

# Problem 1538: Guess the Majority in a Hidden Array

## Problem Information

Difficulty: **Medium**

Acceptance Rate: 0.00%

Paid Only: No

## Problem Description

We have an integer array

`nums`

, where all the integers in

`nums`

are

0

or

1

. You will not be given direct access to the array, instead, you will have an

API

`ArrayReader`

which have the following functions:

```
int query(int a, int b, int c, int d)
```

: where

$0 \leq a < b < c < d < \text{ArrayReader.length}()$

. The function returns the distribution of the value of the 4 elements and returns:

4

: if the values of the 4 elements are the same (0 or 1).

2

: if three elements have a value equal to 0 and one element has value equal to 1 or vice versa.

0

: if two elements have a value equal to 0 and two elements have a value equal to 1.

`int length()`

: Returns the size of the array.

You are allowed to call

`query()`

$2 * n$  times

at most where  $n$  is equal to

`ArrayReader.length()`

.

Return

any

index of the most frequent value in

nums

, in case of tie, return -1.

Example 1:

Input:

nums = [0,0,1,0,1,1,1,1]

Output:

5

Explanation:

The following calls to the API reader.length() // returns 8 because there are 8 elements in the hidden array. reader.query(0,1,2,3) // returns 2 this is a query that compares the elements nums[0], nums[1], nums[2], nums[3] // Three elements have a value equal to 0 and one element has value equal to 1 or viceversa. reader.query(4,5,6,7) // returns 4 because nums[4], nums[5], nums[6], nums[7] have the same value. we can infer that the most frequent value is found in the last 4 elements. Index 2, 4, 6, 7 is also a correct answer.

Example 2:

Input:

nums = [0,0,1,1,0]

Output:

0

Example 3:

Input:

```
nums = [1,0,1,0,1,0,1,0]
```

Output:

-1

Constraints:

$5 \leq \text{nums.length} \leq 10$

5

$0 \leq \text{nums}[i] \leq 1$

Follow up:

What is the minimum number of calls needed to find the majority element?

## Code Snippets

**C++:**

```
/**
 * // This is the ArrayReader's API interface.
 * // You should not implement it, or speculate about its implementation
 * class ArrayReader {
 * public:
 * // Compares 4 different elements in the array
 * // return 4 if the values of the 4 elements are the same (0 or 1).
 * // return 2 if three elements have a value equal to 0 and one element has
 * value equal to 1 or vice versa.
 * // return 0 : if two element have a value equal to 0 and two elements have
 * a value equal to 1.
 * int query(int a, int b, int c, int d);
 *
 * // Returns the length of the array
 * int length();
 * };
 */
```

```

class Solution {
public:
    int guessMajority(ArrayReader &reader) {

    }

};

```

## Java:

```

/**
 * // This is the ArrayReader's API interface.
 * // You should not implement it, or speculate about its implementation
 * interface ArrayReader {
 * public:
 * // Compares 4 different elements in the array
 * // return 4 if the values of the 4 elements are the same (0 or 1).
 * // return 2 if three elements have a value equal to 0 and one element has
 * value equal to 1 or vice versa.
 * // return 0 : if two element have a value equal to 0 and two elements have
 * a value equal to 1.
 * public int query(int a, int b, int c, int d);
 *
 * // Returns the length of the array
 * public int length();
 * };
 */

class Solution {
public int guessMajority(ArrayReader reader) {

}

}

```

## Python3:

```

# """
# This is the ArrayReader's API interface.
# You should not implement it, or speculate about its implementation
# """
#class ArrayReader(object):
# # Compares 4 different elements in the array
# # return 4 if the values of the 4 elements are the same (0 or 1).

