Jolly Jumpers

#10038 - Jolly Jumpers:

Take the absolute difference between a sequence of numbers, ie, [1, 4, 2, 3]:
|(1-4)|=3
|(4-2)|=2
|(2-3)|=1
N=4
Goal: Create the list up to n-1 (aka 3 in this instance) - which we have done here ala 1,2 and 3.
This is considered a “jolly jumper”

- Loop through n -> n-1 to create the set
- Find the difference between each pair (i, i+1…)as we iterate through
- If difference is greater than the last n return false else true

[1, 4, 2, 3]: True
[1, 5, 2, 7]: False

Our approach was to iterate over each number and keep a hashmap of the difference if we encountered any difference out of the range or that previously had a value then return “not jolly”. If none of those cases happen then return “jolly”

Test Case: 1 2
Jolly

Test Case: 3 1 2 3
Not jolly

Problems when Submitting

- Had a continue in a if statement that messed up the count for the scanners

```
import java.util.HashMap;
import java.util.Scanner;

public class Main {
  public static void main(String[] args) {


    Scanner scan = new Scanner(System.in);

    while(scan.hasNextInt()) {

      HashMap<Integer, Boolean> map = new HashMap<>();

      int n = scan.nextInt();
      int[] arr = new int[n];
      for (int k = 0; k < n; k++) {
        arr[k] = scan.nextInt();
      }

      boolean flag = true;


      for(int i = 0; i < n-1; i++) {

        int val = Math.abs(arr[i] - arr[i+1]);
```

```
          // check if the difference is out of range or is not unique
          if (val == 0 || val > (n - 1) || (map.get(val) != null)) {
            flag = false;
          }

          map.put(val, true);

        }

        if(flag) {
          System.out.println("Jolly");
        } else {
          System.out.println("Not jolly");
        }

      }


    }

}
```