Solution 1 Solution 2

Our Solution(s)

```
Run Code
```

Your Solutions

```
Run Code
```

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
 3
   class Program {
        // O(log(n)) time | O(1) space
 4
        func searchForRange(_ array: [Int], _ target: Int) -> [Int] {
            var finalRange = [-1, -1]
            var leftPointer = 0
 9
            var rightPointer = array.count - 1
10
            {\tt alteredBinarySearch(array,\ target,\ \&leftPointer,\ \&rightPointe}
11
12
            leftPointer = 0
14
            rightPointer = array.count - 1
15
            alteredBinarySearch(array, target, &leftPointer, &rightPointer
16
            return finalRange
17
18
        func alteredBinarySearch(_ array: [Int], _ target: Int, _ leftPoin
19
20
            while leftPointer <= rightPointer {</pre>
21
                let middle = (leftPointer + rightPointer) / 2
22
                if array[middle] > target {
23
                    rightPointer = middle - 1
                } else if array[middle] < target {</pre>
26
                    leftPointer = middle + 1
27
                } else {
28
                    if goLeft {
29
                        if middle == 0 || array[middle] != array[middle -
30
                            finalRange[0] = middle
31
32
                        } else {
                            rightPointer = middle - 1
33
```

```
class Program {
    func searchForRange(_ array: [Int], _ target: Int) -> [Int] {
        // Write your code here.
        return []
}
```

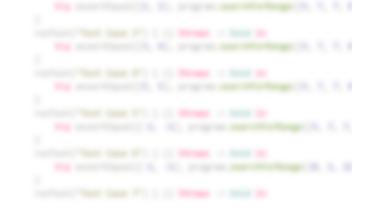
Our Tests

the state of the s

\_\_\_\_

**Custom Output** 

Submit Code



Run or submit code when you're ready.