3. Arrays

Note 1: all your methods should be `public static` in this module
Note 2: try to use the enhanced for loop as much as possible.
Resources:

Arrays

Problems

1. Write a method that takes an array of integers as its input. The method should print every integer in the array.
2. Write a method that takes an array of integers as its input and returns the sum of every integer in the array.
3. Write a method that calculates the mean of an array.

   Example behavior:

   Input: an array of integers
   Output: the mean of the array (might not be an integer)

4. Write a method that normalizes a vector.

   Example behavior:

   Input: a vector represented as a double array (`double[]`)
   Output: the normalized vector represented as a double array (`double[]`)

5. Write a method that calculates the sum of two vectors.

   Example behavior:

   Input: `double[] a, double[] b` (two vectors)
   Output: the sum of the two vectors (`double[]`)

6. Write a method that replaces every 1 in an integer array with a 0.

   Example behavior:

   Input: `int[]`
   Output: the same integer array, but every 1 should be replaced with a 0

7. Write a method that removes every 1 in an integer array.

   Example behavior:

   Input: `int[]`
   Output: the same integer array, but every 1 should be removed from the array (thus the output array may be shorter than the input array)

8. Write a method that concatenates two arrays.

   Example behavior:

   Input: `int[]a, int[] b`
   Output: all the elements from array a and array b in a single array (`int[]`)

9. Write a method that removes a section of an array.

   Example behavior:

   Input: `int[]a, int startIndex, int endIndex`
   Output: array a, but every element from `startIndex` till `endIndex` should be removed (both `startIndex` and `endIndex` are inclusive)

10. Write the following method that sorts an array (do not use any sorting library, instead try to write the algorithm yourself)

    Example behavior:

    Input: `int[] a`
    Output: the sorted version of array a (the order does not matter)