167. Two Sum II - Input array is sorted

Given an array of integers that is already ***sorted in ascending order***, find two numbers such that they add up to a specific target number.

The function twoSum should return indices of the two numbers such that they add up to the target, where index1 must be less than index2.

Note:

- Your returned answers (both index1 and index2) are not zero-based.
- You may assume that each input would have *exactly* one solution and you may not use the *same* element twice.

Example 1:

```
Input: numbers = [2,7,11,15], target = 9
Output: [1,2]
Explanation: The sum of 2 and 7 is 9. Therefore index1 = 1, index2 = 2.
```

Example 2:

```
Input: numbers = [2,3,4], target = 6
Output: [1,3]
```

Example 3:

```
Input: numbers = [-1,0], target = -1
Output: [1,2]
```

Constraints:

- `2 <= nums.length <= 3 * 10^4`
- `-1000 <= nums[i] <= 1000`
- `nums` is sorted in **increasing order**.
- `-1000 <= target <= 1000`

排序数组中找两数之和等于给定值

Two pointers, Array

1. two pointers start from the beginning and end of the array with opposite direction

2. add two numbers `array[beginning] + array[end]`, and compare the sum with given value
3. if the sum is the same, record the positions of two pointers, if the sum is not the same, move either the beginning or the end pointer, loop `beginning < end`
4. return the array with two indexes
