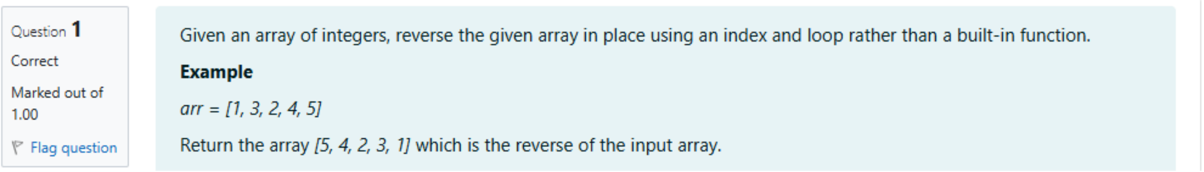
SAISANJAY SS 2024-CSE S2

Week-15-Pointers



Question **1**

Correct

Marked out of 1.00

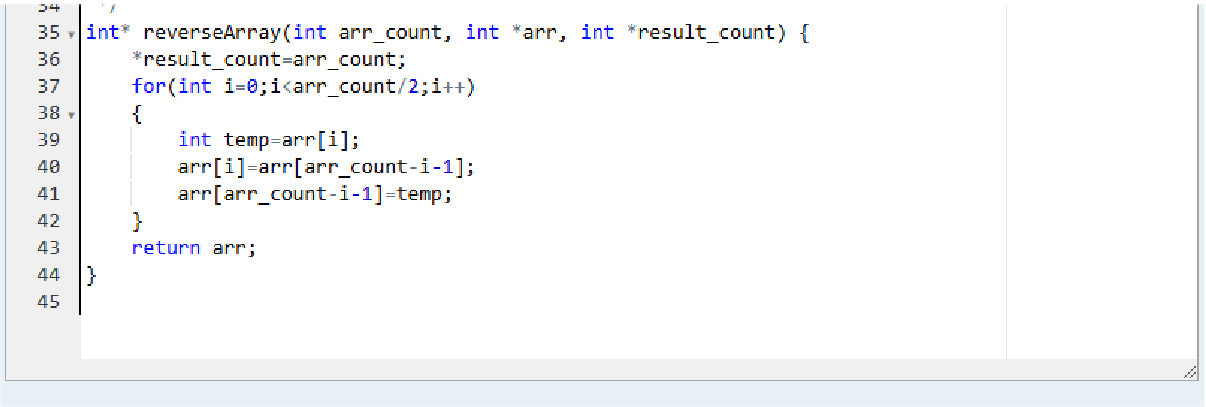
Flag question

Given an array of integers, reverse the given array in place using an index and loop rather than a built-in function.

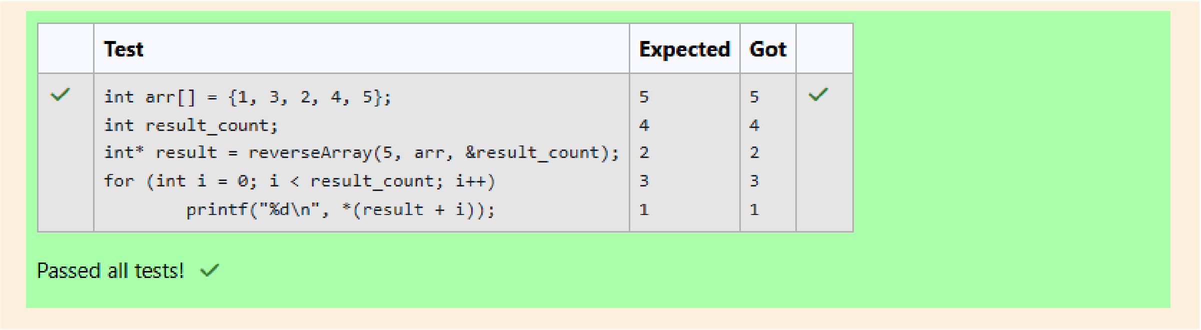
Example

arr = [1, 3, 2, 4, 5]

Return the array *[5, 4, 2, 3, 1]* which is the reverse of the input array.



