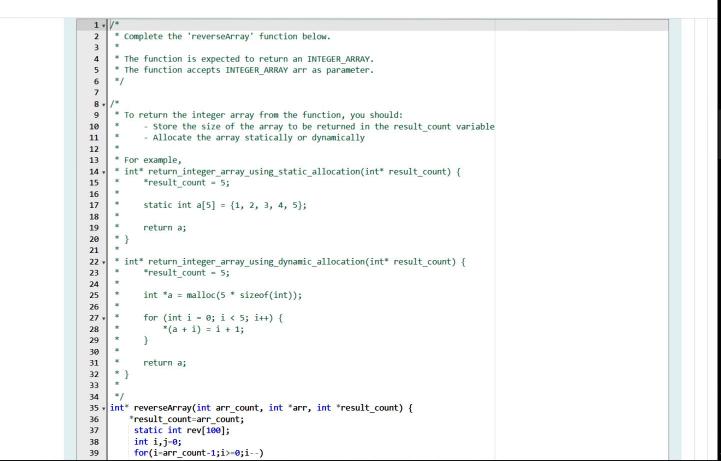


REC-CIS	
	Sample Case 0
	Sample Input Fo
	5
	1
	3
	2
	4
	5
	Sample Output
	5
	4
	2
	3
	1
	Explanation
	The input array is
	Sample Case 1
	Sample Input Fo
	4
	17
	10
	21
	45
	Sample Output

1
1
1 :
;
-3
3
ı
•
1
1

Sample Case 0
Sample Input For Custom Testing
5
1
3
2
4
5
Sample Output
5
4
2
3
1
Explanation
The input array is [1, 3, 2, 4, 5], so the reverse of the input array is [5, 4, 2, 3, 1].
Sample Case 1
Sample Input For Custom Testing
4
17
10
21
45



```
26
27
           for (int i = 0; i < 5; i++) {
               *(a + i) = i + 1;
28
29
30
31
           return a;
     * }
32
33
34
     int* reverseArray(int arr_count, int *arr, int *result_count) {
         *result count=arr count;
36
         static int rev[100];
37
         int i, j=0;
38
         for(i=arr_count-1;i>=0;i--)
39
         rev[j++]=arr[i];
40
41
         return rev;
42
43
44
                                                     Expected Got
    Test
int arr[] = {1, 3, 2, 4, 5};
    int result_count;
    int* result = reverseArray(5, arr, &result_count);
```

3

*result count = 5;

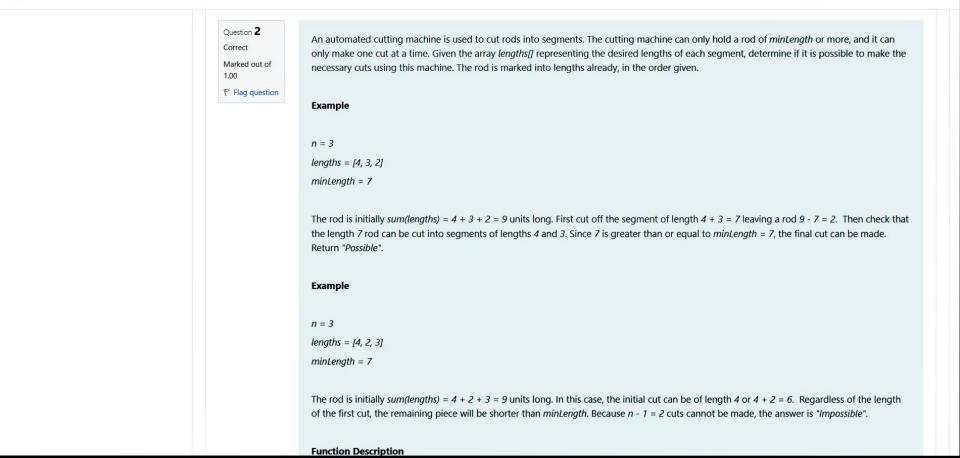
for (int i = 0; i < result_count; i++)</pre>

