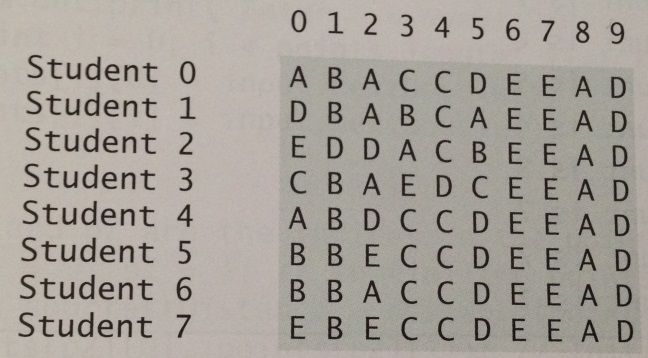
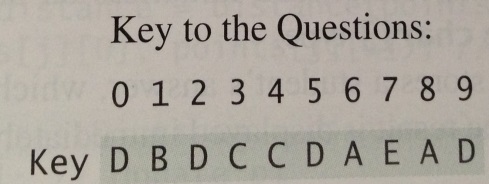
# WORKSHEET #10

## Assumes: Ch1, Ch2, Ch3, Ch5, Ch6, Ch7, Ch8, Ch9-two-dimensional-array

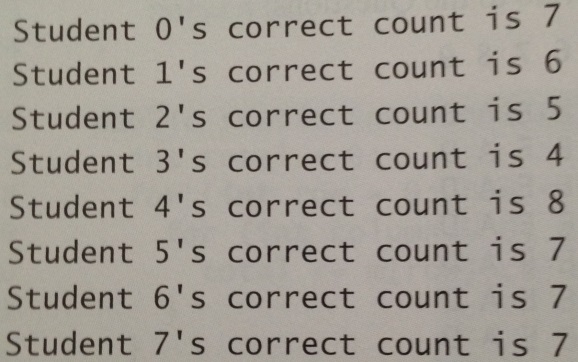
1. Following Object Oriented methodology, write a service class that has a two-dimensional array of ints as its instance variable called **theGrades** where rows represent students and columns represent questions, each row records a student’s answers to the questions as shown in the following array:



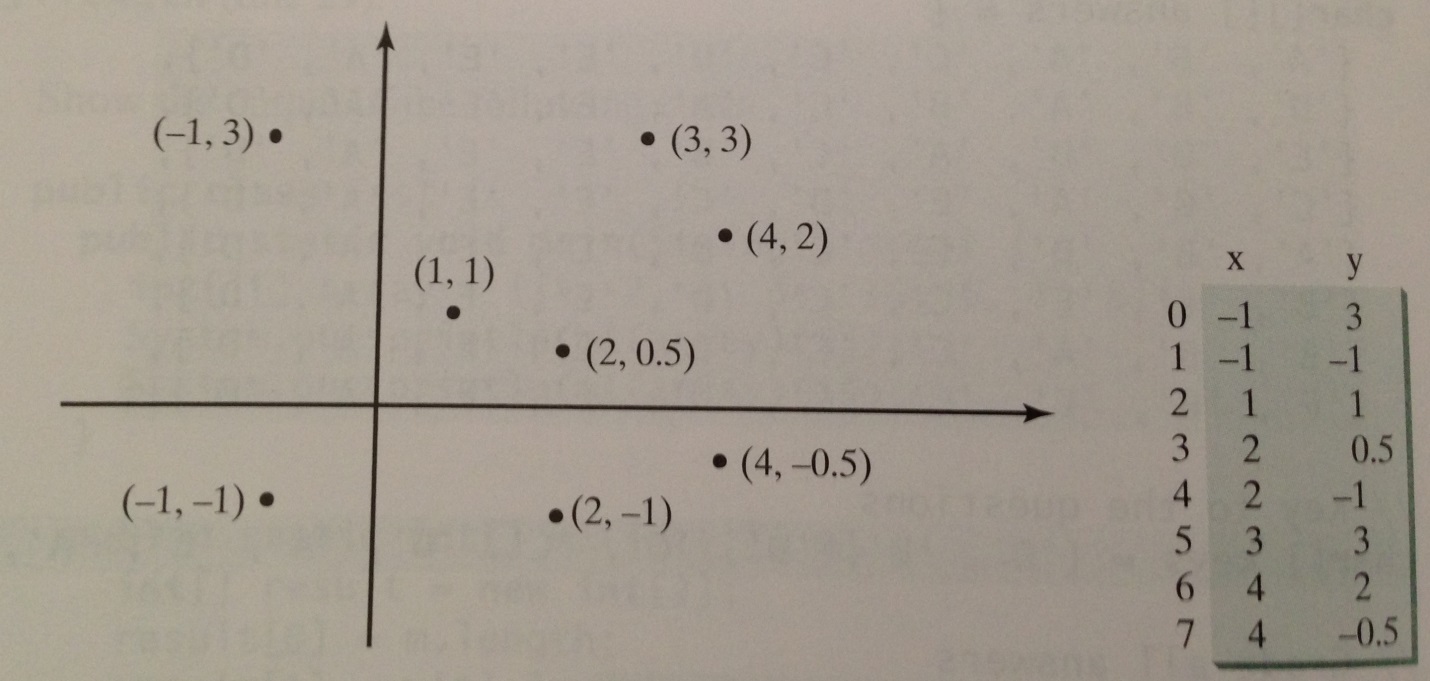
The key is stored in a one-dimensional array:



