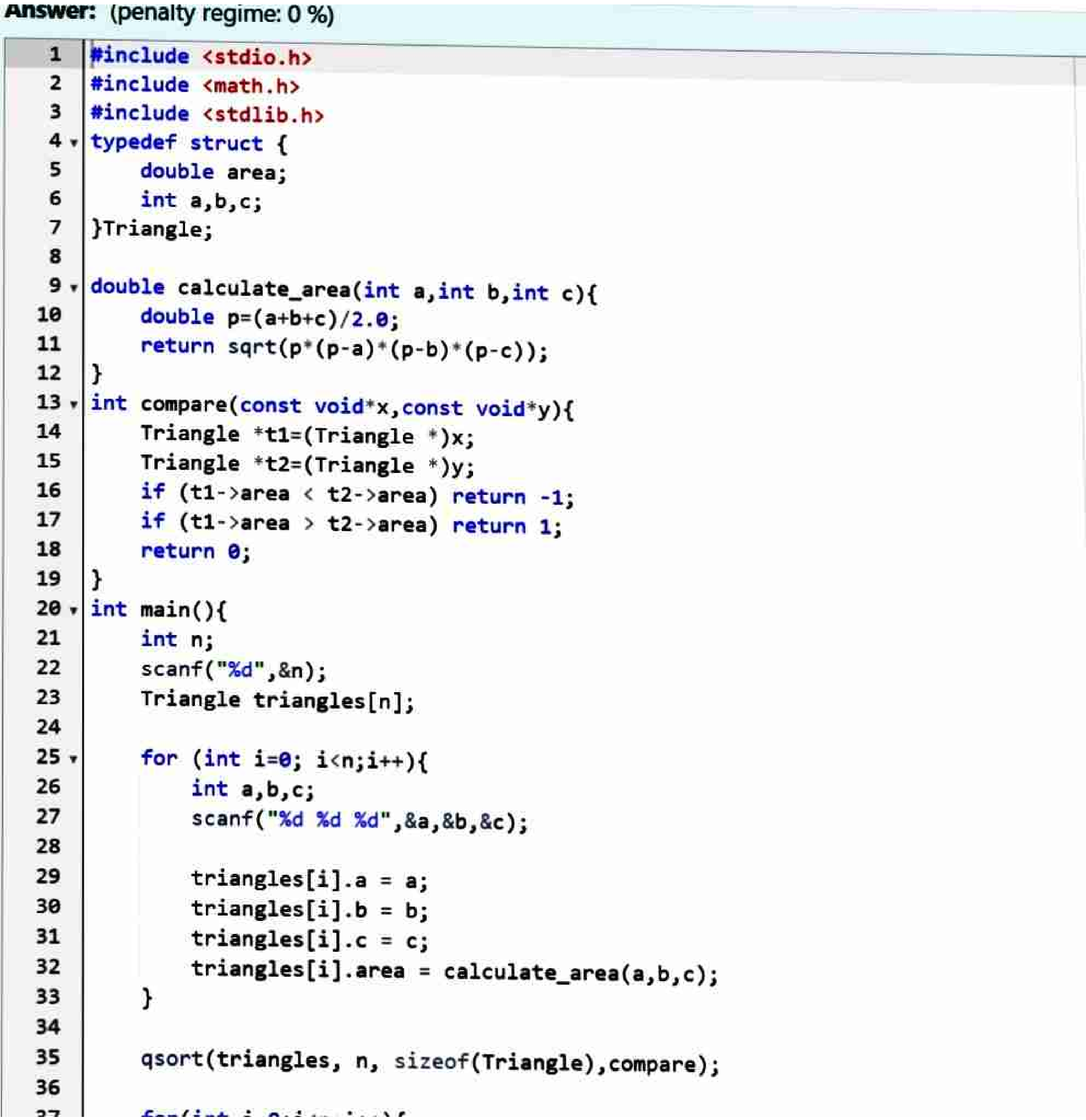


```
/*
 * Complete the 'arraySum' function below.
 *
 * The function is expected to return an INTEGER.
 * The function accepts INTEGER_ARRAY numbers as parameter.
 */

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

Test	Expected	Got	
int arr[] = {1,2,3,4,5};	15	15	✓

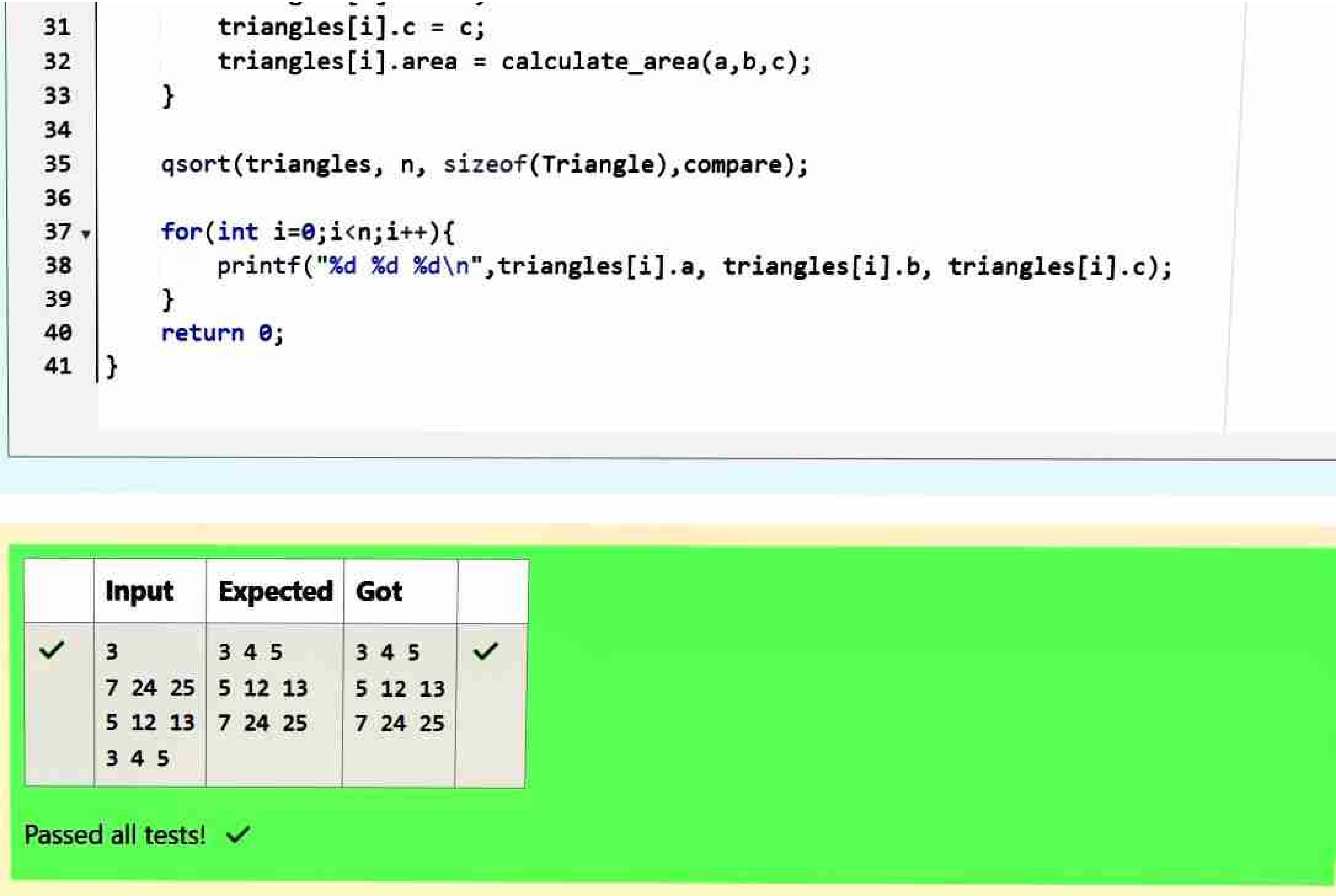


