Section, Name and ID#

Problem 2: Write a Java method ***public static double[] mean(double[] data)*** that takes as its argument an array of data points ***double[] data***, and returns a two-element array – the first element being the mean value of the data points and the second element being the standard deviation. The standard deviation σ of n numbers a_i is computed as:

$$\sigma = \sqrt{\frac{\sum_{i=0}^{n-1}(a_i - mean)^2}{n}}$$

~~public static double[] mean(double[] data)~~

~~{~~

I will write a program, which will output the following table:

A	B	C	D	E	F
5	4	3	2	1	0

```
public static void main(String[] args)
{ char[] letters = {'A', 'B', 'C', 'D', 'E', 'F'};
  int[] nums = {5, 4, 3, 2, 1};

  for(int i=0; i<letters.length; i++) {
      system.out.print(letters[i] + "  ");
      }
      system.out.println();
  for(int n=0; n<nums.length; n++){
      System.out.print(nums[n] + "  ");
      }
}
```

2

Use the backside, if needed

OOP. MT. 170317. L075

ection, Name and ID#,

Problem 4: Implement the following Java methods that swap element values between two 2D integer arrays of the same size ***int[][] a*** and ***int[][] b***:

1. ***public static void swap(int[][] a, int[][] b, int row, int col)*** – swaps element values from the specified row ***int row*** and column ***int col***;
2. ***public static void swapCol(int[][] a, int[][] b, int col)*** – swaps all element values from the specified column ***int col***;
3. ***public static void swapRow(int[][] a, int[][] b, int row)*** – swaps all element values from the specified row ***int row***. Get s bonus, if ***swapRow()*** performs faster than ***swapCol()***.

```
public static void swap(int[] a, in[][] b, int row, int col){
    for(col=0; col< b.length; col++){
        System.out.print(col b[col]+ "  ");
        for(row=0; row< b[col]; row++){
            System.out.println(b[col][row]+ "  ");
    }
}
```

```
public static void swap(int[] a, int[] b, int row, int col){
for(col=0; col< b.length; col++){
    for(row=0; row< b[col]; row++){
        System.out.println(b[col][row]+ "  ");
    }
}
```

1

Use the backside, if needed

Problem 4 of 4

OOP.MT.170317.LOGS