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

public class Program {
  // O(log(n)) time | O(log(n)) 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 left, int right) {
    if (left > right) {
      return -1;
    }
    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) {
        return ShiftedBinarySearch(array, target, left, middle - 1);
      } else {
        return ShiftedBinarySearch(array, target, middle + 1, right);
      }
    } else {
      if (target > potentialMatch && target <= rightNum) {
        return ShiftedBinarySearch(array, target, middle + 1, right);
      } else {
        return ShiftedBinarySearch(array, target, left, middle - 1);
      }
    }
  }
}

```