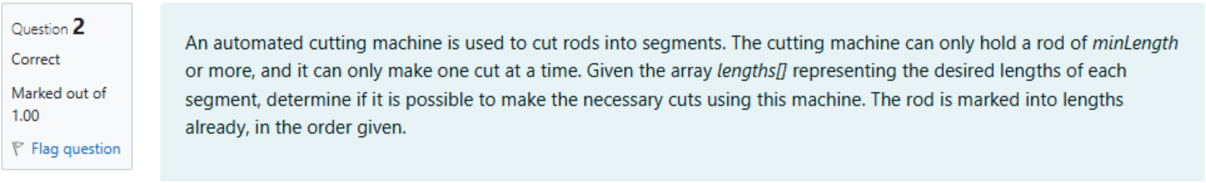
```
 */
int* reverseArray(int arr_count, int *arr, int *result_count) {
    *result_count=arr_count;
    for(int i=0;i<arr_count/2;i++)
    {
        int temp=arr[i];
        arr[i]=arr[arr_count-i-1];
        arr[arr_count-i-1]=temp;
    }
    return arr;
}

```



	Test	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! ✓



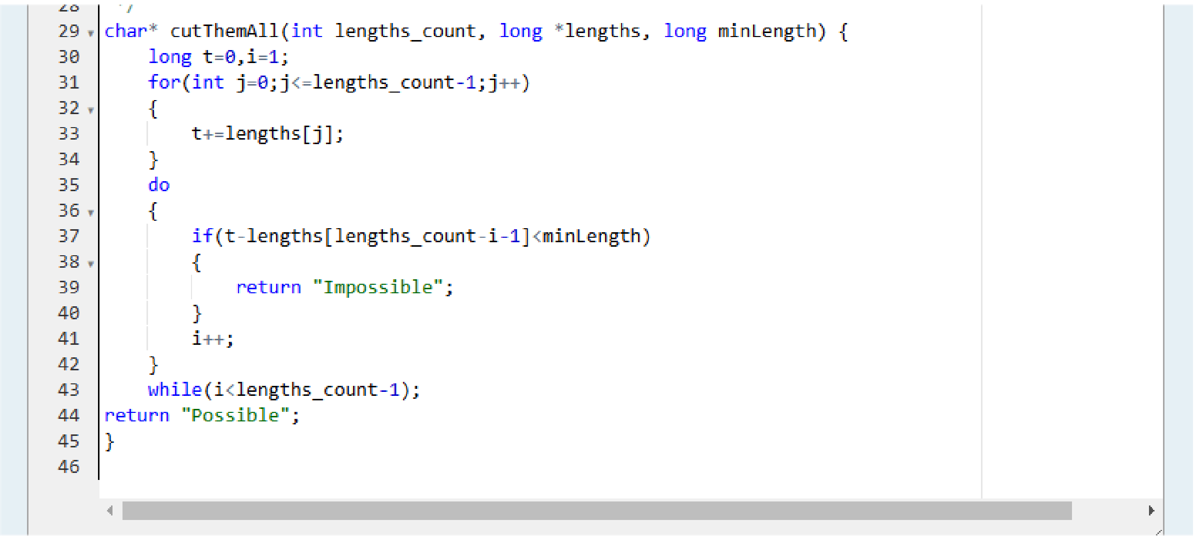
Question **2**

Correct

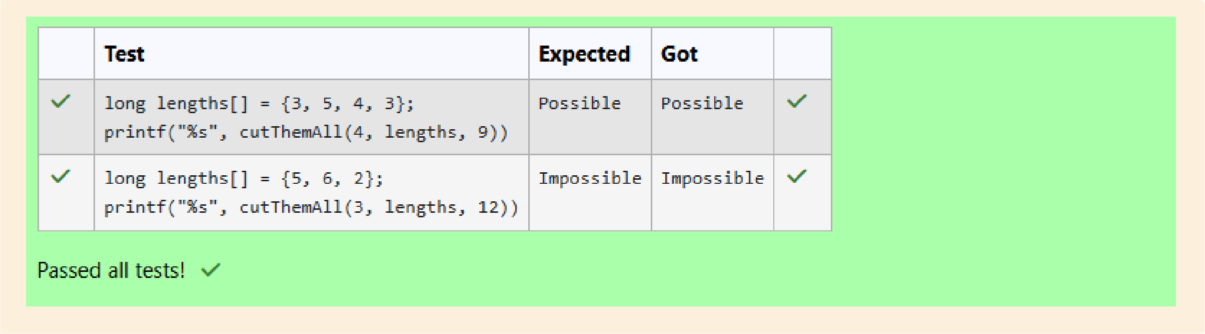
Marked out of 1.00

Flag question

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.



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



	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! ✓