CSCI 428 Assignment 03

60 points

Please read the instructions before working on the assignment:

- Assignments will be given and returned via the D2L as a convenience to the students and the instructor. Do not send assignments via email.
- It is the student's responsibility to have all assignments ready on time by the given due date. Late assignments may not be accepted.
- Discussion is allowed and encouraged, but no cheating. Assignments with copied answers or other cheating (all or in part) receive a grade of 0.
- Assignments or programs with extra features, and fancy output, you may receive extra scores.
- The first lines of the source code of the program should include a comment with the following information and format:

```
/**
* A short description of the program.
*
* @author's last name, first Name
* @assignment CSCI 428 Assignment X -Qn M
* @date MM/DD/ YYYY
*/
```

- For all programming questions, copy the Java source code into the Word file named *"CSCI428-AssignX-XX.doc(x)"* and screenshots of the program's output. Please replace the first `*X*' with the Assignment number and "*XX*" with your first name and last name. For instance, the title for assignment 01 should be "*CSCI428- Assign01-JessicaWang.doc(x)*".
- List answers to all short-answer questions in the Word file.
- Submit the Word (.doc(x)) file and Java source files (.java) into the D2L.
- **NO PDF FILES.**

Questions:

Qn1. (30 points, 5 points each) Write Java statements for performing the following tasks with a 2D array named board that has 20 rows and 30 columns.

a. Declare the array board.
b. Fill all entries with 0.
c. Fill elements alternately with 0s and 1s in a checkerboard pattern.
d. Fill only the elements at the top and bottom row with zeros.
e. Compute the sum of all elements.
f. Print the array in tabular form (Hint: you could use the method System.out.printf()).

Qn2. (10 points, 2.5 points each) The following formula gives the distance between two points (x1, y1)

and (x2, y2) in the Cartesian plane:

$$distance = \sqrt{(x_2 - x_1)^2 + \left(y_2 - y_1\right)^2}$$

Given the center and a point on a circle, you can use this formula to find the radius of the circle. Write a program that prompts the user to enter the center and a point on the circle. The program should then output the circle's radius, diameter, circumference, and area. Your program must have at least the following methods:

a. **distance**: This method takes as its parameters four numbers that represent two points in the plane and returns the distance between them.
b. **radius**: This method takes as its parameters four numbers that represent the center and a point on the circle, calls the method distance to find the radius of the circle, and returns the circle's radius.
c. **circumference**: This method takes as its parameter a number that represents the radius of the circle and returns the circle's circumference. (If r is the radius, the circumference is .)
d. **area**: This method takes as its parameter a number that represents the radius of the circle and returns the circle's area. (If r is the radius, the area is $2\pi r$).
e. Assume that $\pi = 3.1416$

Qn3. (20 points, 5 points each) Consider the following class:

```
public class Sequence {
        public static void main ( String [] args ){
                // add statements below
        }
// add method definition below
}
```

a. Use arrays (not ArrayList) to create a method

```
public static int [] append (int [] a, int [] b )
```

that creates a new array, appending the arrays a and b without modifying any array a or b. For example, if a is

```
1 4 9 16
```

and b is

```
9 7 4 9 11
```

then the call c=append(a,b) returns the array c

```
1 4 9 16 9 7 4 9 11
```

without modifying a or b.

b. Use arrays (not ArrayList) to create a method

```
public static int [] merge (int [] a, int [] b)
```

to the Sequence class that creates a new array by merging two arrays and alternating elements from both arrays. If one array is shorter than the other, then alternate as long as you can and then append the remaining elements from the longer array. For example, if a is

1 4 9 16

and b is

9 7 4 9 11

then a.merge(b) returns the array

1 9 4 7 9 4 16 9 11

without modifying a or b.

c. Use arrays (not ArrayList) to create a method

public static int [] mergeSorted (int [] a, int [] b)

to the Sequence class that merges two sorted arrays, producing a new sorted array. Keep an index in each array, indicating how much of it has been processed already. Each time, append the smallest unprocessed value from either array, then advance the index. For example, if a is

1 4 9 16

and b is

4 7 9 9 11

then c=mergeSorted(a,b) returns the array

1 4 4 7 9 9 9 11 16

If a or b is not sorted, merge the longest prefixes of a and b that are sorted.

d. Add statements in the main function to test your methods designed in Qn3 (a)-(c) with method calls.

e. Submit your .java file.