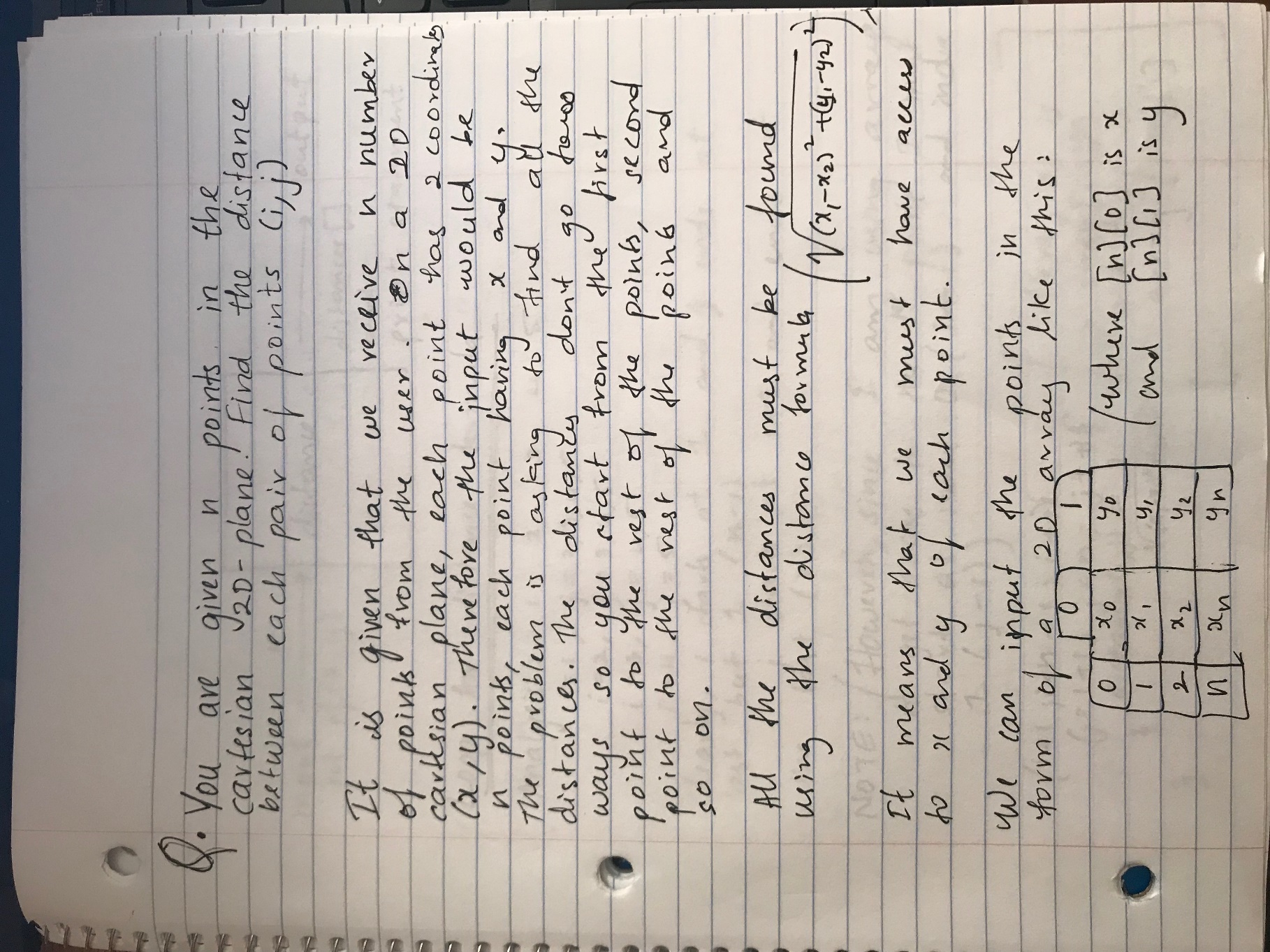
