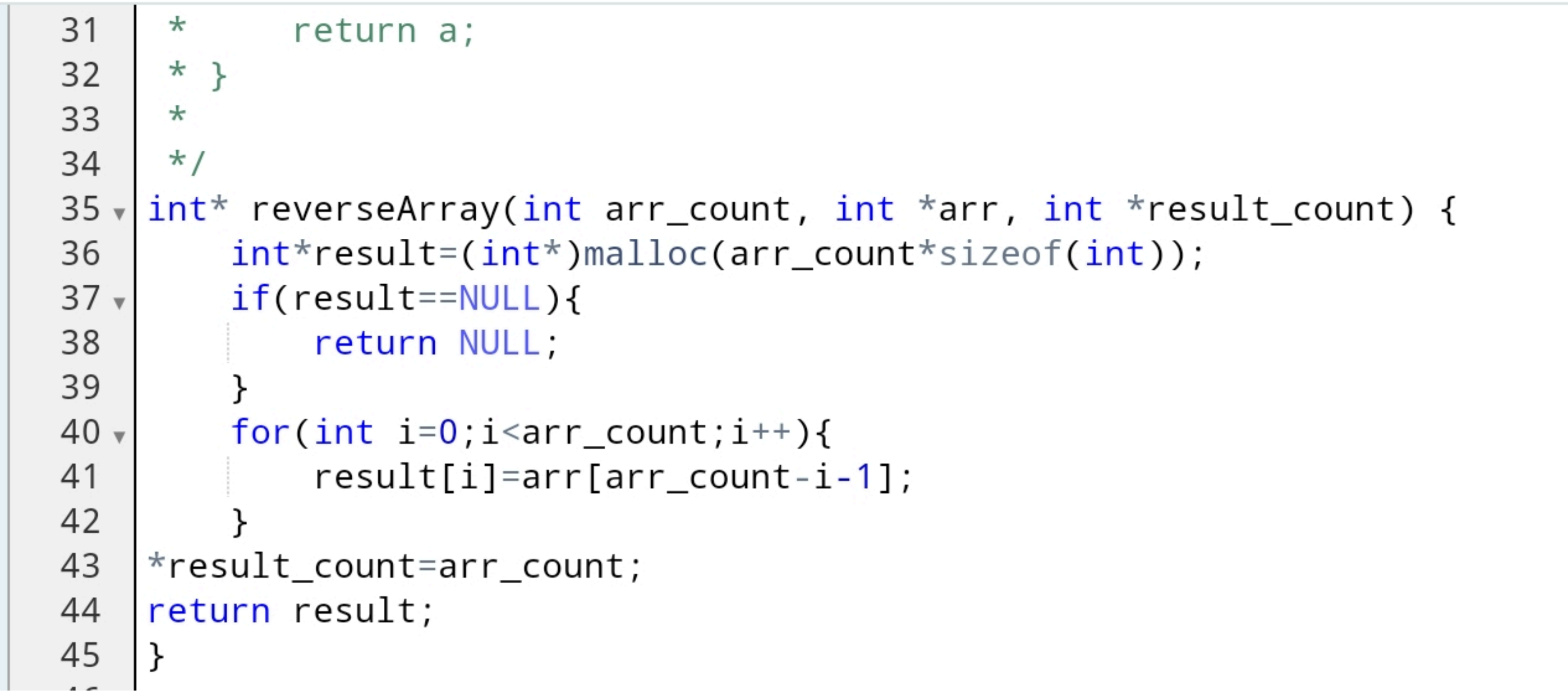
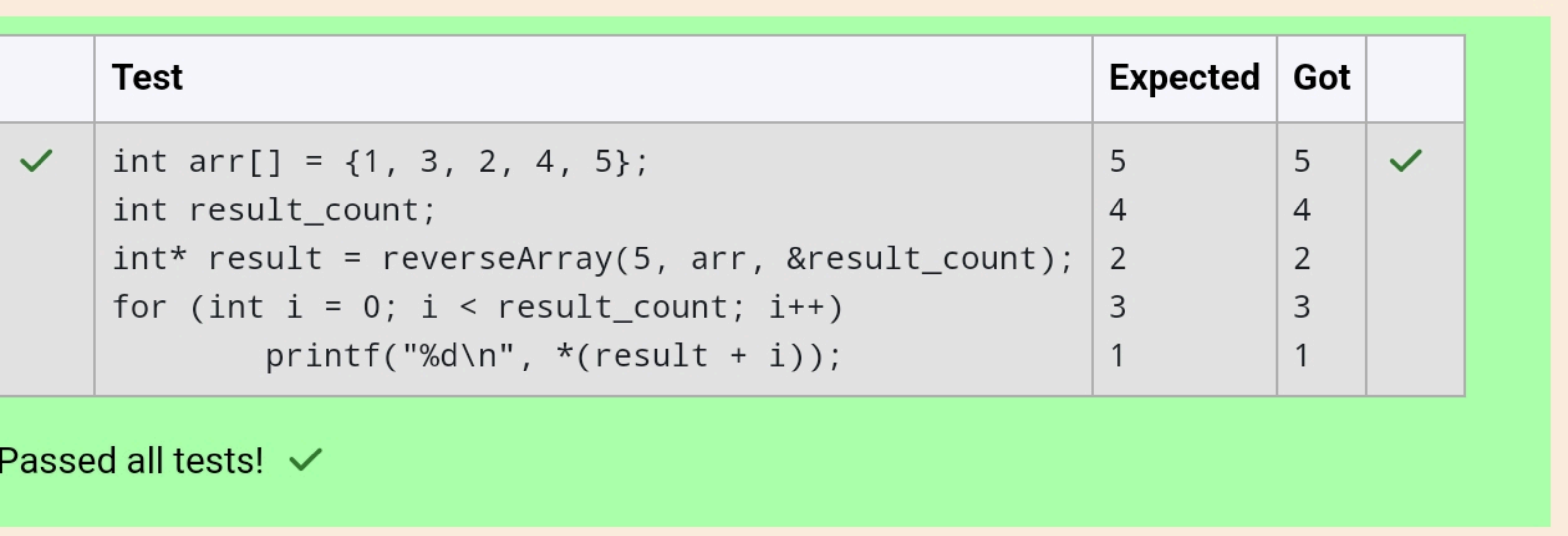


```
 *     return a;
 * }
 *
 */
int* reverseArray(int arr_count, int *arr, int *result_count) {
    int*result=(int*)malloc(arr_count*sizeof(int));
    if(result==NULL){
        return NULL;
    }
    for(int i=0;i<arr_count;i++){
        result[i]=arr[arr_count-i-1];
    }
*result_count=arr_count;
return result;
}
```

