# Assignment 11: The Lottery Number Matching Service

## Objectives

* Create multiple arrays.
* Input values into those array.
* Use a for-loop over the values of the array.
* Use a nested for-loop over the values of the array.

## Motivation

Imagine a lottery in which in order to win, a ticket must have 5 **non-duplicate** numbers selected from the range of 1 to 50 in the exact order of the winning order drawn from a machine. For example,

* Your ticket: [8, 15, 47, 32, 9]
* Winning Draw: [8, 15, 47, 32, 9]

This is a perfect match because all five numbers match the exact order of the winning draw. Also remember: all numbers on a ticket must be unique!

In this lottery, if you don't have a perfect match in the exact order of the winning draw in order to be a winner. You win some money (but not the jackpot) for having some of the winning numbers and these numbers can be in any order. For example,

* Your ticket: [8, 15, 9, 32, 47]
* Winning Draw: [8, 15, 47, 32, 9]

In this example, it's not a perfect match, but there's 5 matching numbers.

Here's another example.

* Your ticket: [9, 10, 11, 12, 17]
* Winning Draw: [29, 21, 36, 17, 9]

In this example, there's 2 matching numbers. Matching numbers on non-perfect tickets can appear in any order.

## Instructions

Name your project FirstnameLastnameAssignmentNumber

Have your program do the following.

### Part 1. Perfect Match

Create a **public static boolean** method which takes two arrays as arguments. It should be named "isPerfectMatch". The method should return **true** if the two arrays are perfect matches and **false** if otherwise. There is a good discussion on how to write an algorithm to detect if two arrays are perfect matches on page 428 and 429 in the textbook. This will require a for loop.

The preconditions on this method are that both arrays should be 5 elements long. The postconditions are none.

### Part 2. Count Matching Numbers

Create a **public static int** method which takes two arrays as arguments. It should be named "countMatchingNumbers". The method should return the number of elements in the first array which also appear in the second array. This will require a nested for-loop (see pages 221 to 224 in Chapter 4). While this section of the book has nothing to do with arrays, you should be able to combine the discussion of nested for-loops with arrays. Here's the general pseudocode.

count <- 0  
for i from 0 to 4  
 for j from 0 to 4  
 if firstArray[i] equals secondArray[j]  
 count <- count + 1  
 end if  
 end for  
end for  
return count

The preconditions on this method are that both arrays should be 5 elements long. The postconditions are none.

### Part 3. The Main Method

If you have any doubts about what this should look like, please see my example below.

1. Declare a keyboard Scanner object.
2. Declare two integer arrays, each 5 elements long. The first one represents the winning draw. The second represents a hypothetical ticket.
3. Greet the user: "Welcome to Dr. Church's ticket matching software!" (Replace my name with yours.)
4. Tell the user to enter 5 non-duplicate numbers from 1 to 50 representing the winning draw.
5. Read in 5 integers from the keyboard for the winning draw.
6. Tell the user to enter a new set of 5 non-duplicate numbers from 1 to 50 representing a ticket.
7. Read in 5 integers from the keyboard for the ticket.
8. Using your "isPerfectMatch" method, test if the two sequences are a perfect match.
   1. If they are a perfect match, print "These tickets are a perfect match."
   2. If they aren't a perfect match, print "These tickets are not a perfect match." Then print the number of elements that they have in common using the "countMatchingNumbers" method. (See my examples below.)

My code (not counting documentation) was 73 lines. If you go over 100 lines, you should probably contact me or ask a question in the online discussion. Unlike the last assignment, there's only one class file to write.

## Example

Here's a perfect match example.

Welcome to Dr. Church's ticket matching software!  
  
Enter the five numbers from 1 to 50 of the winning draw.  
Enter winning draw number 1: 10  
Enter winning draw number 2: 12  
Enter winning draw number 3: 14  
Enter winning draw number 4: 16  
Enter winning draw number 5: 18  
  
Enter the five numbers from 1 to 50 for a ticket.  
Enter ticket number 1: 10  
Enter ticket number 2: 12  
Enter ticket number 3: 14  
Enter ticket number 4: 16  
Enter ticket number 5: 18  
  
These tickets are a perfect match.

Here's another example, but with a different ordering to the numbers.

Welcome to Dr. Church's ticket matching software!  
  
Enter the five numbers from 1 to 50 of the winning draw.  
Enter winning draw number 1: 10  
Enter winning draw number 2: 12  
Enter winning draw number 3: 14  
Enter winning draw number 4: 16  
Enter winning draw number 5: 18  
  
Enter the five numbers from 1 to 50 for a ticket.  
Enter ticket number 1: 14  
Enter ticket number 2: 12  
Enter ticket number 3: 18  
Enter ticket number 4: 10  
Enter ticket number 5: 16  
  
These tickets aren't a perfect match. They have 5 number(s) in common.

Here's the example with only two matching numbers.

Welcome to Dr. Church's ticket matching software!  
  
Enter the five numbers from 1 to 50 of the winning draw.  
Enter winning draw number 1: 9  
Enter winning draw number 2: 10  
Enter winning draw number 3: 11  
Enter winning draw number 4: 12  
Enter winning draw number 5: 17  
  
Enter the five numbers from 1 to 50 for a ticket.  
Enter ticket number 1: 29  
Enter ticket number 2: 21  
Enter ticket number 3: 36  
Enter ticket number 4: 17  
Enter ticket number 5: 9  
  
These tickets aren't a perfect match. They have 2 number(s) in common.

Here's an example with no matching numbers.

Welcome to Dr. Church's ticket matching software!  
  
Enter the five numbers from 1 to 50 of the winning draw.  
Enter winning draw number 1: 5  
Enter winning draw number 2: 10  
Enter winning draw number 3: 15  
Enter winning draw number 4: 20  
Enter winning draw number 5: 25  
  
Enter the five numbers from 1 to 50 for a ticket.  
Enter ticket number 1: 6  
Enter ticket number 2: 11  
Enter ticket number 3: 16  
Enter ticket number 4: 21  
Enter ticket number 5: 26  
  
These tickets aren't a perfect match. They have 0 number(s) in common.

## Documentation

Your source code must include the following documentation:

* Your name
* The class (CS 2070) and the section number (on ground is 08, online is W1).
* The date on which you turned in the assignment.
* A short description of the software. Usually a sentence or two is sufficient.

## Turning it in.

To turn in your application, find the folder containing your entire project (not the folder with the "java" file), zip it up, and turn it in.