Quiz2

Week 1

Thursday

Name: _____

Instructions:

1. **Wait to start until the TA tells you to do so. Do not turn the page yet.**
2. You will have forty minutes to complete up to two of the two questions. We will grade both questions and you will keep the higher score of the two.
3. Please turn in this quiz sheet when finished. You may either sit quietly at your desk (no laptops), or, preferably, leave the room and return when the quiz is over for the class.
4. The maximum possible score for Problem 1 (the problem you have seen) is 80 points; the maximum score of Problem 2 (the unseen problem) is 100 points.

Problem 1: Share Point (max score of 80 points)

Imagine that the user specifies with width and height of a grid, and provides two tiles in that grid. You will write code to determine if the two tiles share only a point. For example, in the following 3x3 gird,

1	2	3
4	5	6
7	8	9

tiles 1 and 5 share a single point, but 1 and 4 do not (they share an edge). In this case, your code would return "point" for tiles 1 and 5, and "not point" for tiles 1 and 4.

You may use the following formulas in your solution as needed:

```
row = (tile - 1) / width
```

Please do not use any Java code or libraries that we have not covered in class; this defeats the purpose of the assessment.

See the test cases below, and then complete the template.

```
/*
            SAMPLE ASSESSMENT2_2
*/

public class Assess2_2_Sample{


  public static void main(String[] args){
    System.out.println("test1: " + checkPoint(1,1,1,1).equals("not point"));
    System.out.println("test2: " + checkPoint(1,2,1,1).equals("not point"));
    System.out.println("test3: " + checkPoint(1,2,1,2).equals("not point"));
    System.out.println("test4: " + checkPoint(1,2,2,1).equals("not point"));
    System.out.println("test5: " + checkPoint(1,2,2,2).equals("not point"));
    System.out.println("test6: " + checkPoint(2,1,1,1).equals("not point"));
    System.out.println("test7: " + checkPoint(2,1,1,2).equals("not point"));
    System.out.println("test8: " + checkPoint(2,1,2,1).equals("not point"));
    System.out.println("test9: " + checkPoint(2,1,2,2).equals("not point"));
    System.out.println("test10: " + checkPoint(2,2,1,2).equals("not point"));
    System.out.println("test11: " + checkPoint(2,2,1,3).equals("not point"));
    System.out.println("test12: " + checkPoint(2,2,1,4).equals("point"));
    System.out.println("test13: " + checkPoint(2,2,2,1).equals("not point"));
    System.out.println("test14: " + checkPoint(2,2,2,3).equals("point"));
    System.out.println("test15: " + checkPoint(2,2,2,4).equals("not point"));
```

```
  System.out.println("test16: " + checkPoint(2,2,3,1).equals("not point"));
  System.out.println("test17: " + checkPoint(2,2,3,2).equals("point"));
  System.out.println("test18: " + checkPoint(2,2,3,4).equals("not point"));
  System.out.println("test19: " + checkPoint(2,2,4,1).equals("point"));
  System.out.println("test20: " + checkPoint(2,2,4,2).equals("not point"));
  System.out.println("test21: " + checkPoint(2,2,4,3).equals("not point"));
  System.out.println("test22: " + checkPoint(2,3,1,2).equals("not point"));
  System.out.println("test23: " + checkPoint(2,3,1,3).equals("not point"));
  System.out.println("test24: " + checkPoint(2,3,1,4).equals("point"));
  System.out.println("test25: " + checkPoint(2,3,1,5).equals("not point"));
  System.out.println("test26: " + checkPoint(2,3,1,6).equals("not point"));
  System.out.println("test27: " + checkPoint(2,3,2,1).equals("not point"));
  System.out.println("test28: " + checkPoint(2,3,2,3).equals("point"));
  System.out.println("test29: " + checkPoint(2,3,2,4).equals("not point"));
  System.out.println("test30: " + checkPoint(2,3,2,5).equals("not point"));
  System.out.println("test31: " + checkPoint(2,3,2,6).equals("not point"));
  System.out.println("test32: " + checkPoint(2,3,3,4).equals("not point"));
  System.out.println("test33: " + checkPoint(2,3,3,5).equals("not point"));
  System.out.println("test34: " + checkPoint(2,3,3,6).equals("point"));
  System.out.println("test35: " + checkPoint(2,3,4,5).equals("point"));
  System.out.println("test36: " + checkPoint(2,3,4,6).equals("not point"));
  System.out.println("test37: " + checkPoint(2,3,5,6).equals("not point"));
  System.out.println("test38: " + checkPoint(3,2,1,2).equals("not point"));
  System.out.println("test39: " + checkPoint(3,2,1,3).equals("not point"));
  System.out.println("test40: " + checkPoint(3,2,1,4).equals("not point"));
  System.out.println("test41: " + checkPoint(3,2,1,5).equals("point"));
  System.out.println("test42: " + checkPoint(3,2,1,6).equals("not point"));
  System.out.println("test43: " + checkPoint(3,2,2,1).equals("not point"));
  System.out.println("test44: " + checkPoint(3,2,2,3).equals("not point"));
  System.out.println("test45: " + checkPoint(3,2,2,4).equals("point"));
  System.out.println("test46: " + checkPoint(3,2,2,5).equals("not point"));
  System.out.println("test47: " + checkPoint(3,2,2,6).equals("point"));
  System.out.println("test48: " + checkPoint(3,2,3,4).equals("not point"));
  System.out.println("test49: " + checkPoint(3,2,3,5).equals("point"));
  System.out.println("test50: " + checkPoint(3,2,3,6).equals("not point"));
  System.out.println("test51: " + checkPoint(3,2,4,5).equals("not point"));
  System.out.println("test52: " + checkPoint(3,2,4,6).equals("not point"));
  System.out.println("test53: " + checkPoint(2,3,5,6).equals("not point"));

}
```

```
public static String checkPoint(int width, int height, int tile1, int tile2){

}
```

```
//extra sheet if needed

  }
}
```

Problem 2: Sum Top Corner (max score of 100 points)

Write code that will sum up the six tiles at the top corners of a grid and return that sum, if it is even, otherwise, return -1 if the sum is odd. For example, in the 7x10 grid below, the top corners are tiles 1 and 7, so you would sum together [1 + 2 + 8] + [7 + 6 + 14] for a total sum of 38, and you would return 38. For a 4x4 board where the sum is 23, you would return -1.

We will guarantee that the board will always be large enough to have six unique values for these tiles (so the board will never be smaller than a width of four and will never be smaller than a height of 2).

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	32	33	34	35
36	37	38	39	40	41	42
43	44	45	46	47	48	49
50	51	52	53	54	55	56
57	58	59	60	61	62	63
64	65	66	67	68	69	70

Here are some example test cases for a function defined as:

```
sumTopCorner(int width, int height)
```

```
sumTopCorner (4, 4) will return -1, (because 1 + 2 + 5 + 4 + 3 + 8
sums to 23)

sumTopCorner (4, 5) will also return -1 (same logic as above)

sumTopCorner (4, 4) will return 38, (because 1 + 2 + 8 + 6 + 7 + 14
sums to 38)
```

Complete the template below. Please do not use any Java code or libraries that we have not covered in class; this defeats the purpose of the assessment.

```
public static int sumTopCorner(int width, int height){

        // YOUR ANSWER HERE
```

```
// extra page if needed

}
```

SCRATCH PAPER

SCRATCH PAPER