```
/*
 * Complete the 'balancedSum' function below.
 *
 * The function is expected to return an INTEGER.
 * The function accepts INTEGER_ARRAY arr as parameter.
 */

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;
}

```



Answer: (penalty regime: 0 %)

```c
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
typedef struct {
    double area;
    int a,b,c;
}Triangle;

double calculate_area(int a,int b,int c){
    double p=(a+b+c)/2.0;
    return sqrt(p*(p-a)*(p-b)*(p-c));
}
int compare(const void*x,const void*y){
    Triangle *t1=(Triangle *)x;
    Triangle *t2=(Triangle *)y;
    if (t1->area < t2->area) return -1;
    if (t1->area > t2->area) return 1;
    return 0;
}
int main(){
    int n;
    scanf("%d",&n);
    Triangle triangles[n];

    for (int i=0; i<n;i++){
        int a,b,c;
        scanf("%d %d %d",&a,&b,&c);

        triangles[i].a = a;
        triangles[i].b = b;
        triangles[i].c = c;
        triangles[i].area = calculate_area(a,b,c);
    }

    qsort(triangles, n, sizeof(Triangle),compare);

    for(int i=0;i<n;i++){
```



```c
        triangles[i].c = c;
        triangles[i].area = calculate_area(a,b,c);
    }

    qsort(triangles, n, sizeof(Triangle),compare);

    for(int i=0;i<n;i++){
        printf("%d %d %d\n",triangles[i].a, triangles[i].b, triangles[i].c);
    }
    return 0;
}
```

	Input	Expected	Got	
✓	3 7 24 25 5 12 13 3 4 5	3 4 5 5 12 13 7 24 25	3 4 5 5 12 13 7 24 25	✓

Passed all tests! ✓