```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(1) 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) {
  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;
}
```