An *array* is a type of data structure that stores elements of the same type in a contiguous block of memory. In an array, , of size , each memory location has some unique index,  (where ), that can be referenced as  or .

Reverse an array of integers.

**Note:** If you've already solved our C++ domain's *Arrays Introduction* challenge, you may want to skip this.

**Example**

Return .

**Function Description**

Complete the function *reverseArray* in the editor below.

*reverseArray* has the following parameter(s):

* *int A[n]*: the array to reverse

**Returns**

* *int[n]*: the reversed array

**Input Format**

The first line contains an integer, , the number of integers in .  
The second line contains  space-separated integers that make up .

**C**

**Sample Input 1**

Array: arr1432

4  
1 4 3 2

2 3 4 1

Answer –

using System.CodeDom.Compiler;

using System.Collections.Generic;

using System.Collections;

using System.ComponentModel;

using System.Diagnostics.CodeAnalysis;

using System.Globalization;

using System.IO;

using System.Linq;

using System.Reflection;

using System.Runtime.Serialization;

using System.Text.RegularExpressions;

using System.Text;

using System;

class Result

{

/\*

\* Complete the 'reverseArray' function below.

\*

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

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

\*/

public static List<int> reverseArray(List<int> a)

{

List<int> b =new List<int>();

a.Reverse();

foreach(var i in a)

{

b.Add(i);

}

return b;

}

}

class Solution

{

public static void Main(string[] args)

{

TextWriter textWriter = new StreamWriter(@System.Environment.GetEnvironmentVariable("OUTPUT\_PATH"), true);

int arrCount = Convert.ToInt32(Console.ReadLine().Trim());

List<int> arr = Console.ReadLine().TrimEnd().Split(' ').ToList().Select(arrTemp => Convert.ToInt32(arrTemp)).ToList();

List<int> res = Result.reverseArray(arr);

textWriter.WriteLine(String.Join(" ", res));

textWriter.Flush();

textWriter.Close();

}

}