Quiz2

Week 1

Thursday

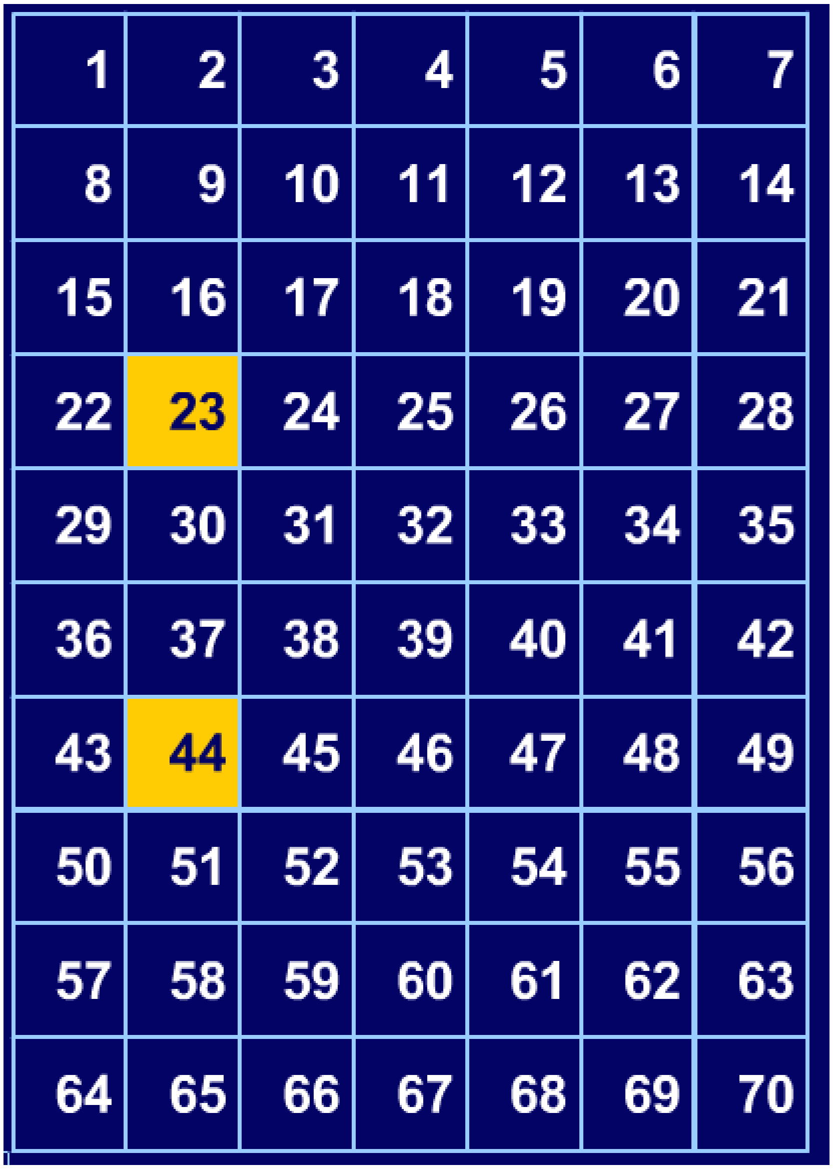
Name: ____

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: Sum Neighbor Plus (max score of 80 points)

Imagine that the user specifies with width and height of a grid, and provides a tile in that grid. You will write code to return the sum of the adjacent neighbors of that tile that are directly above, or to the left and right. For example, in the grid below,



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

tiles 23 will would return the list [16, 30, 22, 24] which sums to 92 and tile 70 would return the list [63, 0, 69, 0], which sums to 132. If a neighbor doesn't exist, you should return a 0 for that item in your result array before summing.

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_8
*/
import java.util.Arrays;

public class Assess2_8_Sample{

  public static void main(String[] args){
    System.out.println("test1: " + (plusSign(1,1,1) == 0));
    System.out.println("test2: " + (plusSign(2,1,1) == 2));
    System.out.println("test3: " + (plusSign(2,1,2) == 1));
    System.out.println("test4: " + (plusSign(1,2,1) == 2));
    System.out.println("test5: " + (plusSign(1,2,2) == 1));
    System.out.println("test6: " + (plusSign(1,3,1) == 2));
    System.out.println("test7: " + (plusSign(1,3,2) == 4));
    System.out.println("test8: " + (plusSign(1,3,3) == 2));
    System.out.println("test9: " + (plusSign(3,1,1) == 2));
    System.out.println("test10: " + (plusSign(3,1,2) == 4));
    System.out.println("test11: " + (plusSign(3,1,3) == 2));
    System.out.println("test12: " + (plusSign(2,3,1) == 5));
    System.out.println("test13: " + (plusSign(2,3,2) == 5));
    System.out.println("test14: " + (plusSign(2,3,3) == 10));
    System.out.println("test15: " + (plusSign(2,3,4) == 11));
    System.out.println("test16: " + (plusSign(2,3,5) == 9));
    System.out.println("test17: " + (plusSign(3,2,1) == 6));
    System.out.println("test18: " + (plusSign(3,2,2) == 9));
    System.out.println("test19: " + (plusSign(3,2,3) == 8));
    System.out.println("test20: " + (plusSign(3,2,4) == 6));
    System.out.println("test21: " + (plusSign(3,2,5) == 12));
    System.out.println("test22: " + (plusSign(3,2,6) == 8));
    System.out.println("test23: " + (plusSign(3,3,1) == 6));
    System.out.println("test24: " + (plusSign(3,3,2) == 9));
    System.out.println("test25: " + (plusSign(3,3,3) == 8));
    System.out.println("test26: " + (plusSign(3,3,4) == 13));
    System.out.println("test27: " + (plusSign(3,3,5) == 20));
    System.out.println("test28: " + (plusSign(3,3,6) == 17));
    System.out.println("test29: " + (plusSign(3,3,7) == 12));
    System.out.println("test30: " + (plusSign(3,3,8) == 21));
    System.out.println("test31: " + (plusSign(3,3,9) == 14));
    System.out.println("test32: " + (plusSign(5,5,2) == 11));
    System.out.println("test33: " + (plusSign(5,5,19) == 76));
    System.out.println("test34: " + (plusSign(5,5,13) == 52));
    System.out.println("test35: " + (plusSign(5,5,25) == 44));

  }
```

```
public static int plusSign(int width, int height, int tile){
  //YOUR ANSWER BELOW
//extra sheet if needed

  }
}
```

Problem 2: Top Right Quadrant (max score of 100 points)

Write code that will return one of the two integers:

1) If the board width and height are not both even, return -1
2) Otherwise, return 1 if a tile is in the top right quadrant of a board, or 0 if it is not the case.

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

```
public static int checkQuad(int width, int height, int tile)
```

```
checkQuad(3, 4, 2) will return -1

checkQuad(3, 5, 2) will return -1

checkQuad(3, 3, 2) will return -1

checkQuad(4, 5, 2) will return -1

checkQuad(6, 8, 4) will return 1

checkQuad(6, 8, 9) will return 0

checkQuad(6, 8, 11) will return 1
```

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 checkQuad(int width, int height, int tile)

        // YOUR ANSWER HERE
```

```
// extra page if needed

}
```

SCRATCH PAPER

SCRATCH PAPER