Name ____ Period ______

Latin Square

Your Tasks (Mark these off as you go)

- ☐ Define key vocabulary
- ☐ Get acquainted with Latin Squares
- ☐ Write the containsDuplicates method
- ☐ Write the getColumn method
- ☐ Write the hasAllValues method
- ☐ Write the isLatin method
- ☐ Receive credit for this lab guide

☐ Define key vocabulary

2D Array

☐ Get acquainted with Latin Squares

A two-dimensional square array of integers is a Latin square if the following conditions are true,

- The first row has no duplicate values
- All values in the first row of the square appear in each row of the square
- All values in the first row of the square appear n each column of the square

Examples of Latin Squares

1	2	3
2	3	1
3	1	2

10	30	20	0
0	20	30	10
30	0	10	20
20	10	0	30

Examples that are NOT Latin Squares

1	2	1
2	1	1
1	1	2

Not a Latin square because the first row contains duplicate values

1	2	3
3	1	2
7	8	9

Not a Latin square because the elements of the first row do not all appear in the third row

1	2
1	2

Not a Latin square because the elements of the first row do not all appear in either column

Consider the grid below, fill in the grid below to create a Latin Square

		cols				
		0	1	2	3	4
rows	0	0	1	2	3	4
	1					
	2					
	3					
	4					

Consider the 2D grid above which represents a Latin Square. Write code that could be used to declare and initialize the square.

```
int[ ][ ] latinSquare = new int[5][5];
```

Write code that could be used to populate the square using the values you have indicated.

```
latinSquare[0] = {0, 1, 2, 3, 4}
latinSquare[1] = {4, 0, 1, 2, 3}
latinSquare[2] = {3, 4, 0, 1, 2}
etc...
```

Given the values in row = 0 are defined and that there are no duplicates, think of an algorithm you could use to populate any grid to ensure it results in a Latin square. In your algorithm reference the locations in the grid in terms of rows and cols. Write your algorithm below.

```
 or(int row = 0; row < latinSquare.length; row++}
    for(int col = 0; col < latinSquare[row].length; col++){
            latinSquare[row][col] = latinSquare[0][(col + row) % latinSquare.length];
    }
}
```

☐ Write the containsDuplicates method

The first criteria of a Latin Square is that none of the rows or columns in the square contain duplicate values. The containsDuplicate method accepts an array as a parameter and checks the array for duplicate values. If the array has duplicate values it will return true otherwise it will return false. Consider the examples below,

Returns true	Returns false
6 | 0 | 1 | 9 | 8 | 0 | 7	6 | 5 | 1 | 9 | 8 | 0 | 7

The contains duplicates has the following signature

```
public boolean containsDuplicates(int[] arr){}
```

Write the containsDuplicates method in the space below.

```
for(int i = 0; i < arr.length; i++){
    int val = arr[i];
    for(int j = 0; j < arr.length; j++){
      if(val == arr[j] && i != j){
          return true;
      }
}
return false;
```

☐ Write the getColumns method

Another condition of a Latin Square is that all values in the first row of the square appear in each row of the square. And, all values of the in the first row of the square also appear in each column. Before we can check whether or not each column has the same values as the first row, we must first be able to retrieve a column from our Latin Square.

The getColumn method should accept an integer which represents the column in the grid and has the following signature,

```
public int[] getColumn(int arr2D[][], int col){}
```

Write the getColumn method in the space below

```
int temp[ ] = new int[arr2D.length];

for(int rows = 0; rows < arr2D.length; rows++){
      temp[rows] = arr2D[rows][col];
}

return temp;
```

☐ Write the hasAllValues method

Now that we have a way to retrieve our columns from the Latin Square, we can check to see if each of the columns has all the values as the first row. Recall, that to be a Latin square the values that appear in the first row show also appear in every other row too.

The has AllValues method compares two arrays to see if they each have the same values. If they do, it returns true, otherwise it returns false. The hasAllValues method has the following signature,

```
public boolean hasAllValues(int arr1[], int arr2[]){}
```

Write the hasAllValues method in the space below

```
for(int i = 0; i < arr1.length; i++){
     boolean found = false;
     for(int j = 0; j < arr2.length; j++){
          if(arr1[i] == arr2[j]){
               found = true;
          }
     }
     if(!found){
          return false;
     }
}
return true;
```

☐ Write the isLatin method

Now that we have all the necessary helper methods written we can check to see if any square is a Latin Square. The isLatin method should check for the following,

- The first row has no duplicate values
- All values in the first row of the square appear in each row of the square
- All values in the first row of the square appear n each column of the square

The signature for the isLatin method is shown below. If a square meets the above criteria, isLatin returns true, otherwise it returns false.

```
public boolean isLatin(int square[][]){}
```

In the space below, write the isLatin method.

```
if(containsDuplicates(square[0]){
    return false;
}
for(int i = 0; i < square.length; i++){
    if(!containsAllValues(square[0], getColumn(square, i)){
        return false;
    }
    if(!containsAllValues(square[0], square[i]){
        return false;
    }
}
return true;
```

☐ Receive credit for this lab guide

Submit this portion of the lab to Pluska to receive credit for the lab guide. Once received, your completed code challenges will also be graded and will count towards your final lab grade.