```

```

# # return 2 if three elements have a value equal to 0 and one element has
value equal to 1 or vice versa.
# # return 0 : if two element have a value equal to 0 and two elements have a
value equal to 1.
# def query(self, a: int, b: int, c: int, d: int) -> int:
#
# # Returns the length of the array
# def length(self) -> int:
#

class Solution:
def guessMajority(self, reader: 'ArrayReader') -> int:

```

## Python:

```

# """
# This is the ArrayReader's API interface.
# You should not implement it, or speculate about its implementation
# """
#class ArrayReader(object):
# # Compares 4 different elements in the array
# # return 4 if the values of the 4 elements are the same (0 or 1).
# # return 2 if three elements have a value equal to 0 and one element has
value equal to 1 or vice versa.
# # return 0 : if two element have a value equal to 0 and two elements have a
value equal to 1.
# def query(self, a, b, c, d):
# """
# :type a, b, c, d: int
# :rtype int
# """
#
# # Returns the length of the array
# def length(self):
# """
# :rtype int
#

class Solution(object):
def guessMajority(self, reader):
"""
:type reader: ArrayReader

```

```
:rtype: integer
"""
```

## JavaScript:

```
/**
 * // This is the ArrayReader's API interface.
 * // You should not implement it, or speculate about its implementation
 * function ArrayReader() {
 * // Compares 4 different elements in the array
 * // return 4 if the values of the 4 elements are the same (0 or 1).
 * // return 2 if three elements have a value equal to 0 and one element has
 * // value equal to 1 or vice versa.
 * // return 0 : if two element have a value equal to 0 and two elements have
 * // a value equal to 1.
 * @param {number} a, b, c, d
 * @return {number}
 * this.query = function(a, b, c, d) {
 * ...
 * };
 *
 * // Returns the length of the array
 * @return {number}
 * this.length = function() {
 * ...
 * };
 * };
 */

/**
 * @param {ArrayReader} reader
 * @return {number}
 */
var guessMajority = function(reader) {

};
```

## TypeScript:

```
/**
 * // This is the ArrayReader's API interface.
 * // You should not implement it, or speculate about its implementation
```

```

* class ArrayReader {
* // Compares 4 different elements in the array
* // return 4 if the values of the 4 elements are the same (0 or 1).
* // return 2 if three elements have a value equal to 0 and one element has
value equal to 1 or vice versa.
* // return 0 : if two element have a value equal to 0 and two elements have
a value equal to 1.
* query(a: number, b: number, c: number, d: number): number { };
*
* // Returns the length of the array
* length(): number { };
* };
*/

function guessMajority(reader: ArrayReader): number {

};

```

## C#:

```

/**
* // This is the ArrayReader's API interface.
* // You should not implement it, or speculate about its implementation
* class ArrayReader {
* public:
* // Compares 4 different elements in the array
* // return 4 if the values of the 4 elements are the same (0 or 1).
* // return 2 if three elements have a value equal to 0 and one element has
value equal to 1 or vice versa.
* // return 0 : if two element have a value equal to 0 and two elements have
a value equal to 1.
* public int Query(int a, int b, int c, int d) {}
*
* // Returns the length of the array
* public int Length() {}
* };
*/

class Solution {
public int GuessMajority(ArrayReader reader) {

}
}

```



```
}
```

**C:**

```
/**
 * *****
 * // This is the ArrayReader's API interface.
 * // You should not implement it, or speculate about its implementation
 * *****
 *
 * // Compares 4 different elements in the array
 * // return 4 if the values of the 4 elements are the same (0 or 1).
 * // return 2 if three elements have a value equal to 0 and one element has
 * // value equal to 1 or vice versa.
 * // return 0 : if two element have a value equal to 0 and two elements have
 * // a value equal to 1.
 * int query(ArrayReader *, int a, int b, int c, int d);
 *
 * // Returns the length of the array
 * int length(ArrayReader *);
 */

int guessMajority(ArrayReader* reader) {

}
```

**Go:**

```
/**
 * // This is the ArrayReader's API interface.
 * // You should not implement it, or speculate about its implementation
 * type ArrayReader struct {
 * }
 *
 * // Compares 4 different elements in the array
 * // return 4 if the values of the 4 elements are the same (0 or 1).
 * // return 2 if three elements have a value equal to 0 and one element has
 * // value equal to 1 or vice versa.
 * // return 0 : if two element have a value equal to 0 and two elements have
 * // a value equal to 1.
 * func (this *ArrayReader) query(a, b, c, d int) int {}
 *
 * // Returns the length of the array
```

```

* func (this *ArrayReader) length() int {}
*/

func guessMajority(reader *ArrayReader) int {

}

