Exercise 11: Integer Arrays
This assignment asks you to write a collection of basic functions to operate on an array of integers.
It is worth 7 points (1 for each function). Later, you will reuse your same code to build an object oriented version of this assignment.

Step 0:

Write a main() that will eventually call each of the functions in the following steps.
In your main(), declare and initialize two integer arrays with the following values:
`{4,7,9,3,2,8,15,6,8,7}`
`{12,6,4,8,3,7,11,1,6}`
(If you still do not know how to do this, review the book chapter 8 on arrays.)
Use named constants for the two array lengths (10 and 9).
(Named constants are explained in chapter 2.)
For the following functions, call each function twice, once with each of the two arrays.

1. Create a function called printArray that prints out the values of an array in array literal format. The output should look like the initialization values above.

2. Create a function called sumOfArray that takes an int array and returns the sum of the values in the array.

3. Create a function called maxInArray that takes an int array and returns the maximum value in the array.

4. Create a function called minInArray that takes an int array and returns the minimum value in the array.

5. Create a function called rangeInArray that takes an int array and returns the range of the values in the array.

6. Create a function called meanOfArray that takes an int array and returns the average value in the array. (If you know what you are doing, return the average as a double. You may need to look up "cast" to see how to make the division not be an integer divide: `average = sum / (double)length.`)

7. Create a function called clipArray that takes an int array together with a maximum value and changes any value that is higher than the maximum value, to be just the maximum. (This function could also be called "haircut").

Hint1: Except for the printArray function, no other function should use cout. They just return the results. Print the results that are returned where you call the function in main: e.g.

```
cout << "The sum of array1 is " << sumOfArray(array1, ARRAY_SIZE1) << endl;
```

Hint2: When looking for min or max, you need to remember to min or max value that you have seen so far. Do not initialize that value to an arbitrary extreme. Observe that when you check the first value of the array, that should become the min or max you have seen so far. So initialize the minSeen or maxSeen to that value right away, before starting the loop. That way you don't have to know anything about possible extremes.

Hint3: In the average and range functions, use the sum, min, and max functions to compute their parts, rather than repeating the work of computing the sum, or looking for min and max.