CSCI-101 Exam 3

Name _____

Instructions:

Please follow the rules below as you work through this exam.

- Please leave all notebooks and electronics (including cell phones and smart watches) at the side of the room.
- This is a closed book/closed notes exam.
- **Do not spend too much time on any one problem.** You have 50 minutes to complete this exam.
- Partial credit is awarded.
- Please write legibly. If I cannot read your answers, I cannot give you credit.
- Please write your answers **in the order specified**. If you need additional paper, please raise your hand to ask your instructor for additional paper.
- Your code must be written to behave as specified.
- You must properly use all identifiers that are explicitly stated.
- Please use proper and consistent coding conventions (spacing, naming identifiers, etc.).
- Please stay in your seat until you are ready to hand in your exam. You may leave when you are finished.
- Once you leave the testing room you cannot return until the exam is over. If you need to use the restroom, please use it now.

Instructions

Write a class named **Utility** and a class named **Driver**. Provide static methods in the **Utility** class that satisfy the requirements in the section below named Utility. Provide a method named **main** in the **Driver** class that satisfies the requirements in the section below named Driver.

Utility

1. Define a method named **print** that takes an array of integers as an argument and prints to the console the elements in the array on a single line.

2. Define a method named **search** that has an array of Strings as the first parameter and a String as the second parameter. The method returns the number of instances of the String (passed in as the second argument) that exist in the array (passed in as the first argument).

3. Define a method named **sum** that takes 2 integer arrays as arguments. If the arrays that are passed into the method have the same length, then the method returns a new array of the same length as the arrays that are passed into the method. The array that is returned contains, at each index **i**, the sum of the values at index **i** in both of the arrays that are passed into the method. If the arrays have different lengths, then the method returns null.

   For example, if 2 arrays are passed into the method with the values {1, 2, 3, 4} and {1, 2, 3, 4}, then the array that is returned should have the values {2, 4, 6, 8}.

4. Define a method named **getDiagonal** that takes a 2D array of integers as an argument. The method returns null if the rows of the 2D array have different lengths. The method also returns null if the number of rows does not equal the number of columns. If the number of rows does equal the number of columns, then the method returns a new array containing the values of the elements on the diagonal of the 2D array that is passed into the method.

   For example, the method will return an array containing the values {1, 2, 3, 4} if the 2D array that is passed into the method has the following values:

   1 0 0 0
   0 2 0 0
   0 0 3 0
   0 0 0 4

Driver

Ask the user for 3 Strings and add them to an array. Ask the user for another String and store it in a variable. Determine the number of times that the 4th string that the user entered is found in the array by calling the **search** method in the **Utility** class. Print the number of times the string is found in the array.

Ask the user for 8 integers. Add 4 of the integers to one array and add the remaining 4 to a second array. Compute the sum of the arrays by calling the **sum** method in the **Utility** class. Print the values in the array returned by **sum** using the **print** method in the **Utility** class.

Ask the user for 9 integers and populate a 3x3 array with the values entered by the user. Get the values on the diagonal of the 3x3 array by calling **getDiagonal** in the **Utility** class. Print the values in the array returned by **getDiagonal** by calling the **print** method in the **Utility** class.