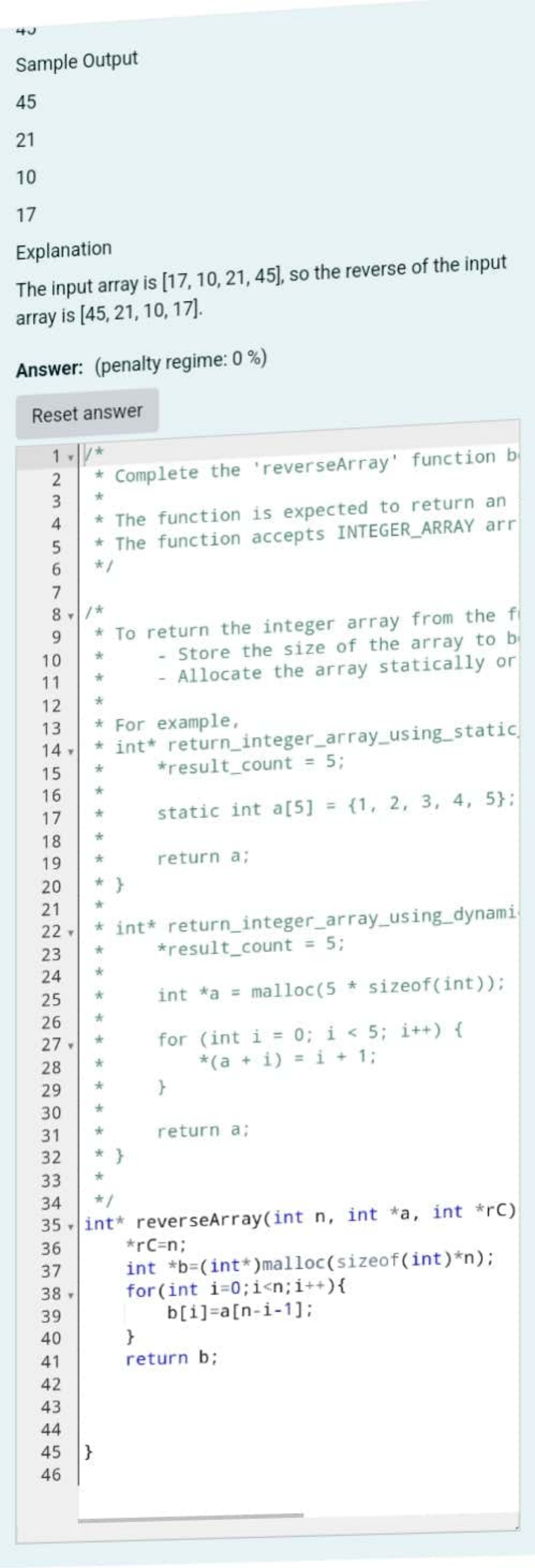


45

Sample Output

45

21

10

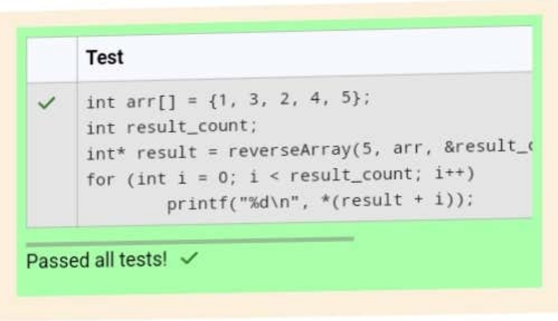
17

Explanation

The input array is [17, 10, 21, 45], so the reverse of the input array is [45, 21, 10, 17].

Answer: (penalty regime: 0 %)

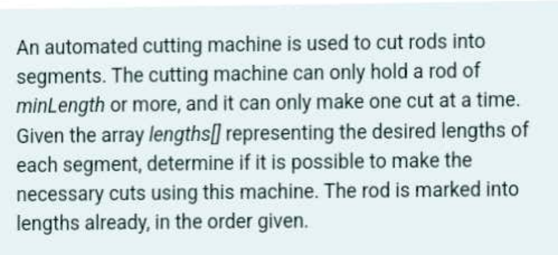
```
/*
 * Complete the 'reverseArray' function b
 *
 * The function is expected to return an
 * The function accepts INTEGER_ARRAY arr
 */

/*
 * To return the integer array from the f
 *     - Store the size of the array to b
 *     - Allocate the array statically or
 *
 * For example,
 * int* return_integer_array_using_static
 *     *result_count = 5;
 *
 *     static int a[5] = {1, 2, 3, 4, 5};
 *
 *     return a;
 * }
 *
 * int* return_integer_array_using_dynami
 *     *result_count = 5;
 *
 *     int *a = malloc(5 * sizeof(int));
 *
 *     for (int i = 0; i < 5; i++) {
 *         *(a + i) = i + 1;
 *     }
 *
 *     return a;
 * }
 *
 */
int* reverseArray(int n, int *a, int *rC)
    *rC=n;
    int *b=(int*)malloc(sizeof(int)*n);
    for(int i=0;i<n;i++){
        b[i]=a[n-i-1];
    }
    return b;



}
```



