

Lab 4 Arrays:

You will write a test plan and a Java program does the following for a one-dimensional array:

1. Declares an array of 10 ints
2. Declares an int to hold the index of the largest array element
3. Declares an int to hold the index of the smallest array element
4. Declares an int to hold the sum of the array elements
5. Declares a double to hold the value of the average of the array elements
6. Defines a method that assigns a random value from 1 to 300 to each array element
7. Defines a method that displays all the elements in the array
8. Defines a method that sorts the array using the insertion technique (the textbook provides sample code for this)
9. Defines a method that identifies and returns the index of the highest value in the array
10. Defines a method that identifies and returns the index of the lowest value in the array
11. Defines a method that calculates the sum of the array elements
12. Defines a method that calculates the average of the array elements
13. Defines a method that displays the results calculated by the methods described in items 5 through 8.

Remember that Java passes arrays **by reference**, so you don't need to return an array. Whatever changes a method makes to an array passed in as a parameter happen to the original array in the method that invokes it.

Please avoid the integer division problem that causes quotients to lose their decimal places.

The main method will:

1. Declare the variables for the index and value of the largest array element, the index and value of the smallest array element, the sum of the elements and the average of the elements
2. Declare the array and pass it to the method that assigns random values to it
3. Pass the array into the method that displays its elements, so we can view their original order
4. Pass it into the method that determines the index of the largest element
5. Pass it into the method that determines the index of the smallest element
6. Pass it into the method that calculates the sum of the elements
7. Pass it into the method that calculates the average of the elements
8. Invoke the method that displays the index and value of the largest array element, the index and value of the smallest array element, the sum of the elements and the average of the elements
9. Pass the array into the sort method
10. Pass the array into the method that displays its elements again, so we can view their sorted order

Your program already has a method for calculating the sum of the array elements. Question: will you re-calculate the sum in the method that obtains the average? No, in the spirit of writing re-usable code, you'll use invoke your previously written sum method.

The method in items #3 and #10 that displays the unsorted and sorted array elements will produce the following output:

(Note that the values are for illustration only. Your array will have different randomly assigned values.)

Before the sorting:

Array Elements

102 97 53 78 45 150 26 10 3 100

After the sorting:

Array Elements

3 10 26 45 53 78 97 100 102 150

The method in item #8 that displays results will produce the following output:

Results of Array Processing

The index of the highest value is 5. Its value is 150.

The index of the lowest value is 8. Its value is 3.

The sum of the array elements is 664.

The average of the array elements is 66.4.

Make sure your output makes sense. For example, do you see what is wrong with this output?

The sum of the array elements is 344.

The average of the array elements is 123.4.

For 25 points Extra Credit:

In the same program, also do all the above for a 2-dimensional array.