

Two Dimensional Array worksheet #2

1. Write a code segment that computes the averages of the values in the columns of the array `scores` and stores them in the parallel array `colAves`. Modify the code segment from a previous worksheet which computes the row averages. However, you do not need to compute or display the average of **all** the elements in `scores`.

```
final int NUM_ROWS = 3;
final int NUM_COLS = 2;
int[][] scores = {{100, 200}, {200, 300}, {100, 100}};

int colSum = 0;
double[] colAves = new double[2];
```

2. Write a code segment that finds & displays the ***smallest even number*** in the two-dimensional array `numbers`. You can assume that `numbers` is full of integers that are both odd and even. Your code should automatically account for the number of rows and columns in `numbers`. Declare & initialize any necessary variables other than `numbers`.

3. On the back of this paper, write a static method named `biggestRow` that is passed a two-dimensional array of integers named `numbers`. The method must return the row number that has the greatest sum of its elements. For example, the value 2 would be returned if `numbers` is

12	3	5
1	-8	9
20	15	9
0	18	-2

because row 2 has a sum of 44 which is greater than the sum of any other rows. The method must work for a two-dimensional array of any size.

```
public static int biggestRow(int[][] numbers)
{
```