Reverse a list

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.

**Function Description**

Complete the function *reverseArray* in the editor below.

reverseArray has the following parameter(s):

    int *arr[n]:*  an array of integers

****Return****

*int[n]:* the array in reverse order

**Constraints**

* 1 ≤ *n* ≤ 100
* 0 < *arr[i]* ≤ 100

Input Format For Custom Testing

The first line contains an integer, *n*, the number of elements in *arr*.  
Each line *i* of the *n* subsequent lines (where *0 ≤ i < n*) contains an integer, *arr[i]*.

Sample Case 0

**Sample Input For Custom Testing**

5

1

3

2

4

5

**Sample Output**

5

4

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].

import java.io.\*;

import java.math.\*;

import java.security.\*;

import java.text.\*;

import java.util.\*;

import java.util.concurrent.\*;

import java.util.function.\*;

import java.util.regex.\*;

import java.util.stream.\*;

import static java.util.stream.Collectors.joining;

import static java.util.stream.Collectors.toList;

class Result {

/\*

\* Complete the 'reverseArray' function below.

\*

\* The function is expected to return an INTEGER\_ARRAY.

\* The function accepts INTEGER\_ARRAY arr as parameter.

\*/

public static List<Integer> reverseArray(List<Integer> arr) {

// Write your code here

}

}

public class Solution {

public static void main(String[] args) throws IOException {

BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(System.in));

BufferedWriter bufferedWriter = new BufferedWriter(new FileWriter(System.getenv("OUTPUT\_PATH")));

int arrCount = Integer.parseInt(bufferedReader.readLine().trim());

List<Integer> arr = IntStream.range(0, arrCount).mapToObj(i -> {

try {

return bufferedReader.readLine().replaceAll("\\s+$", "");

} catch (IOException ex) {

throw new RuntimeException(ex);

}

})

.map(String::trim)

.map(Integer::parseInt)

.collect(toList());

List<Integer> result = Result.reverseArray(arr);

bufferedWriter.write(

result.stream()

.map(Object::toString)

.collect(joining("\n"))

+ "\n"

);

bufferedReader.close();

bufferedWriter.close();

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*