Our Solution(s) Run Code

Solution 1 Solution 2

```
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

public class Program {
  // O(log(n)) time | O(1) space
  public static int ShiftedBinarySearch(int[] array, int target) {
    return ShiftedBinarySearch(array, target, 0, array.Length - 1);
  }

  public static int ShiftedBinarySearch(int[] array, int target, int
    while (left <= right) {
      int middle = (left + right) / 2;
      int potentialMatch = array[middle];
      int leftNum = array[left];
      int rightNum = array[right];
      if (target == potentialMatch) {
        return middle;
      } else if (leftNum <= potentialMatch) {
        if (target < potentialMatch && target >= leftNum) {
          right = middle - 1;
        } else {
          left = middle + 1;
        }
      } else {
        if (target > potentialMatch && target <= rightNum) {
          left = middle + 1;
        } else {
          right = middle - 1;
        }
      }
    }
    return -1;
  }
}
```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```
public class Program {
  public static int ShiftedBinarySearch(int[] array, int target) {
    // Write your code here.
    return -1;
  }
}

```

Our Tests

Custom Output Submit Code

Run or submit code when you're ready.