Your program needs to grade the test and display the result. To grade the test, it compares each student’s answers with the key, counts the number of correct answers, and displays it:



Your service class does not have a default constructor, only the secondary constructor that takes two-dimensional array as the input. This array is defined in the client and contains students’ answers for one test. The client should be able to get test results for multiple tests for different number of students and different number of questions per test.

1. Given a set of points, the closest-pair problem is to find the two points that are nearest to each other. In the following figure, for example, point (1,1) and (2, 0.5) are closest to each other:

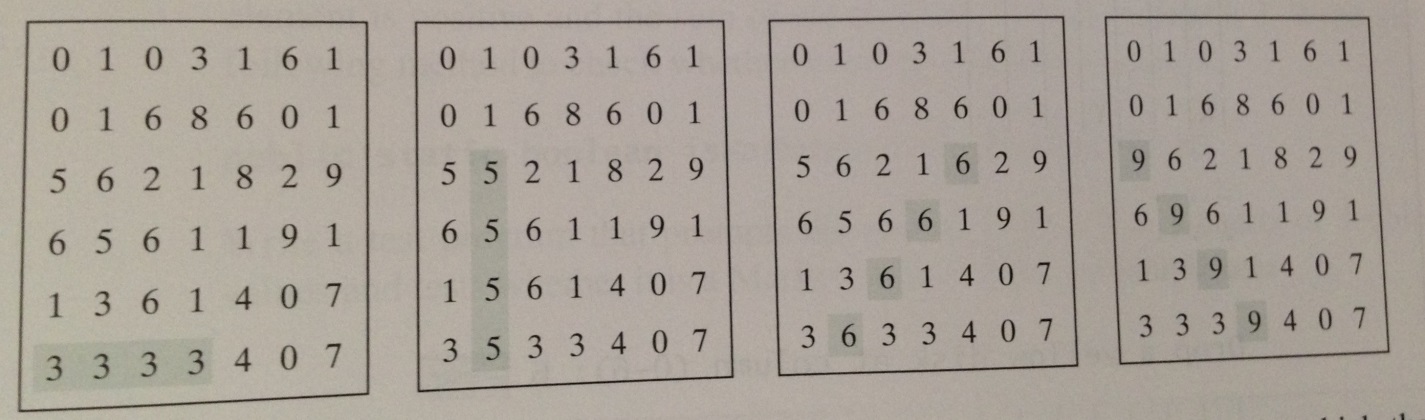
There are several ways to solve this problem. An intuitive approach is to compute the distances between all pairs of points and find the minimum distance.

Using Object Oriented methodology, design an application that solves this problem.

