This question involves a two-dimensional array of integers that represents a collection of randomly generated data. A partial declaration of the `Data` class is shown. You will write two methods of the `Data` class.

```
public class Data
{
      public static final int MAX = / * value not shown * / ;
      private int[][] grid;

      // Fills all elements of grid with randomly generated values
      // PreCondition: grid is not null and has at least one element
      public void repopulate()
      { /*  to be implemented in part (a)  */ }

// Returns the number of columns in grid that are in increasing order
      // PreCondition: grid is not null and has at least one element
      public int countIncreasingCols()
      { /* to be included in part (b)  */ }

      // There may be instance variables, constructors, and methods that are not
shown.
      }
```

a. Write the `repopulate` method, which assigns a newly generated random value to each element of `grid`.

Each value is computed to meet all of the following criteria, and all valid values must have an equal chance of being generated.

- The value is between 1 and `MAX`, inclusive.
- The value is divisible by 10.
- The value is not divisible by 100.

Complete the `repopulate` method.

// Fills all elements of `grid` with randomly generated values
// PreCondition: `grid` is not `null` and has at least one element

b. Write the `countIncreasingCols` method, which returns the number of columns in `grid` that are in increasing order. A column is considered in increasing order if the elements in each row after the first row is greater than or equal to the elements in the previous row. A column with only one row is considered to be in increasing order.

The following examples show the `countIncreasingCols` return values for possible contents of `grid`.

The return value for the following contents of `grid` is 1, since the first column is in increasing order but the second and third columns are not.

10	50	40
20	40	20
30	50	30

The return value for the following contents of `grid` is 2, since the first and third columns are in increasing order but the second and fourth columns are not.

10	540	440	440
220	450	440	190

Complete the `countIncreasingCols` method.

```
// Returns the number of columns in grid that are in increasing order
// PreCondition: grid is not null and has at least one element
public int countIncreasingCols()
{
```