Array Coding 1 – `linearSearch` and `printHistogram`

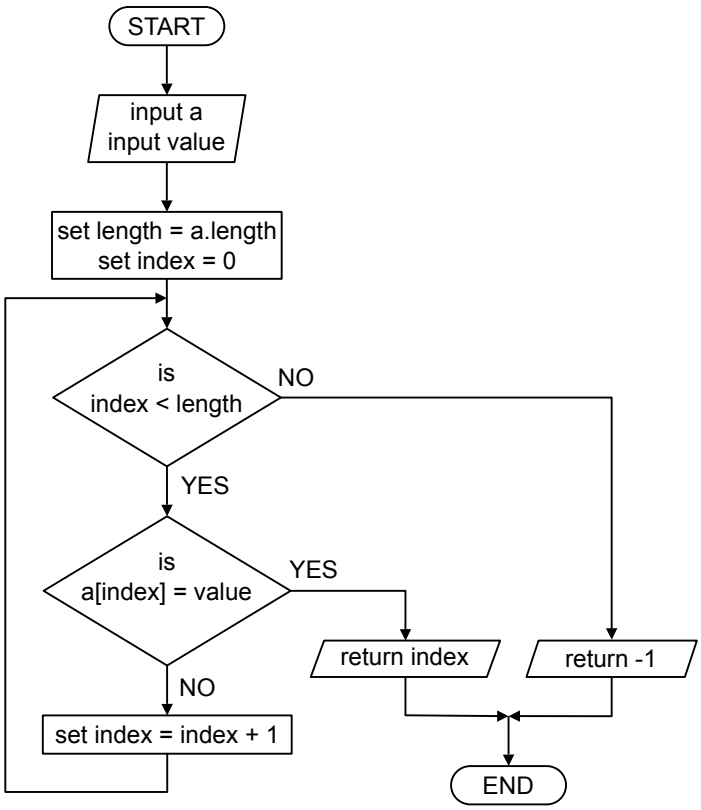
To begin, copy the template code (available online as a "`.java`" file, and at the end of this question paper) into a class named `ArrayAssignment1`. Running this code without modification should produce output as follows:

```
printArray not implemented
printArrayReverse not implemented
printArray not implemented
printArrayReverse not implemented
The value 6 was not found in the array.
The value 8 was not found in the array.
The value 0 was not found in the array.
The value 2 was not found in the array.
The value 1 was not found in the array.

Histogram for array 3:
printHistogram not implemented
```

Modify the specified portions of the template code to complete the following. Although you can add to the main method in order to expand the testing, do not modify the main method in order to obtain the example output correctly. The example output is correct if the methods are implemented according to specification.

1. Write a method named `printArray` that takes an array of integers as a parameter and prints the values stored in the array in set notation – meaning enclosed in curly braces, with the elements separated by commas. Here is an example: `{ 2, 10, 7, 4, 6 }`. Note that there is no comma after the last element in the array.
2. Write a method named `printArrayReverse` that operates similar to the printArray method, except elements are printed in reverse order. For example: `{ 6, 4, 7, 10, 2 }`.
3. To the right is a flowchart representing the algorithm of ***linear search***. This algorithm traverses the given list looking for a specified value, returns the index of the value if it is found, or a value of `-1` if the value is not found. Replace the code given in the method `linearSearch` inside the template code with this algorithm. The method should take two parameters – an array of integers, `a`, and an integer, named `value`, that it will search the array for. The method must use a `for` loop to traverse the array in search of `value`. The method is to return the lowest index of the array that contains an element that is equal to `value`, or `-1` if the array does not contain an element equal to `value`.
4. Implement the same algorithm as in part (3), except modify the algorithm so that the list is traversed in reverse order, finding the highest index of the array that contains an element that is equal to value, or `-1` if the array does not contain an element equal to `value`.



5. Write a method named `printHistogram` that prints a histogram using the values in an array of integers. The method is to take an array of integers as a parameter. It is to first print the array values in set notation. It is to then loop through the array, and for each element of the array, print on a single line a number of asterisks (`*`) equal to the value of the element. For example, an array containing elements `3`, `5` and `2` would display:

```
{ 3, 5, 2 }
***
*****
**
```

After implementing the code specified above, running the test code given in the template will produce the following output.

```
{ 0, 0, 0, 0, 0, 0, 0, 0, 999, 0 }
{ 0, 999, 0, 0, 0, 0, 0, 0, 0, 0 }
{ 2, 10, 7, 4, 6 }
{ 6, 4, 7, 10, 2 }
The value 6 was was found at index 2
The value 8 was not found in the array.
The value 0 was was found at index 9
The value 2 was was found at index 0
The value 1 was not found in the array.

Histogram for array 3:
{ 2, 10, 7, 4, 6 }
**
**********
*******
****
******
```

The template code for this assignment is given on the following pages.

Template Code for this assignment:

```
public class ArrayAssignment1 {
   public static void main(String[] args) {
      int[] a1 = { 2, 4, 6, 7, 10 };
      int[] a2 = new int[10];
      int[] a3 = { 2, 10, 7, 4, 6 };
      a2[8] = 999;
      printArray(a2);
      printArrayReverse(a2);
      printArray(a3);
      printArrayReverse(a3);
      findValue(a1, 6);
      findValue(a1, 8);
      findValueReverse(a2,0);
      findValueReverse(a1,2);
      findValueReverse(a1,1);
      System.out.println("\nHistogram for array 3:");
      printHistogram(a3);
   }
   public static void findValue(int[] a, int value, boolean reverse) {
      int index;
      if(reverse) {
         index = linearSearchReverse(a, value);
      } else {
         index = linearSearch(a, value);
      }
      System.out.print("The value " + value + " was ");
      if(index == -1) {
         System.out.println("not found in the array.");
      } else {
         System.out.println("was found at index " + index);
      }
   }
   public static void findValue(int[] a, int value) {
      findValue(a, value, false);
   }
   public static void findValueReverse(int[] a, int value) {
      findValue(a, value, true);
   }

   // *** CODE CONTINUED ON NEXT PAGE
```

```
   public static void printArray(int[] a) {
      // *** REPLACE THE FOLLOWING LINE OF CODE WITH YOUR CODE ***
      System.out.println("printArray not implemented");
   }
   public static void printArrayReverse(int[] a) {
      // *** REPLACE THE FOLLOWING LINE OF CODE WITH YOUR CODE ***
      System.out.println("printArrayReverse not implemented");
   }
   public static int linearSearch(int[] a, int value) {
      // *** WRITE YOUR CODE FOR linearSearch HERE ***
      return -1;
   }
   public static int linearSearchReverse(int[] a, int value) {
      // *** WRITE YOUR CODE FOR linearSearchReverse HERE ***
      return -1;
   }
   public static void printHistogram(int[] a) {
      // *** REPLACE THE FOLLOWING LINE OF CODE WITH YOUR CODE ***
      System.out.println("printHistogram not implemented");
   }
}
```