We define the *middle* of the array arr as follows:

* if arr contains an odd number of elements, its*middle* is the element whose index number is the same when counting from the beginning of the array and from its end;
* if arr contains an even number of elements, its*middle* is the sum of the two elements whose index numbers when counting from the beginning and from the end of the array differ by one.

An array is called *smooth* if its its first and its last elements are equal to one another and to the *middle*. Given an array arr, determine if it is *smooth* or not.

**Example**

* For arr = [7, 2, 2, 5, 10, 7], the output should be  
  isSmooth(arr) = true.

The first and the last elements of arr are equal to7, and its *middle* also equals 2 + 5 = 7. Thus, the array is *smooth* and the output is true.

* For arr = [-5, -5, 10], the output should be  
  isSmooth(arr) = false.

The first and *middle* elements are equal to -5, but the last element equals 10. Thus, arr is not*smooth* and the output is false.

**Input/Output**

* **[time limit] 3000ms (cs)**
* **[input] array.integer arr**

The given array.

*Constraints:*  
2 ≤ arr.length ≤ 105,  
-109 ≤ arr[i] ≤ 109.

* **[output] boolean**

true if arr is *smooth*, false otherwise.

<https://codefights.com/arcade/code-arcade/list-forest-edge/3LmZafR9wMCWpdgra>

static bool isSmooth(int[] arr)

{

if (arr[0] == arr[arr.Length - 1])

{

if (arr.Length % 2 == 0)

{

int medio = arr[arr.Length / 2 - 1] + arr[arr.Length / 2];

if (medio == arr[0])

{

return true;

}

}

else

{

if (arr[arr.Length / 2] == arr[0])

{

return true;

}

}

}

return false;

}