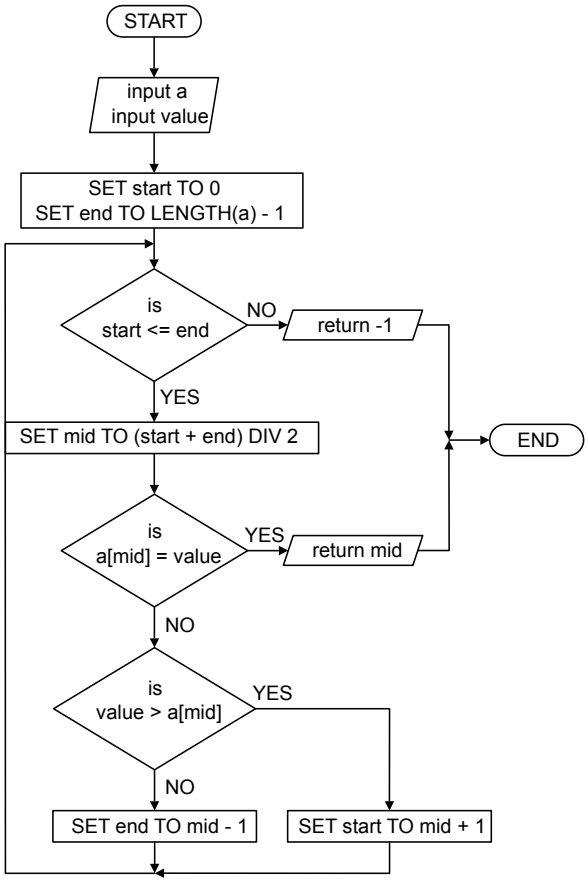
Array Assignment: `binarySearch`

Recall that when we studied flowcharts and pseudocode, we encountered ***binary search***. Recall also that the binary search algorithm uses a strategy of ***divide and conquer*** in order to more efficiently find a value in a list, and that one prerequisite for using the binary search algorithm is that the data is sorted.

A solution to the flowchart for our previous assignment is given to the right, and the Pearson pseudocode representation of that flowchart is given below.




```
FUNCTION binarySearch( a, value )
BEGIN FUNCTION
   SET start TO 0
   SET end TO LENGTH(a) – 1
   WHILE (start <= end) DO
      SET mid TO (start+end) DIV 2
      IF (a[mid] = value) THEN
         RETURN mid
      END IF
      IF (value > a[mid]) THEN
         SET start TO mid + 1
      ELSE
         SET end TO mid – 1
      END IF
   END WHILE
   RETURN –1
END FUNCTION
```

The pseudocode contains a function that takes as inputs the array to search and the value to search for, and returns the index of the location in the array where the value can be found. A value of `-1` (which is not a valid array index) is returned if the value is not found in the array.

1. In an appropriately organized package, create two Java classes: `BinarySearch` and `TestBinarySearch`. Copy the code given for `TestBinarySearch` from below, write the code for the `BinarySearch` class, and test the `BinarySearch` class by running the main method in the `TestBinarySearch` class.

Array Assignment: `binarySearch`

Code for the `TestBinarySearch` class:

```
public class TestBinarySearch {
   public static void main(String[] args) {
      int[] a1 = { 1, 3, 5, 7, 9 };
      int[] a2 = { 2, 4, 6, 8, 10 };
      int value;
      value = BinarySearch.binarySearch(a1, 3);
      System.out.println(value);
      value = BinarySearch.binarySearch(a2, 3);
      System.out.println(value);
      value = BinarySearch.binarySearch(a2, 8);
      System.out.println(value);
      value = BinarySearch.binarySearch(a1, 99);
      System.out.println(value);
   }
}
```