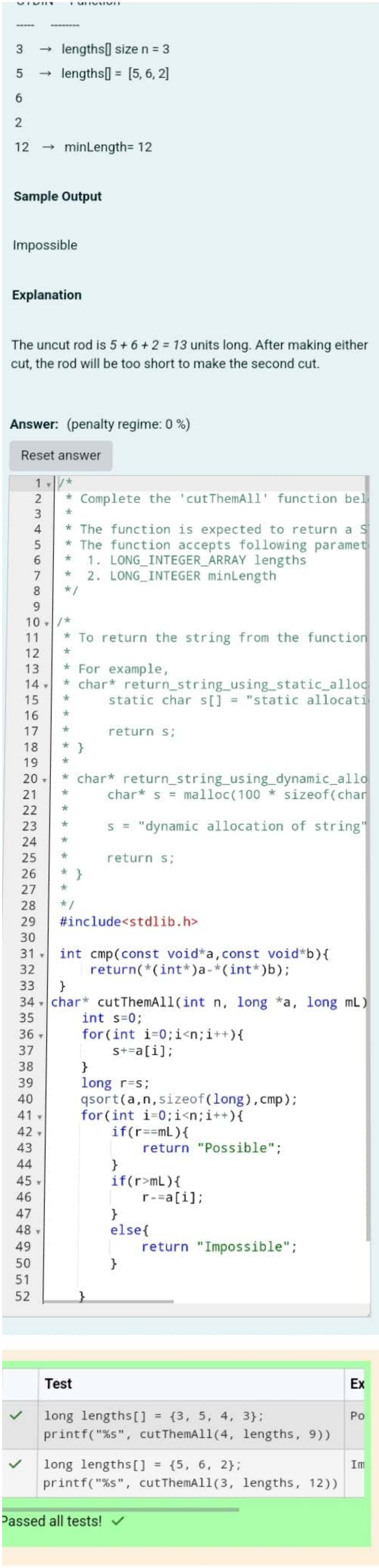
	Test

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

Passed all tests! ✓



An automated cutting machine is used to cut rods into segments. The cutting machine can only hold a rod of *minLength* or more, and it can only make one cut at a time. Given the array *lengths[]* representing the desired lengths of each segment, determine if it is possible to make the necessary cuts using this machine. The rod is marked into lengths already, in the order given.



```
STDIN    Function
-----    --------
3    →  lengths[] size n = 3
5    →  lengths[] =  [5, 6, 2]
6
2
12   →  minLength= 12
```

Sample Output

Impossible

Explanation

The uncut rod is *5 + 6 + 2 = 13* units long. After making either cut, the rod will be too short to make the second cut.

Answer: (penalty regime: 0 %)

```
/*
 * Complete the 'cutThemAll' function bel
 *
 * The function is expected to return a S
 * The function accepts following paramet
 *  1. LONG_INTEGER_ARRAY lengths
 *  2. LONG_INTEGER minLength
 */

/*
 * To return the string from the function
 *
 * For example,
 * char* return_string_using_static_alloc
 *     static char s[] = "static allocati
 *
 *     return s;
 * }
 *
 * char* return_string_using_dynamic_allo
 *     char* s = malloc(100 * sizeof(char
 *
 *     s = "dynamic allocation of string"
 *
 *     return s;
 * }
 *
 */
#include<stdlib.h>

int cmp(const void*a,const void*b){
    return(*(int*)a-*(int*)b);
}
char* cutThemAll(int n, long *a, long mL)
    int s=0;
    for(int i=0;i<n;i++){
        s+=a[i];
    }
    long r=s;
    qsort(a,n,sizeof(long),cmp);
    for(int i=0;i<n;i++){
        if(r==mL){
            return "Possible";
        }
        if(r>mL){
            r-=a[i];
        }
        else{
            return "Impossible";
        }

    }
```

	Test	Ex

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

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

Passed all tests! ✓