Answer: (penalty regime: 0 %)

Reset answer

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

/*
 * To return the integer array from the fu
 *     - Store the size of the array to be
 *     - 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_dynamic
 *     *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 arr_count, int *arr,
    *result_count=arr_count;
    for(int i=0;i<arr_count/2;i++)
    {
        int temp=arr[i];
        arr[i]=arr[arr_count-i-1];
        arr[arr_count-i-1]=temp;
    }
    return arr;

}

```

	Test

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

Passed all tests! ✓

Answer: (penalty regime: 0 %)

Reset answer

```
/*
 * 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;
 * }
 *
 */
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-i-1]<m
        {
            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! ✓