Sorted Two Sum

Given a sorted array of integers and a target, return a pair of indices where the corresponding values in the array add up to the target.  
  
Input: Sorted Array of Integers, Target Integer  
Output: Two element Array of Integers

# Example

Input: [-5, 1, 3, 6, 7], -2 => Output: [0,2]

Input: [1, 9, 10], 8 => Output: [-1,-1]

# Constraints

Time Complexity: O(N)  
Auxiliary Space Complexity: O(1)  
  
May not use a hash to store values because that breaks the space complexity.

# Solution

1. Initiate two indices i = 0, and j = length of array -1
2. While i < j and sum of input[i] + input[j] does not equal to target
   1. If the sum is less than target increment i
   2. If the sum is greater than target decrement j
3. After the while loop, if the sum does not equal to the target
   1. return [-1, -1],
4. Otherwise return [i, j]

# Notes

The sorted nature of the input allows us to perform the algorithm with constant auxiliary space.

# Resources

https://leetcode.com/problems/two-sum-ii-input-array-is-sorted/