Gowurisankar.P
240801089
ECE B

Gowurisankar.P
240801089
ECE B

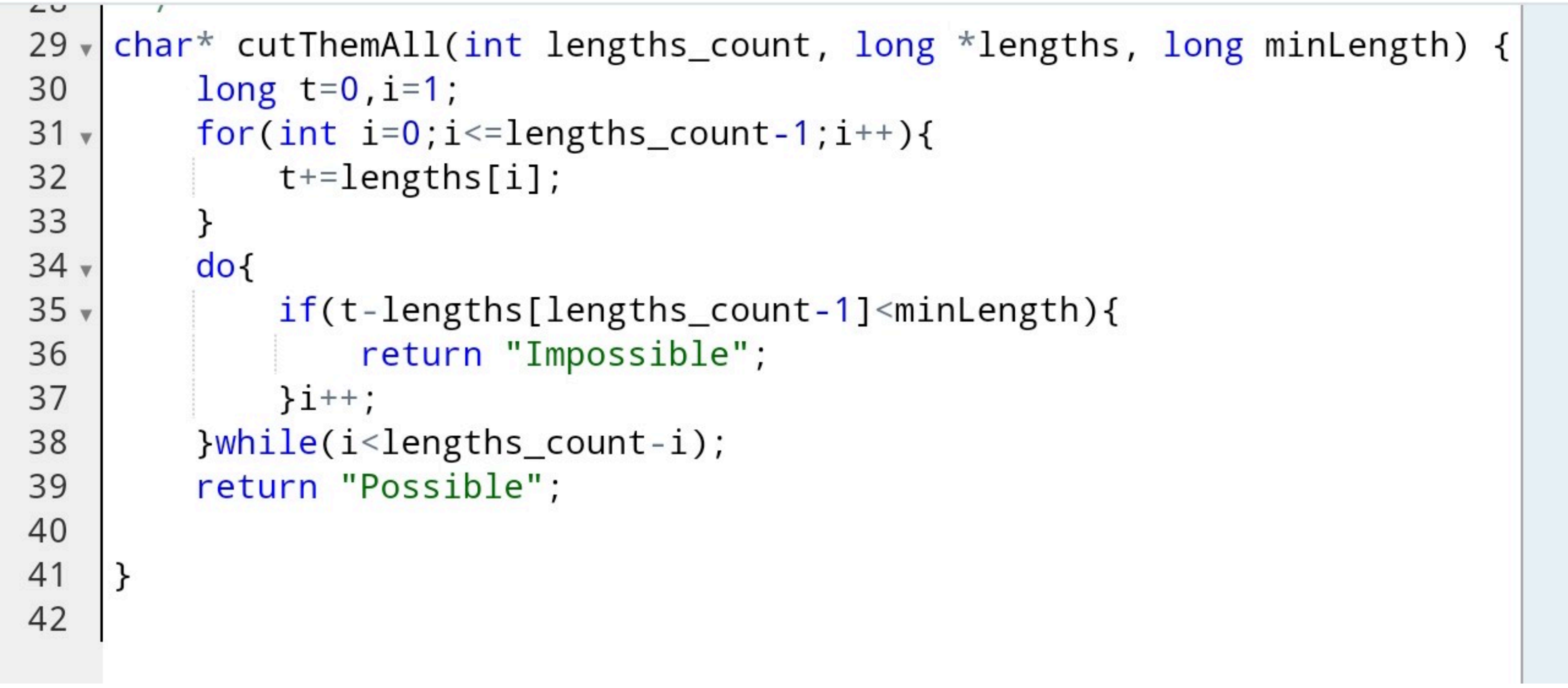


	Test	Expected	Got	

✓	`int arr[] = {1, 3, 2, 4, 5};`
	`int result_count;`
	`int* result = reverseArray(5, arr, &result_count);`
	`for (int i = 0; i < result_count; i++)`
	`        printf("%d\n", *(result + i));`
		5
		4
		2
		3
		1
			5
			4
			2
			3
			1
				✓

Passed all tests! ✓

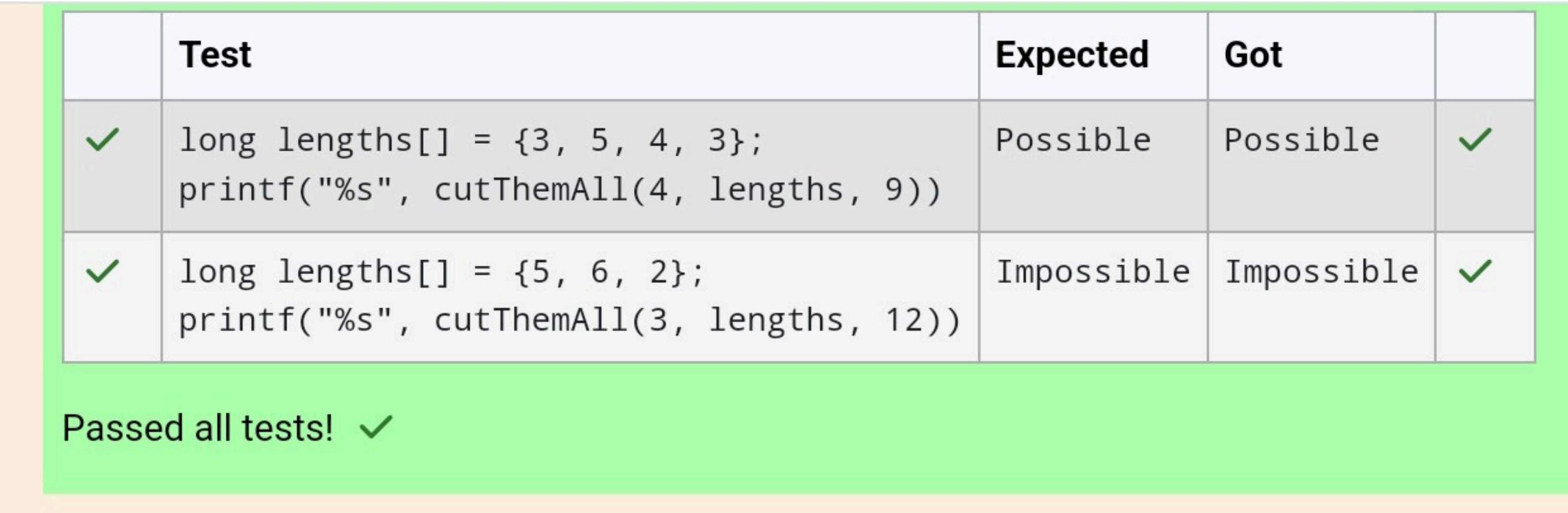
Gowurisankar.P
240801089
ECE B



```
char* cutThemAll(int lengths_count, long *lengths, long minLength) {
    long t=0,i=1;
    for(int i=0;i<=lengths_count-1;i++){
        t+=lengths[i];
    }
    do{
        if(t-lengths[lengths_count-1]<minLength){
            return "Impossible";
        }i++;
    }while(i<lengths_count-i);
    return "Possible";

}
```

Gowurisankar.P
240801089
ECE B



	Test	Expected	Got	

✓	`long lengths[] = {3, 5, 4, 3};`
	`printf("%s", cutThemAll(4, lengths, 9))`
		Possible	Possible	✓

✓	`long lengths[] = {5, 6, 2};`
	`printf("%s", cutThemAll(3, lengths, 12))`
		Impossible	Impossible	✓

Passed all tests! ✓