```

## Kotlin:

```

/**
 * // This is ArrayReader's API interface.
 * // You should not implement it, or speculate about its implementation
 * interface ArrayReader {
 * // Compares 4 different elements in the array
 * // return 4 if the values of the 4 elements are the same (0 or 1).
 * // return 2 if three elements have a value equal to 0 and one element has
 * // value equal to 1 or vice versa.
 * // return 0 : if two element have a value equal to 0 and two elements have
 * // a value equal to 1.
 * fun query(a: Int, b: Int, c: Int, d: Int): Int {}
 *
 * // Returns the length of the array
 * fun length(): Int {}
 * }
 */

class Solution {
fun guessMajority(reader: ArrayReader): Int {

}

}

```

## Swift:

```

/**
 * // This is ArrayReader's API interface.
 * // You should not implement it, or speculate about its implementation
 * interface ArrayReader {
 * // Compares 4 different elements in the array
 * // return 4 if the values of the 4 elements are the same (0 or 1).
 * // return 2 if three elements have a value equal to 0 and one element has
 * // value equal to 1 or vice versa.

```

```

* // return 0 : if two element have a value equal to 0 and two elements have
a value equal to 1.
* public func query(_ a: Int, _ b: Int, _ c: Int, _ d: Int) -> Int {}
*
* // Returns the length of the array
* public func length() -> Int {}
* }
*/

class Solution {
func guessMajority(_ reader: ArrayReader) -> Int {

}
}

```

## Rust:

```

/**
 * // This is the ArrayReader's API interface.
 * // You should not implement it, or speculate about its implementation
 * struct ArrayReader;
 * impl ArrayReader {
 * // Compares 4 different elements in the array
 * // return 4 if the values of the 4 elements are the same (0 or 1).
 * // return 2 if three elements have a value equal to 0 and one element has
value equal to 1 or vice versa.
 * // return 0 : if two element have a value equal to 0 and two elements have
a value equal to 1.
 * pub fn query(a: i32, b: i32, c: i32, d: i32) -> i32 {}
 *
 * // Returns the length of the array
 * pub fn length() -> i32 {}
 * };
 */

impl Solution {
pub fn get_majority(reader: &ArrayReader) -> i32 {

}
}

```

## Ruby:

```
# This is ArrayReader's API interface.
# You should not implement it, or speculate about its implementation
# class ArrayReader
# # Compares 4 different elements in the array
# # return 4 if the values of the 4 elements are the same (0 or 1).
# # return 2 if three elements have a value equal to 0 and one element has
# # value equal to 1 or vice versa.
# # return 0 : if two element have a value equal to 0 and two elements have a
# # value equal to 1.
# def query(a, b, c, d):
#
#
# end
#
# # Returns the length of the array
# def length()
#
#
# end
# end

# @param {ArrayReader} reader
# @return {int}
def guess_majority(reader)

end
```

## PHP:

```
/**
 * // This is ArrayReader's API interface.
 * // You should not implement it, or speculate about its implementation
 * interface ArrayReader {
 * // Compares 4 different elements in the array
 * // return 4 if the values of the 4 elements are the same (0 or 1).
 * // return 2 if three elements have a value equal to 0 and one element has
 * // value equal to 1 or vice versa.
 * // return 0 : if two element have a value equal to 0 and two elements have
 * // a value equal to 1.
 * function query($a, $b, $c, $d) {}
 *
 * // Returns the length of the array
 * function length() {}
 */
```

```

* }
*/

class Solution {
  /**
   * @param ArrayReader $reader
   * @return Integer
   */
  function guessMajority($reader) {

  }
}

```

## Scala:

```

/**
 * // This is ArrayReader's API interface.
 * // You should not implement it, or speculate about its implementation
 * interface ArrayReader {
 * // Compares 4 different elements in the array
 * // return 4 if the values of the 4 elements are the same (0 or 1).
 * // return 2 if three elements have a value equal to 0 and one element has
 * value equal to 1 or vice versa.
 * // return 0 : if two element have a value equal to 0 and two elements have
 * a value equal to 1.
 * def query(a: Int, b: Int, c: Int, d: Int): Int {}
 *
 * // Returns the length of the array
 * def length(): Int {}
 * }
 */

object Solution {
  def guessMajority(reader: ArrayReader): Int = {

  }
}

