Assignment 1: Java Arrays

In this assignment you'll need to build a class that performs a few operations on the array(s) passed to them, namely:

1. Determine if two arrays are equal.
2. Copy a portion of a given array to a new array.
3. Calculate the sum of each of the rows in a 2-dimensional array (or matrix).
4. Calculate the Hamming distance between two arrays.
5. Perform a recursive binary search for a key in a sorted array and return the index of the key (see below for more details). Return -1 if it is not found.

All of these should be implemented as static methods in a single class. Furthermore, you will need to build some form of automated way to test these methods. A separate class can be used to store these tests and have them run in the main method. When using multiple files in a single folder, you don't need to perform any imports between the files to use it.

No builtin/import methods are allowed in this assignment.

Hamming Distance of Two Arrays:

The hamming distance is the number of positions at which the symbols differ when comparing two arrays. For example, these arrays have a hamming distance of 2:

```
int[] a =  {0,1,1,0}, b = {1,1,1,1}; // Positions 0, and 3 are different
```

Here is an example with a hamming distance of 4:

```
int[] c = {1,0,1,0}, d = {0,1,0,1};
```

```
int[] e = {1,2,3,4}, f = {5,6,7,8};
```

Binary Search:

A binary search involves searching though a sorted array by taking the midpoint, checking if the element matches the key, and if not, searching either the left section (if the element is less than the midpoint) or the right section (if the element is greater than the midpoints). Here is a visualization of this.

```
Key = 5, Array = {1,3,5,7,10,11,12};
                          Midpoint, key less than 7, go left
Key = 5, Array = {1,3,5};
                        Midpoint, key greater than 3, go right
Key = 5, Array = {5};
```

```
Found!
```

Bonus (5%):

Allow us to test the hamming distance method with user input alongside automated tests.

You can accept input using:

```
Scanner input = new Scanner(System.in);
```

See the documentation on the Scanner class here:
https://docs.oracle.com/en/java/javase/21/docs/api/java.base/java/util/Scanner.html

Grading Criteria:

Style/submission guidelines: https://gmierzwinski.github.io/bishops/cs321/style_guidelines.html

Comments, Formatting, & Readability	**5 Marks**
Submission Guidelines	**5 Marks**
Parts 1-5	**5 Marks each**
Testing	**5 Marks**
Total	**40 Marks**