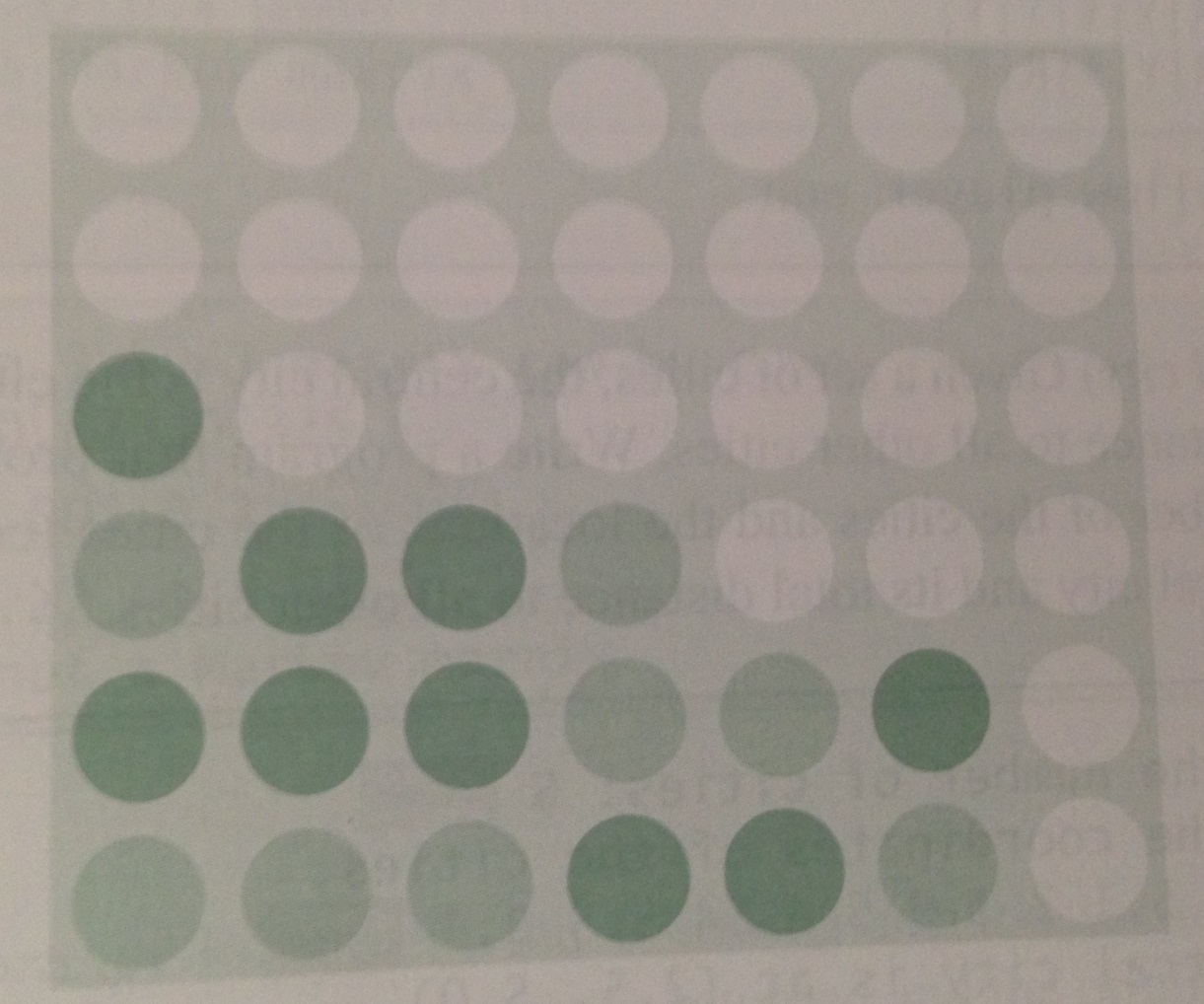
The client should prompt the user to enter the number of points. The points are read from the keyboard and stored in a two-dimensional array of doubles. These values should be passed to your class constructor!

The distance between two points (x1, y1) and (x2, y2) should be calculated by an appropriate method, which should utilize the following formula:

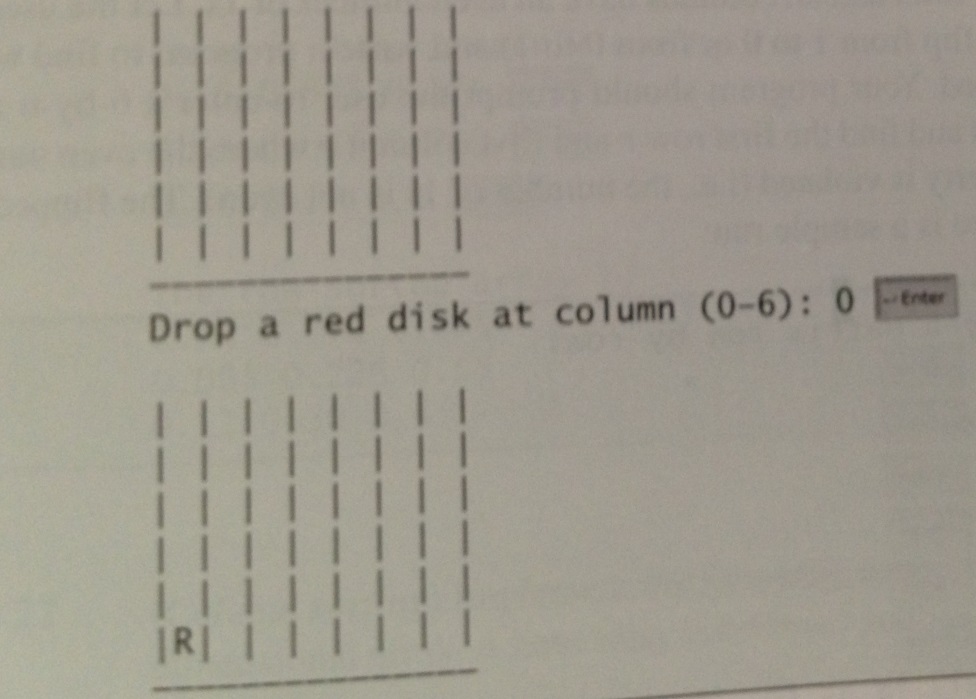
1. Using Object Oriented methodology, write a service class that has a two-dimensional array of ints as its instance variable called **theNumbers**, the array is filled by the secondary constructor with random one digit integers. The number of rows and columns is given by the user when prompted from the client. It has a method that tests whether the **theNumbers** array has four consecutive numbers of the same value, either horizontally, vertically, or diagonally. Here are some examples of the ***true*** cases:

