Challenge: Binary Search

Complete the doSearch function so that it implements a binary search, following the pseudo-code below (this pseudo-code was described in the previous article):

1. Let min = 0 and max = n-1.
2. If max < min, then stop: target is not present in array. Return -1.
3. Compute guess as the average of max and min, rounded down (so that it is an integer).
4. If array[guess] equals target, then stop. You found it! Return guess.
5. If the guess was too low, that is, array[guess] < target, then set min = guess + 1.
6. Otherwise, the guess was too high. Set max = guess - 1.
7. Go back to step 2.

Java | Python | C++ | JS

```
import java.util.Arrays;
import java.lang.Integer;

class Solution {
  // Returns either the index of the location in the array,
  // or -1 if the array did not contain the targetValue
  public static int doSearch(int[] array, int targetValue) {
    int min = 0;
        System.out.println(Arrays.toString(array));
    int max = array.length - 1;
    int guess;


        return -1;
  }
};
```