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

#include <vector>
using namespace std;

int shiftedBinarySearch(vector<int> array, int target);
int shiftedBinarySearchHelper(vector<int> array, int target, int left,
                              int right);

// O(log(n)) time | O(log(n)) space
int shiftedBinarySearch(vector<int> array, int target) {
  return shiftedBinarySearchHelper(array, target, 0, array.size() - 1);
}

int shiftedBinarySearchHelper(vector<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 shiftedBinarySearchHelper(array, target, left, middle - 1);
    } else {
      return shiftedBinarySearchHelper(array, target, middle + 1, right);
    }
  } else {
    if (target > potentialMatch && target <= rightNum) {
      return shiftedBinarySearchHelper(array, target, middle + 1, right);
    } else {
      return shiftedBinarySearchHelper(array, target, left, middle - 1);
    }
  }
}

```