

## 167. Two Sum II - Input array is sorted

Easy

👍 2754

💬 736

❤ Add to List

📄 Share

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

Return *the indices of the two numbers (1-indexed)* as an integer array `answer` of size 2, where `1 <= answer[0] < answer[1] <= numbers.length`.

The tests are generated such that there is **exactly one solution**. 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 <= numbers.length <= 3 * 104`
- `-1000 <= numbers[i] <= 1000`
- `numbers` is sorted in **non-decreasing order**.
- `-1000 <= target <= 1000`
- The tests are generated such that there is **exactly one solution**.

## Related Topics

[Array](#)[Two Pointers](#)[Binary Search](#)

---

## Similar Questions

[Two Sum](#)[Easy](#)[Two Sum IV - Input is a BST](#)[Easy](#)[Two Sum Less Than K](#)[Easy](#)

---