```

## Solutions

## C++ Solution:

```
/*
 * Problem: Guess the Majority in a Hidden Array
 * Difficulty: Medium
 * Tags: array, math
 *
 * Approach: Use two pointers or sliding window technique
 * Time Complexity: O(n) or O(n log n)
 * Space Complexity: O(1) to O(n) depending on approach
 */

/**
 * // This is the ArrayReader's API interface.
 * // You should not implement it, or speculate about its implementation
 * class ArrayReader {
 * public:
 * // Compares 4 different elements in the array
 * // return 4 if the values of the 4 elements are the same (0 or 1).
 * // return 2 if three elements have a value equal to 0 and one element has
 * value equal to 1 or vice versa.
 * // return 0 : if two element have a value equal to 0 and two elements have
 * a value equal to 1.
 * int query(int a, int b, int c, int d);
 *
 * // Returns the length of the array
 * int length();
 * };
 */

class Solution {
public:
int guessMajority(ArrayReader &reader) {

}
};
```

## Java Solution:

```
/**
 * Problem: Guess the Majority in a Hidden Array
 * Difficulty: Medium
 * Tags: array, math
```

```

*
* Approach: Use two pointers or sliding window technique
* Time Complexity:  $O(n)$  or  $O(n \log n)$ 
* Space Complexity:  $O(1)$  to  $O(n)$  depending on approach
*/

/**
 * // This is the ArrayReader's API interface.
 * // You should not implement it, or speculate about its implementation
 * interface ArrayReader {
 * public:
 * // Compares 4 different elements in the array
 * // return 4 if the values of the 4 elements are the same (0 or 1).
 * // return 2 if three elements have a value equal to 0 and one element has
 * value equal to 1 or vice versa.
 * // return 0 : if two element have a value equal to 0 and two elements have
 * a value equal to 1.
 * public int query(int a, int b, int c, int d);
 *
 * // Returns the length of the array
 * public int length();
 * };
 */

class Solution {
public int guessMajority(ArrayReader reader) {

}

}

```

## Python3 Solution:

```

"""
Problem: Guess the Majority in a Hidden Array
Difficulty: Medium
Tags: array, math

Approach: Use two pointers or sliding window technique
Time Complexity:  $O(n)$  or  $O(n \log n)$ 
Space Complexity:  $O(1)$  to  $O(n)$  depending on approach
"""

```

```

# """
# This is the ArrayReader's API interface.
# You should not implement it, or speculate about its implementation
# """
#class ArrayReader(object):
# # Compares 4 different elements in the array
# # return 4 if the values of the 4 elements are the same (0 or 1).
# # return 2 if three elements have a value equal to 0 and one element has
# value equal to 1 or vice versa.
# # return 0 : if two element have a value equal to 0 and two elements have a
# value equal to 1.
# def query(self, a: int, b: int, c: int, d: int) -> int:
#
# # Returns the length of the array
# def length(self) -> int:
#
#
class Solution:
def guessMajority(self, reader: 'ArrayReader') -> int:
# TODO: Implement optimized solution
pass

```

## Python Solution:

```

# """
# This is the ArrayReader's API interface.
# You should not implement it, or speculate about its implementation
# """
#class ArrayReader(object):
# # Compares 4 different elements in the array
# # return 4 if the values of the 4 elements are the same (0 or 1).
# # return 2 if three elements have a value equal to 0 and one element has
# value equal to 1 or vice versa.
# # return 0 : if two element have a value equal to 0 and two elements have a
# value equal to 1.
# def query(self, a, b, c, d):
# """
# :type a, b, c, d: int
# :rtype int
# """

```



```

#
# # Returns the length of the array
# def length(self):
# """
# :rtype int
#
class Solution(object):
def guessMajority(self, reader):
"""
:type reader: ArrayReader
:rtype: integer
"""

```

## JavaScript Solution:

```

/**
 * Problem: Guess the Majority in a Hidden Array
 * Difficulty: Medium
 * Tags: array, math
 *
 * Approach: Use two pointers or sliding window technique
 * Time Complexity: O(n) or O(n log n)
 * Space Complexity: O(1) to O(n) depending on approach
 */

/**
 * // This is the ArrayReader's API interface.
 * // You should not implement it, or speculate about its implementation
 * function ArrayReader() {
 * // Compares 4 different elements in the array
 * // return 4 if the values of the 4 elements are the same (0 or 1).
 * // return 2 if three elements have a value equal to 0 and one element has
 * // value equal to 1 or vice versa.
 * // return 0 : if two element have a value equal to 0 and two elements have
 * // a value equal to 1.
 * @param {number} a, b, c, d
 * @return {number}
 * this.query = function(a, b, c, d) {
 * ...
 * };

