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

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 %)

Reset answer

```c
/*
 * 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;
 * }
 *
 */
#include<stdio.h>
#include<stdlib.h>
int* reverseArray(int arr_count, int *arr
    int*result=(int*)malloc(arr_count*siz
    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;

}
```

	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! ✓

The uncut rod is *3 + 5 + 4 + 3 = 15* units long. Cut the rod into lengths of *3 + 5 + 4 = 12* and *3*. Then cut the *12* unit piece into lengths *3* and *5 + 4 = 9*. The remaining segment is *5 + 4 = 9* units and that is long enough to make the final cut.

Sample Case 1

Sample Input For Custom Testing

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 %)

Reset answer

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

}
```

	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! ✓

Finish review