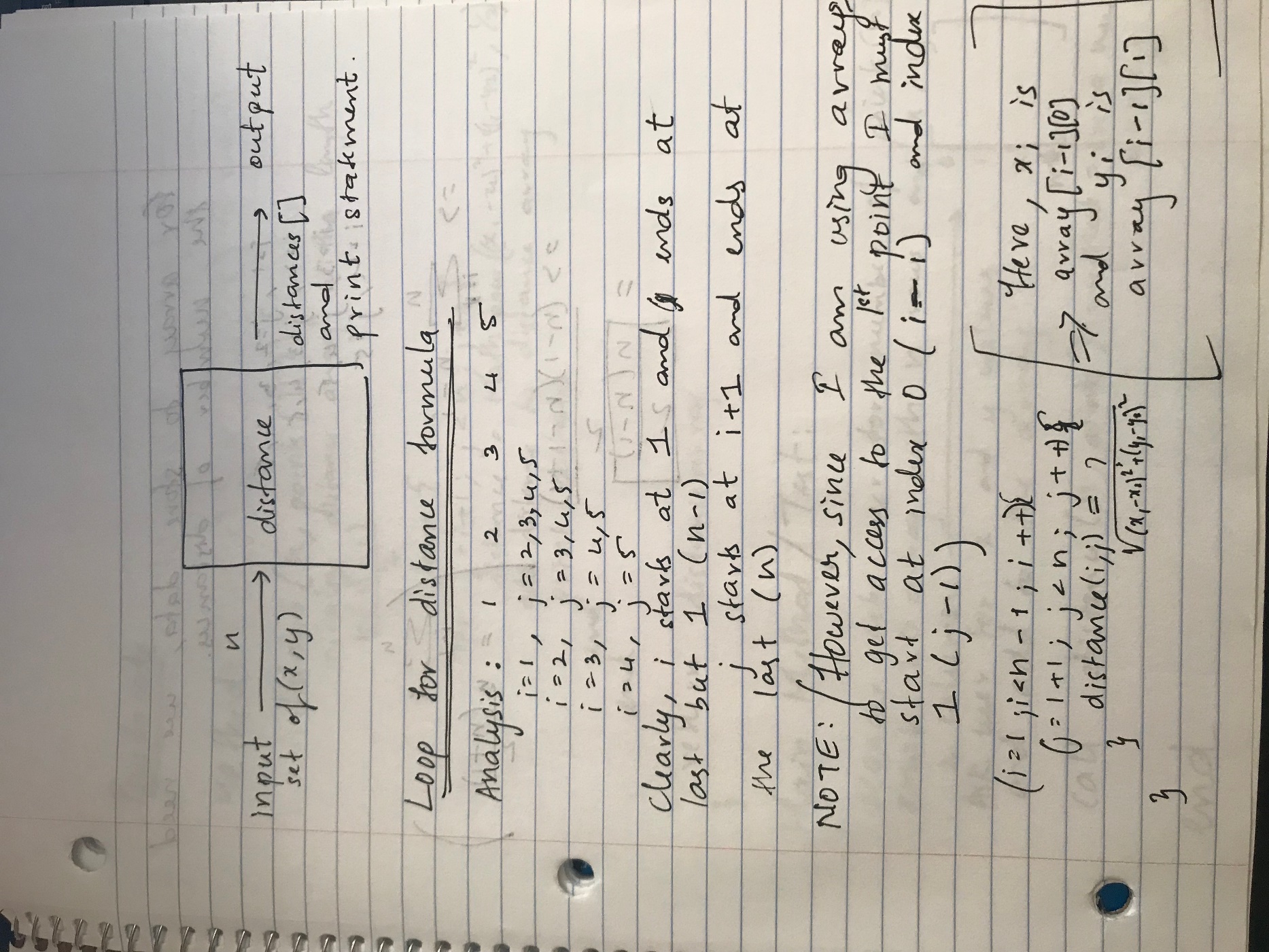
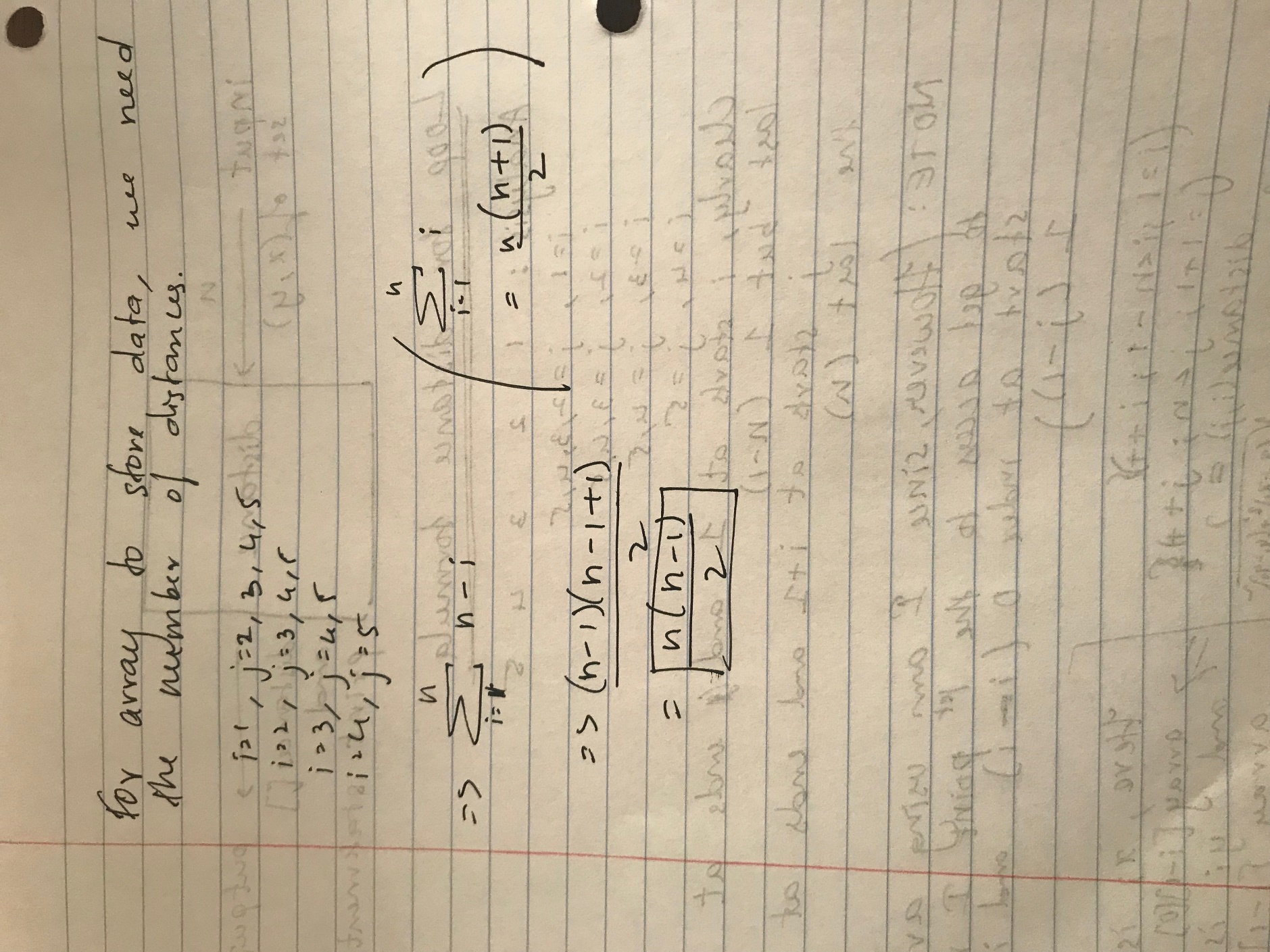
