CSC 381-34: Proj4A (JAVA)

Swrajit Paul

Due date: Oct. 4, 2018

```
III. Algorithms
*****************************

step 0: inFile ← open input file
      numRows, numCols, minVal, maxVal ← read from inFile
      dynamically allocate zeroFramedAry and skeletonAry with extra 2 rows and 2 cols

step 1: zeroFramed (ZeroFramedAry)

Step 2: loadImage (ZeroFramedAry)


step 3: fistPass_4Distance (ZeroFramedAry)

step 4: prettyPrintDistance (ZeroFramedAry)to outFile_3
           // with proper caption i.e., Pass-1 result

step 5: secondPass_4Distance  (ZeroFramedAry)

Step 6: output newMinVal and newMaxVal to outFile_1

Step 7:  printImage(ZeroFramedAry, outFile_1)

           // output the result of Pass-2 to outFile_1
           (*without* the 2 extra rows and columns)

Step 8: prettyPrintDistance (ZeroFramedAry) // to outFile_3
           // with proper caption i.e., Pass-2 result

step 9: compute_localMaxima(ZeroFramedAry, skeletonAry)
           // see this method given in the above

Step 10:  output newMinVal and newMaxVal to outFile_2

Step 11:  printImage(skeletonAry, outFile_2)

           // output the result of skeleton to outFile_2
           (*without* the 2 extra rows and columns)

Step 12: prettyPrintSkeleton (skeletonAry)// to outFile_3

Step 13: close all files
```

