19m left

⌘

ALL

ⓘ

1

2

3

2. Reverse Array Queries

For a given array of integers, perform operations on the array. Return the resulting array after all operations have been applied in the order given. Each operation contains two indices. Reverse the subarray between those zero-based indices, inclusive.

Example
arr = [5, 3, 2, 1, 3]
operations = [[0, 1], [1, 3]]

In the first operation, reverse the subarray from *arr[0]* through *arr[1]: arr' = [3, 5, 2, 1, 3]*
In the second operation, reverse the subarray from *arr'[1]* through *arr'[3]: arr'' = [3, 1, 2, 5, 3]*
All operations have been performed, so return the array *[3, 1, 2, 5, 3]*.

Function Description
Complete the function *performOperations* in the editor below.

The function has the following parameter(s):
int arr[n]: an array of integers
int operations[q][2]: a 2-dimensional array of starting and ending indices

Returns:
int[n]: the final array after all operations have been performed

Constraints

- $1 \le n, q \le 10^3$
- $1 \le arr[i] \le 10^3$
- $0 \le operations[i][0] \le operations[i][1] < n$

▶ **Input Format Format for Custom Testing**

▼ **Sample Case 0**

Sample Input

```
STDIN       Function
-----       -----
3       →   arr[] size n = 3
1       →   arr = [1, 2, 3]
```