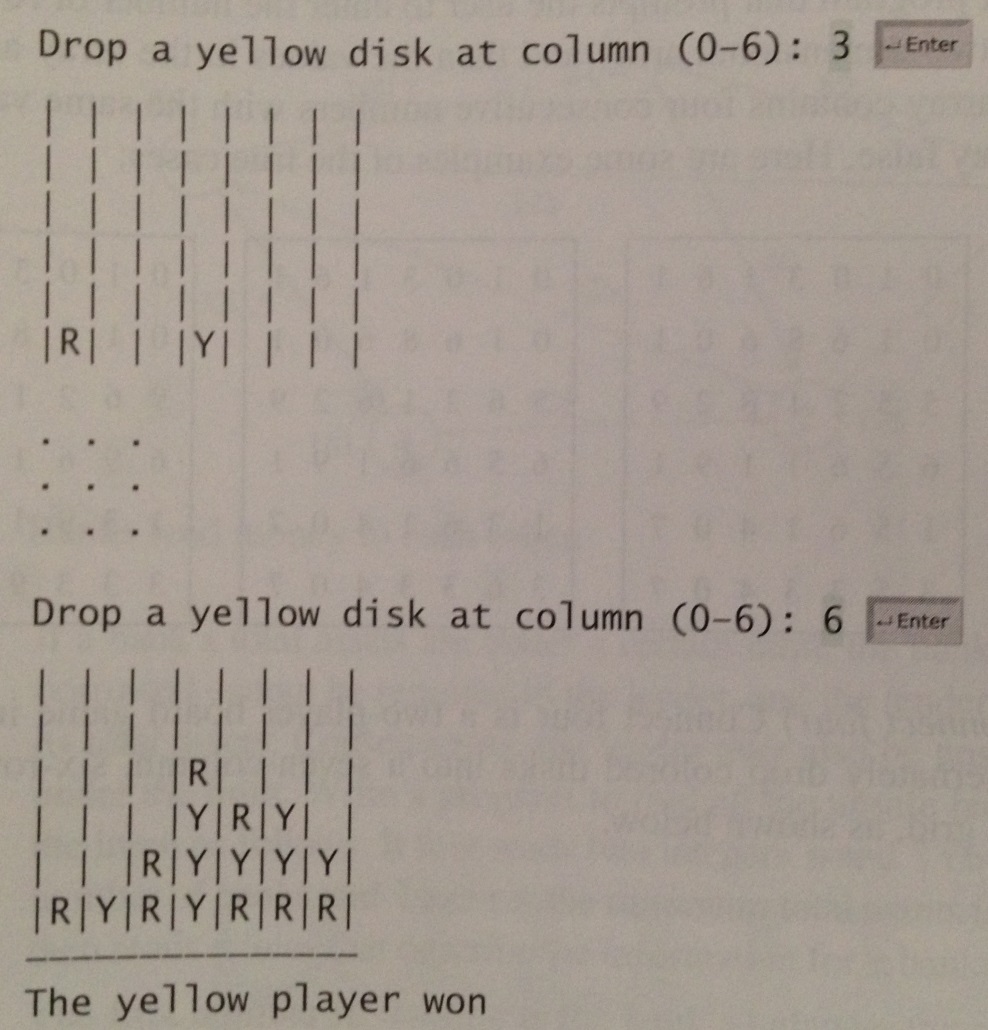


1. Expand on your solution for P3 and implement a game called Connect Four – it is a two-player board game in which the players alternately drop colored disks into a seven-column, six-row vertically suspended grid, as shown below:



The objective of the game is to connect four same-colored disks in a row, a column, or diagonal before your opponent can do likewise. The program client should prompt two players to drop a red or yellow disk alternately. Whenever a disk is dropped, the program redisplays the board on the console and determines the status of the game (win, draw, or continue). Here is a possible sample run:





1. ☺