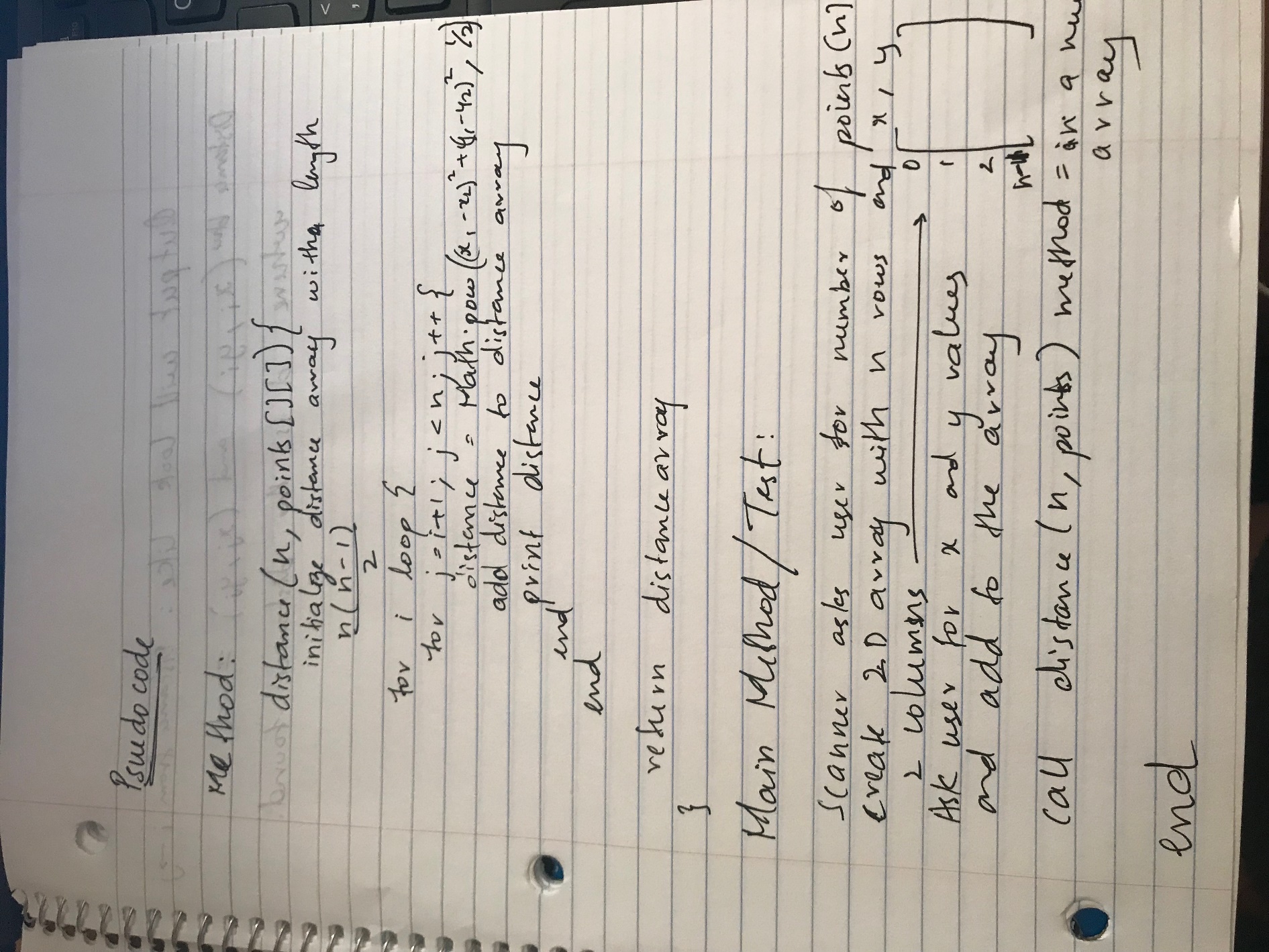
**Shilpa Kannan**