```

```

*
* // Returns the length of the array
* @return {number}
* this.length = function() {
* ...
* };
* };
*/

/**
* @param {ArrayReader} reader
* @return {number}
*/
var guessMajority = function(reader) {

};

```

## TypeScript Solution:

```

/**
* Problem: Guess the Majority in a Hidden Array
* Difficulty: Medium
* Tags: array, math
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
*/

/**
* // This is the ArrayReader's API interface.
* // You should not implement it, or speculate about its implementation
* class ArrayReader {
* // Compares 4 different elements in the array
* // return 4 if the values of the 4 elements are the same (0 or 1).
* // return 2 if three elements have a value equal to 0 and one element has
value equal to 1 or vice versa.
* // return 0 : if two element have a value equal to 0 and two elements have
a value equal to 1.
* query(a: number, b: number, c: number, d: number): number { };
*

```

```

* // Returns the length of the array
* length(): number { };
* };
*/

function guessMajority(reader: ArrayReader): number {

};

```

## C# Solution:

```

/*
* Problem: Guess the Majority in a Hidden Array
* Difficulty: Medium
* Tags: array, math
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
*/

/**
* // This is the ArrayReader's API interface.
* // You should not implement it, or speculate about its implementation
* class ArrayReader {
* public:
* // Compares 4 different elements in the array
* // return 4 if the values of the 4 elements are the same (0 or 1).
* // return 2 if three elements have a value equal to 0 and one element has
value equal to 1 or vice versa.
* // return 0 : if two element have a value equal to 0 and two elements have
a value equal to 1.
* public int Query(int a, int b, int c, int d) {}
*
* // Returns the length of the array
* public int Length() {}
* };
*/

class Solution {
public int GuessMajority(ArrayReader reader) {

```

```
}  
}
```

## C Solution:

```
/*  
 * Problem: Guess the Majority in a Hidden Array  
 * Difficulty: Medium  
 * Tags: array, math  
 *  
 * Approach: Use two pointers or sliding window technique  
 * Time Complexity: O(n) or O(n log n)  
 * Space Complexity: O(1) to O(n) depending on approach  
 */  
  
/**  
 * *****  
 * // This is the ArrayReader's API interface.  
 * // You should not implement it, or speculate about its implementation  
 * *****  
 *  
 * // Compares 4 different elements in the array  
 * // return 4 if the values of the 4 elements are the same (0 or 1).  
 * // return 2 if three elements have a value equal to 0 and one element has  
 * value equal to 1 or vice versa.  
 * // return 0 : if two element have a value equal to 0 and two elements have  
 * a value equal to 1.  
 * int query(ArrayReader *, int a, int b, int c, int d);  
 *  
 * // Returns the length of the array  
 * int length(ArrayReader *);  
 */  
  
int guessMajority(ArrayReader* reader) {  
  
}
```

## Go Solution:

```

// Problem: Guess the Majority in a Hidden Array
// Difficulty: Medium
// Tags: array, math
//
// Approach: Use two pointers or sliding window technique
// Time Complexity: O(n) or O(n log n)
// Space Complexity: O(1) to O(n) depending on approach

/**
 * // This is the ArrayReader's API interface.
 * // You should not implement it, or speculate about its implementation
 * type ArrayReader struct {
 * }
 * // Compares 4 different elements in the array
 * // return 4 if the values of the 4 elements are the same (0 or 1).
 * // return 2 if three elements have a value equal to 0 and one element has
value equal to 1 or vice versa.
 * // return 0 : if two element have a value equal to 0 and two elements have
a value equal to 1.
 * func (this *ArrayReader) query(a, b, c, d int) int {}
 *
 * // Returns the length of the array
 * func (this *ArrayReader) length() int {}
 */

func guessMajority(reader *ArrayReader) int {

}

```

## Kotlin Solution:

```

/**
 * // This is ArrayReader's API interface.
 * // You should not implement it, or speculate about its implementation
 * interface ArrayReader {
 * // Compares 4 different elements in the array
 * // return 4 if the values of the 4 elements are the same (0 or 1).
 * // return 2 if three elements have a value equal to 0 and one element has
value equal to 1 or vice versa.
 * // return 0 : if two element have a value equal to 0 and two elements have
a value equal to 1.
 * fun query(a: Int, b: Int, c: Int, d: Int): Int {}

