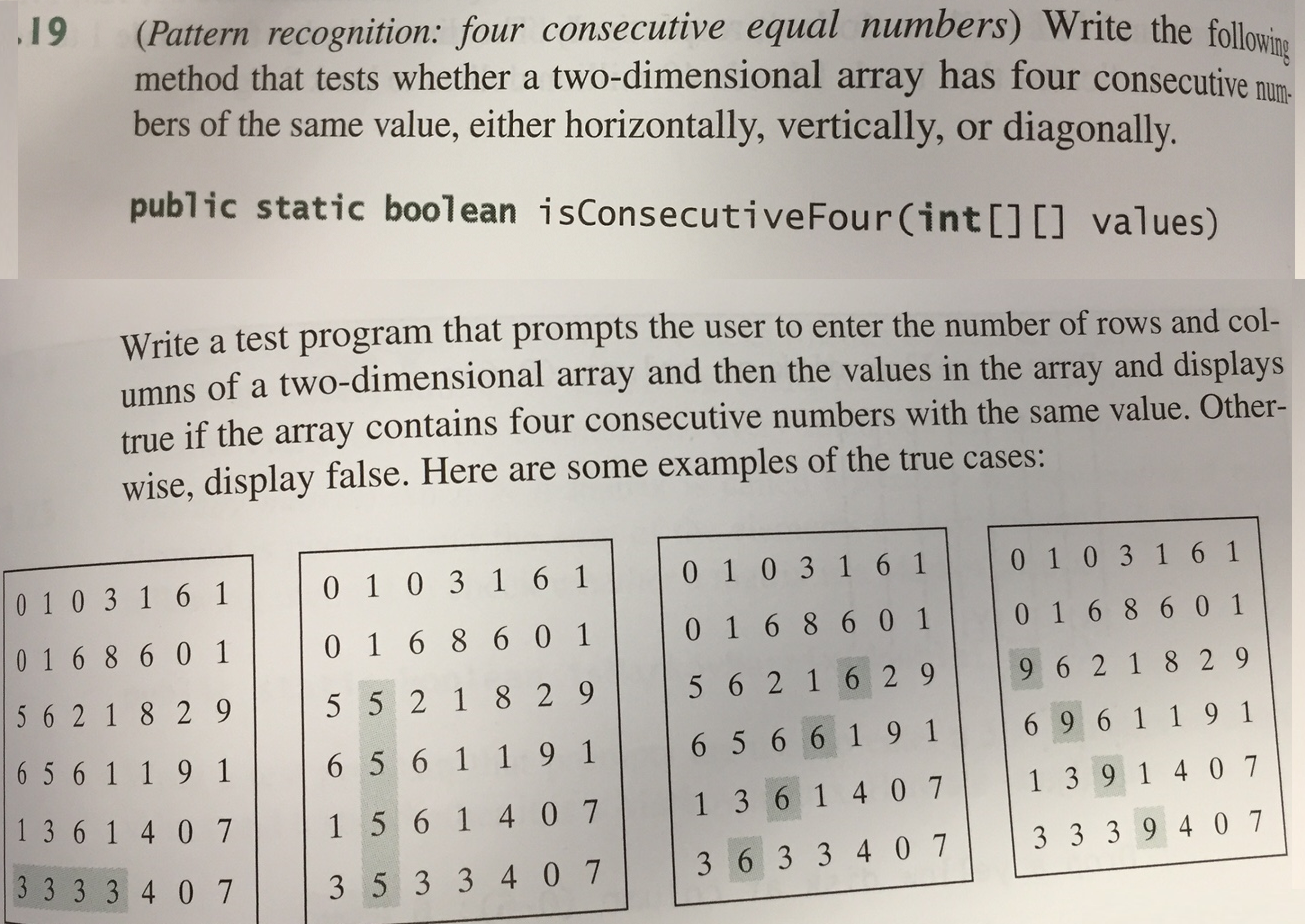
**CIS 357 Assignment 2**

**Due:**11:59 pm, Monday, February 13

**What To Submit:** All source files (.java) and output file submitted to Canvass. Print out of Java source code and output in class.

Refer to problem 8.19 in page 312 – 313. Name your program to Homework2LastName.java.