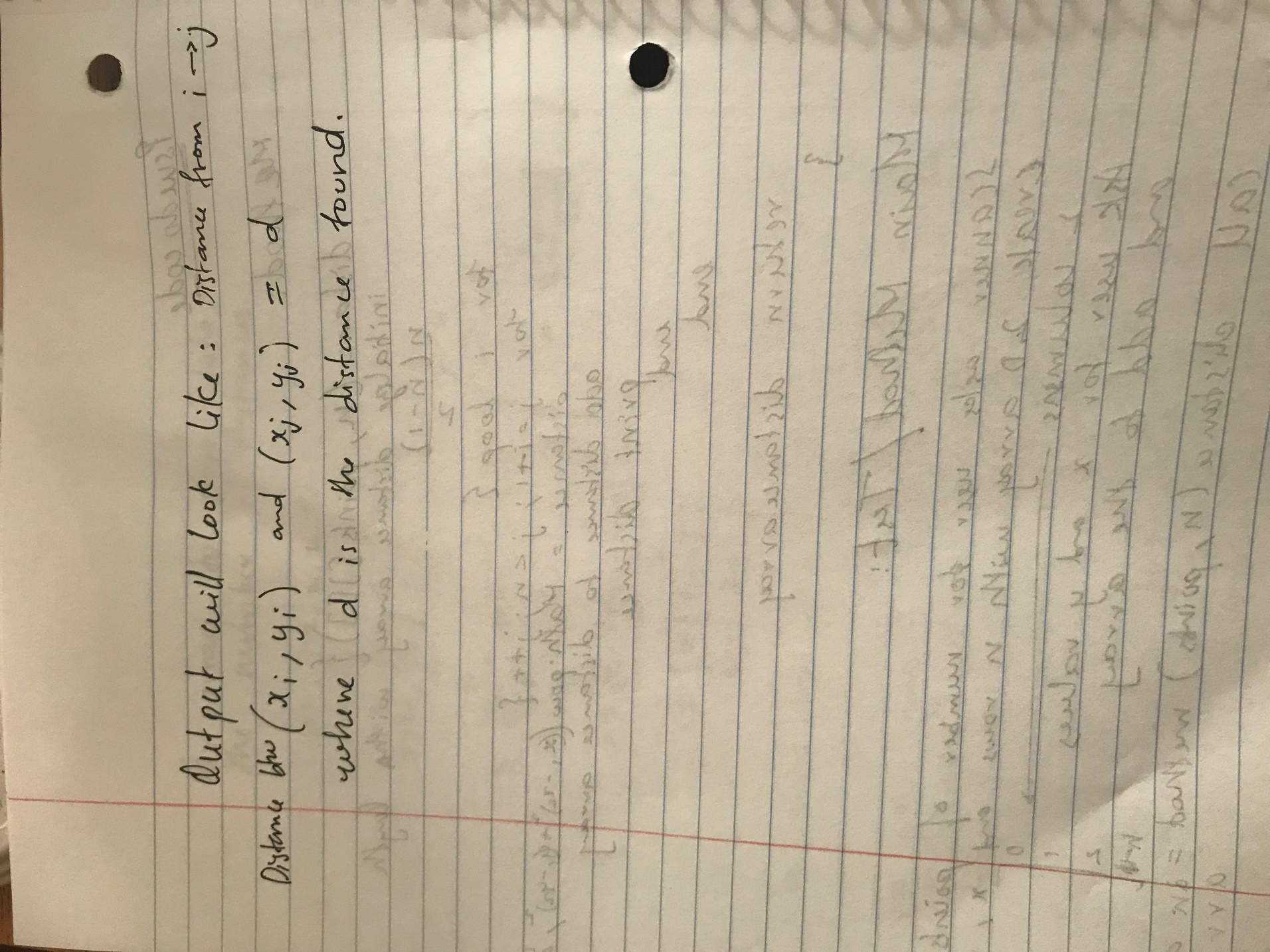
**Data Structures and Algorithms Assignment 1**