```

```

*
* // Returns the length of the array
* fun length(): Int {}
* }
*/

class Solution {
fun guessMajority(reader: ArrayReader): Int {

}
}

```

### Swift Solution:

```

/**
 * // This is ArrayReader's API interface.
 * // You should not implement it, or speculate about its implementation
 * interface ArrayReader {
 * // Compares 4 different elements in the array
 * // return 4 if the values of the 4 elements are the same (0 or 1).
 * // return 2 if three elements have a value equal to 0 and one element has
 * // value equal to 1 or vice versa.
 * // return 0 : if two element have a value equal to 0 and two elements have
 * // a value equal to 1.
 * public func query(_ a: Int, _ b: Int, _ c: Int, _ d: Int) -> Int {}
 *
 * // Returns the length of the array
 * public func length() -> Int {}
 * }
 */

class Solution {
func guessMajority(_ reader: ArrayReader) -> Int {

}
}

```

### Rust Solution:

```

// Problem: Guess the Majority in a Hidden Array
// Difficulty: Medium

```

```

// Tags: array, math
//
// Approach: Use two pointers or sliding window technique
// Time Complexity: O(n) or O(n log n)
// Space Complexity: O(1) to O(n) depending on approach

/**
 * // This is the ArrayReader's API interface.
 * // You should not implement it, or speculate about its implementation
 * struct ArrayReader;
 * impl ArrayReader {
 * // Compares 4 different elements in the array
 * // return 4 if the values of the 4 elements are the same (0 or 1).
 * // return 2 if three elements have a value equal to 0 and one element has
value equal to 1 or vice versa.
 * // return 0 : if two element have a value equal to 0 and two elements have
a value equal to 1.
 * pub fn query(a: i32, b: i32, c: i32, d: i32) -> i32 {}
 *
 * // Returns the length of the array
 * pub fn length() -> i32 {}
 * };
 */

impl Solution {
pub fn get_majority(reader: &ArrayReader) -> i32 {

}
}

```

## Ruby Solution:

```

# This is ArrayReader's API interface.
# You should not implement it, or speculate about its implementation
# class ArrayReader
# # Compares 4 different elements in the array
# # return 4 if the values of the 4 elements are the same (0 or 1).
# # return 2 if three elements have a value equal to 0 and one element has
value equal to 1 or vice versa.
# # return 0 : if two element have a value equal to 0 and two elements have a
value equal to 1.

```

```

# def query(a, b, c, d):
#
# end
#
# # Returns the length of the array
# def length()
#
# end
# end

# @param {ArrayReader} reader
# @return {int}
def guess_majority(reader)

end

```

## PHP Solution:

```

/**
 * // This is ArrayReader's API interface.
 * // You should not implement it, or speculate about its implementation
 * interface ArrayReader {
 * // Compares 4 different elements in the array
 * // return 4 if the values of the 4 elements are the same (0 or 1).
 * // return 2 if three elements have a value equal to 0 and one element has
 * // value equal to 1 or vice versa.
 * // return 0 : if two element have a value equal to 0 and two elements have
 * // a value equal to 1.
 * function query($a, $b, $c, $d) {}
 *
 * // Returns the length of the array
 * function length() {}
 * }
 */

class Solution {
/**
 * @param ArrayReader $reader
 * @return Integer
 */
function guessMajority($reader) {

```



```
}  
}
```

### Scala Solution:

```
/**  
 * // This is ArrayReader's API interface.  
 * // You should not implement it, or speculate about its implementation  
 * interface ArrayReader {  
 * // Compares 4 different elements in the array  
 * // return 4 if the values of the 4 elements are the same (0 or 1).  
 * // return 2 if three elements have a value equal to 0 and one element has  
 * // value equal to 1 or vice versa.  
 * // return 0 : if two element have a value equal to 0 and two elements have  
 * // a value equal to 1.  
 * def query(a: Int, b: Int, c: Int, d: Int): Int {}  
 *  
 * // Returns the length of the array  
 * def length(): Int {}  
 * }  
 */  
  
object Solution {  
  def guessMajority(reader: ArrayReader): Int = {  
  
  }  
}
```