We are modifying the given problem in the textbook. You want to provide the following methods:

public static int[][] generate2DArray(int row, int col)

public static void minimumConsecutiveSum(int[][] values, int num)

public static void testWithFixedValues()

public static void testWithRandomValues()

* generate2DArray(n, m) generates an *n X m* two dimensional array with array element values ranging between 0 and 9. Use the Random class to generate numbers in random without using the seed value 1.
* minimumConsecutiveSum(twoDArray, num) finds the num consecutive numbers either horizontally, vertically, or diagonally from the array, whose sum is minimum among all consecutive numbers.
* testWithFixedValues() tests the program with 6 x 7 array and fixed values of elements. Use the first and third array shown above. It needs to call minimumConsecutiveSum(twoDArray1, 4) and minimumConswcutiveSum(twoDArray3, 4) to find the minimum sum of four consecutive numbers from each array.
* testWithRandomValues() tests the program with number of rows and columns entered by the user. It calls generate2DArray(n, m) to generate a random 2D array of size n by m. Make sure that the number of rows and columns entered by the user are greater than 4 and less than 11. It asks the user to enter num which is between 3 and 5 to find the minimum consecutive sum (of 3, 4, or 5 values).
* you can add more methods if the length of any method exceeds 60 lines of code.

Ask the user for fixed value testing or random value testing. The following shows a sample output. Note that the italicized bold face is used to indicate user input.

> java Homework2-LastName

Enter a choice: (1 for fixed value, 2 for random values) ***1***

Consecutive four: found ([5,0] – [5,3])

0 1 0 3 1 6 1

0 1 6 8 6 0 1

5 6 2 1 8 2 9

6 5 6 1 1 9 1

1 3 6 1 4 0 7

3 3 3 3 4 0 7

Minimum of consecutive four: 12

> java Homework2-LastName

Enter a choice: (1 for fixed value, 2 for random values) ***1***

Consecutive four: found ([2, 4] – [5, 1])

0 1 0 3 1 6 1

0 1 6 8 6 0 1

5 6 2 1 6 2 9

6 5 6 6 1 9 1

1 3 6 1 4 0 7

3 6 3 3 4 0 7

Minimum of consecutive four: 24

> java Homework2-LastName

Enter a choice: (1 for fixed value, 2 for random values) ***2***

Enter row and column: ***7 7***

Enter consecutive number (3, 4, 5): ***6***

Enter consecutive number (3, 4, 5): ***5***

Consecutive five: not found

0 1 0 3 1 6 1

0 1 6 8 6 0 1

5 6 2 1 8 2 9

6 5 6 1 1 2 1 <== Note: your array element values should look different

1 3 6 1 4 2 7

3 3 3 2 4 2 7

1 1 1 1 2 2 3

> java Homework2-LastName

Enter a choice: (1 for fixed value, 2 for random values) ***3***

Wrong input. Try again!

Enter a choice: (1 for fixed value, 2 for random values) ***2***

Enter row and column: ***4 5***

Enter consecutive number (3, 4, 5): ***3***

Consecutive three: found ([0,3] – [3,3])

0 1 0 2 1

0 1 6 2 6

5 6 2 2 8 <== Note: your array element values should look different

6 1 1 3 1

Minimum sum of consecutive three: 6

* Provide three more outputs for random value tests with the size of 9x9 array (with 3 consecutive numbers) and10x10 array (with 5 consecutive numbers).

**Important Notes and Requirements:**

* Use the Scanner class for the user input.
* Format your code nicely (indenting, layout, etc.) and be sure to **use Javadoc comments** to explain your code where appropriate.  Refer to homework 1 for program commenting.

// homework 2: ...

// Student Name:

// Course, semester:

// Instructor:

// Date finished:

// Program description:

public class Homework2LastName

{

...

public void static main(String[] args)

{

...

}

/\*\*

\* method description for ...

\*/

...

}

* Your output should also be nicely formatted.
* Follow the Java convention we covered in class to get a full credit.
* Start a class name with a capital letter
* Start a variable name with a lower case letter
* Use descriptive name for variables, member data, and member function.
* Be consistent in formatting and indenting your code. Refer to code layout provided in the textbook.
* Modularize your code. If you have a method that contains more than 60 lines of code, there will be a deduction, up to 5%. You can use multiple classes if you want, but it is not required. You can use static functions.
* Do not use packages. If you do, you will get a 2-point deduction. Try to use a text editor (That is, avoid using Java IDE tools like Eclipse or NetBeans. We will cover them later!).
* A program that does not compile will not get more than 40 point.
* A program without proper description or Javadoc comments will lose up to 10 point.
* Codes with poor and inconsistent layout will lose up to 5 point.
* Submission without output will lose up to 10 points.