CSE 121 Section 15: 2D Arrays

This worksheet will walk through solving the problem below. Start by carefully reading the specifications.

Write a method named increasingColumns that has one parameter, a 2D integer array. It should modify this 2D array so that all of its "columns" are in increasing order. The method should not return anything.

In particular, items are in an increasing order when each item is strictly greater than the items preceding it. When an element within a column is **not** in increasing order, replace its value with the previous element's value, and then add one. You may assume that all arrays in the problem (both the 2D array and its sub-arrays) are not empty or null. In addition, you may assume that each sub-array has the same length.

For example, let's say we have the 2D array **arr**, which is initialized as:

```
{{1, 2, 3, 4},

{1, 5, 6, 7},

{1, 8, 9, 10}}.
```

The first column has values that are all 1. After executing *increasingColumns*, **arr** should contain:

```
{{1, 2, 3, 4},

{2, 5, 6, 7},

{3, 8, 9, 10}}.
```

Step 1: Write a method header for this problem. In particular, what are the parameters and returns?

```
public static void increasingColumns(int[][] arr)
```

Step 2: Write some preliminary pseudocode for your method.
Consider:

- What method will we use to traverse the elements of the 2D array?
    - One approach could have the outer loop represent the “rows”, and the inner loop represent the “columns”.
    - Another approach could have the outer loop represent “columns” and the inner loop represent the “rows”.
    - Do either of these approaches feel natural? Is there a different approach you’d prefer instead?
    - What kinds of comparisons and math do we need to do? How frequently will we do them? Do we need to store any temporary variables?
    - What is the condition for changing a value in our array?

```
For each column
    For each row of the column
        If non-increasing compared to previous -> update value to previous + 1

(Answers vary)
```

Step 3: Now, implement increasingColumns. We've supplied a sample class and main method.

```
import java.util.*;

public class IncreasingColumns {
    public static void main(String[] args) {
        int[][] arr1 = {{1, 4, 6, 7},
                        {3, 2, 0, 3},
                        {7, 2, 8, 8}};

        int[][] arr2 = {{5, 2, 3},
                        {5, 2, 3},
                        {9, -1, 5}};

        int[][] arr3 = {{3},
                        {6},
                        {-5}};

        increasingColumns(arr1);
        increasingColumns(arr2);
        increasingColumns(arr3);

        System.out.println(Arrays.deepToString(arr1));
        System.out.println(Arrays.deepToString(arr2));
        System.out.println(Arrays.deepToString(arr3));
    }

    // TODO: Write your code here!
    // Full solution on next page
```

```
    // Modifies the given int[][] arr such that each column is
    // in increasing order.
    // If an element is not in increasing order, it will be
    // changed to one more
    // than the previous value in it's column.
    public static void increasingColumns(int[][] arr) {
        for (int col = 0; col < arr[0].length; col++) {
            int previousValue = arr[0][col] - 1;
            for (int row = 0; row < arr.length; row++) {
                if (arr[row][col] <= previousValue) {
                    arr[row][col] = previousValue + 1;
                }
                previousValue = arr[row][col];
            }
        }
    }
```

Step 4: Now, we will write an additional test for your method. First, write a sample input 2D array. Then, write down what the expected output **should** be.

Answers vary

Step 5: Test your code manually by tracing through your test case (e.g., writing them down on paper and going through them). Did it match your expected output? If not, identify the issue and edit your code to fix this issue.

Answers Vary

As a note, this problem also serves as good practice for the final exam!
Reflect on your work to prepare for the exam; did you find writing the method header and pseudocode helpful? Did it speed up your implementation? How did testing your code help you verify your solution? ***Try to identify what practices make you successful at writing code on paper!***