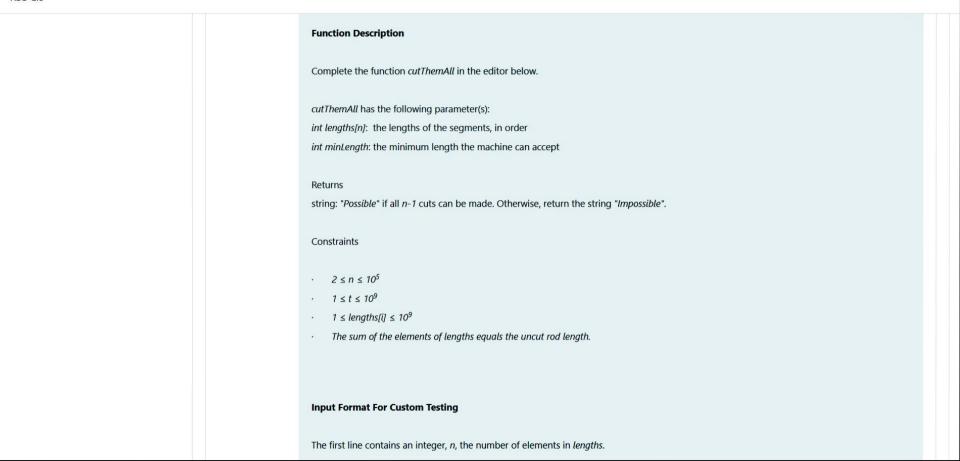
Passed all tests! <

printf("%d\n", *(result + i));

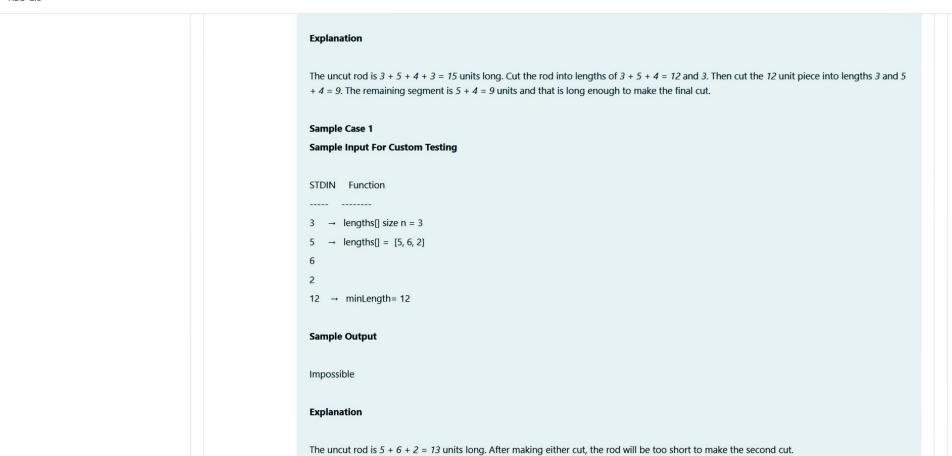
int *a = malloc(5 * sizeof(int));

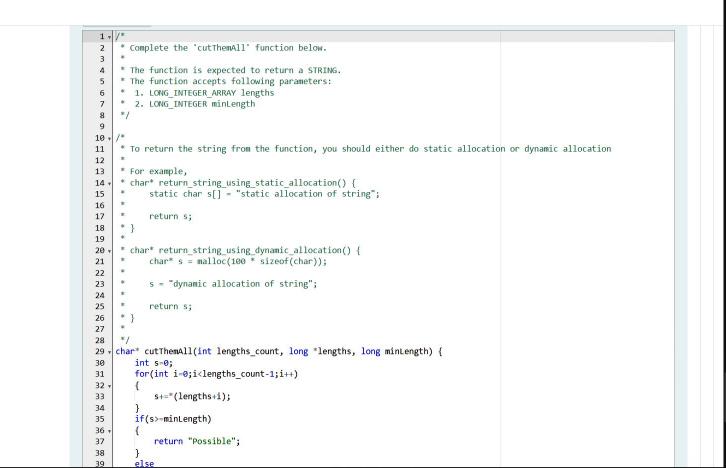
23 24 25





Possible





```
return s;
26
27
28
    char* cutThemAll(int lengths_count, long *lengths, long minLength) {
29
30
        int s=0;
        for(int i=0;i<lengths_count-1;i++)</pre>
31
32
            s+=*(lengths+i);
33
34
        if(s>=minLength)
35
36 •
            return "Possible";
37
38
        else
39
40
            return "Impossible";
41
42
43
44
45
```

	Test	Expected	Got	
~	<pre>long lengths[] = {3, 5, 4, 3}; printf("%s", cutThemAll(4, lengths, 9))</pre>	Possible	Possible	~
~	<pre>long lengths[] = {5, 6, 2}; printf("%s", cutThemAll(3, lengths, 12))</pre>	Impossible	Impossible	~