

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 public class Program {
4     // O(n) time | O(1) space - where n is the length of the input array
5     public static bool HasSingleCycle(int[] array) {
6         int numElementsVisited = 0;
7         int currentIdx = 0;
8         while (numElementsVisited < array.Length) {
9             if (numElementsVisited > 0 && currentIdx == 0) return false;
10            numElementsVisited++;
11            currentIdx = getNextIdx(currentIdx, array);
12        }
13        return currentIdx == 0;
14    }
15
16    public static int getNextIdx(int currentIdx, int[] array) {
17        int jump = array[currentIdx];
18        int nextIdx = (currentIdx + jump) % array.Length;
19        return nextIdx >= 0 ? nextIdx : nextIdx + array.Length;
20    }
21 }
22
```

Solution 1 Solution 2 Solution 3

```
1 public class Program {
2     public static bool HasSingleCycle(int[] array) {
3         // Write your code here.
4         return false;
5     }
6 }
7
```

Run or submit code when you're ready.