**Java Program**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* This program takes n points on the cartisan 2D plane as input and returns and prints all the distances  
\* between the points  
\* Author : Shilpa Kannan  
\* Date : 08/20/2019  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
import java.util.Scanner;  
public class DistanceTest{  
  
 //Main Method  
 public static void main(String[] args){  
 //int [][] points = {{1,12},{2,32},{8,9},{5,63},{4,16}};  
 Scanner input = new Scanner(System.in);  
 System.out.println("Enter how many points n are on the plane");  
 int size = input.nextInt();  
   
 //Creates a 2D array which has n rows and two columns for x and y coordinates of each row/point  
 int [][] points = new int[size][2];  
   
 //Let the user input the coordinates for each point  
 for(int i = 1; i <= size; i++){  
 System.out.println("Point "+i);  
 System.out.println("Enter x value");  
 points[i-1][0] = input.nextInt();  
 System.out.println("Enter y value");  
 points[i-1][1] = input.nextInt();  
 System.out.println("-----------");  
 }  
 //Call the distance method with the given size and points  
 double [] distances = distance(size, points);  
 }   
 //Method to calculate the distances, returns all the distances in an array and prints the distances   
 public static double[] distance(int n, int[][] gPoints){  
 //Declare a distance array to store all the distances  
 double[] distances = new double[((n\*(n-1))/2)];  
   
 //initialize index for the new distance array  
 int a = 0;  
   
 //Nested loop starts at point i = 1 and finds the distances between point i and every point j that comes after it  
 for(int i = 1; i <= n-1; i++){  
 for(int j = i+1; j <= n; j++){  
 //Distance formula  
 double d = Math.sqrt(((Math.pow((gPoints[i-1][0]-gPoints[j-1][0]),2)+(Math.pow((gPoints[i-1][1]-gPoints[j-1][1]),2)))));  
   
 //Print the distances   
 System.out.println("Distance between ("+gPoints[i-1][0]+", "  
 +gPoints[i-1][1]+") and ("+gPoints[j-1][0]+", "+gPoints[j-1][1]+") = "+d);   
 //Assign distance to the array  
 distances[a] = d;  
 a++;  
 }   
 }  
 return distances;  
 }  
}

**Data**

int [][] points = {{1,12},{2,32},{8,9},{5,63},{4,16}};

Same as 5 coordinate points: (1,12), (2,32), (8,9), (5,63), (4,16)

**Expected Output**

Distance between (1, 12) and (2, 32) = 20.024984394500787  
Distance between (1, 12) and (8, 9) = 7.615773105863909  
Distance between (1, 12) and (5, 63) = 51.15662224971465  
Distance between (1, 12) and (4, 16) = 5.0  
Distance between (2, 32) and (8, 9) = 23.769728648009426  
Distance between (2, 32) and (5, 63) = 31.144823004794873  
Distance between (2, 32) and (4, 16) = 16.1245154965971  
Distance between (8, 9) and (5, 63) = 54.08326913195984  
Distance between (8, 9) and (4, 16) = 8.06225774829855  
Distance between (5, 63) and (4, 16) = 47.01063709417264

**Class File**