SOURCE CODE

```
/**
 * Project 4
 * Author: Swrajit Paul
 */

import java.io.*;
import java.util.Scanner;

public class ImageProcessing {

	static int numRows;
	static int numCols;
	static int minVal;
	static int maxVal;
	static double newMinVal = 0;
	static double newMaxVal = 0;
	static double[][] zeroFramedAry;
	static int[][] skeletonAry;

   static FileInputStream fInput = null;
   static FileOutputStream fOutputone;
   static FileOutputStream fOutputtwo;
   static FileOutputStream fOutputthree;

   static Scanner inputfile;

   public ImageProcessing() {

   }

   private static void loadImage(double[][] arrayB) {
		for(int i = 1; i < numRows+1; i++) {
			for(int j = 1; j < numCols+1; j++) {
				arrayB[i][j] = inputfile.nextInt();
			}
		}
	}

   private static void zeroFrame(double[][] FramedAry) {
		for(int j = 0; j < numCols+2; j++) {
			FramedAry[0][j] = 0;
			FramedAry[numRows+1][j] = 0; }

		for(int j = 0; j < numRows+2; j++) {
			FramedAry[j][0] = 0;
			FramedAry[j][numCols+1] = 0;
		}
	}

   private static void fistPass_EuclidianDistance (double[][] imgAry) {
		for(int i = 1; i < numRows+1; i++) {
			for(int j = 1; j < numCols+1; j++) {
				if (imgAry[i][j] > 0) {
					double[] tempAry = new double[4];
```

```
                                        tempAry[0] = Double.parseDouble(String.format("%.2f", imgAry[i-1][j-1] +
Math.sqrt(2)));
                                        tempAry[1] = imgAry[i-1][j] + 1;
                                        tempAry[2] = imgAry[i-1][j+1] + Math.sqrt(2);
                                        tempAry[3] = imgAry[i][j-1] + 1;

                                        double min = 20000000.0;

                                        for(int k =0; k < 4; k++){
                                                if(Double.parseDouble(String.format("%.2f", tempAry[k])) < min){
                                                min = Double.parseDouble(String.format("%.2f", tempAry[k])); } }

                                        imgAry[i][j] = min;
                                }
                        }
                }
        }

    private static void secondPass_EuclidianDistance (double[][] imgAry) {
                for(int i = numRows+1; i>= 1; i--) {
                        for(int j = numCols+1; j >= 1; j--) {
                                if (imgAry[i][j] > 0){
                                        double []tempAry = new double[4];
                                        tempAry[0] = imgAry[i][j+1] +1;
                                        tempAry[1] = imgAry[i+1][j-1] + Math.sqrt(2);
                                        tempAry[2] = imgAry[i+1][j] +1;
                                        tempAry[3] = imgAry[i+1][j+1] + Math.sqrt(2);

                                        double min = 20000000.0;

                                        for(int k =0; k < 4; k++){
                                                if(Double.parseDouble(String.format("%.2f", tempAry[k])) < min){
                                                        min = Double.parseDouble(String.format("%.2f", tempAry[k])) ;
                                                }
                                        }
                                        if (imgAry[i][j] >= min){

                                                imgAry[i][j] = min;
                                        }
                                        if(imgAry[i][j] >= newMaxVal){
                                                newMaxVal = imgAry[i][j];
                                        }
                                }
                        }
                }
        }

    private static int is_maxima (double[][] imgAry, int i, int j){

                double tempAry[] = new double[8];
                tempAry[0] = imgAry[i-1][j-1];
                tempAry[1] = imgAry[i-1][j];
                tempAry[2] = imgAry[i-1][j+1];
                tempAry[3] = imgAry[i][j-1];
                tempAry[4] = imgAry[i][j+1];
                tempAry[5] = imgAry[i+1][j-1];
```

```
                tempAry[6] = imgAry[i+1][j];
                tempAry[7] = imgAry[i+1][j+1];

                for(int k =0; k < 8; k++){
                        if (!(imgAry[i][j] >= tempAry[k])){
                                return 0;
                        }
                }
          return 1;
        }

private static void compute_localMaxima(double[][] imgAry, int[][] skAry){
        for(int i = 1; i < numRows+1; i++) {
                        for(int j = 1; j < numCols+1; j++) {
                                if (imgAry[i][j] > 0){
                                        if(is_maxima(imgAry, i,j) ==1){
                                                skeletonAry[i][j] = 1;
                                        }
                                        else{
                                                skeletonAry[i][j] = 0;
                                        }
                                }
                        }
                }
        }

private static void printImage(double[][] imgAry,  FileOutputStream oFile) {
        PrintStream print = new PrintStream(oFile);
        print.println(numRows + " " + numCols + " " + newMinVal + " " + newMaxVal );
        for(int i = 1; i < numRows+1; i++) {
                        for(int j = 1; j < numCols+1; j++) {
                                print.print(imgAry[i][j] + " ");
                        }
                        print.println();
                }
        }

private static void printSkeleton(int[][] imgAry,  FileOutputStream oFile) {
        PrintStream print = new PrintStream(oFile);
        print.println(numRows + " " + numCols + " " + newMinVal + " " + 1 );
        for(int i = 1; i < numRows+1; i++) {
                        for(int j = 1; j < numCols+1; j++) {
                                print.print(imgAry[i][j] + " ");
                        }
                        print.println();
                }
        }

private static void prettyPrintDistance (double[][] imgAry, String pass) {
        PrintStream print = new PrintStream(fOutputthree);
        print.println(pass);
                for(int i = 1; i < numRows+1; i++) {
                        for(int j = 1; j < numCols+1; j++) {
                                if (imgAry[i][j] == 0.0)
                                        print.print("   ");
                                else {
```

```
                                        if(imgAry[i][j] / 10.0 == 0.0)
                                                print.print(String.format("%.1f",imgAry[i][j]) + " ");
                                        else
                                                print.print(String.format("%.1f",imgAry[i][j]));
                                }
                        }
                        print.println();
                }
                print.println();
        }

private static void prettyPrintSkeleton (int[][] imgAry) {
        PrintStream print = new PrintStream(fOutputthree);
                for(int i = 1; i < numRows+1; i++) {
                        for(int j = 1; j < numCols+1; j++) {
                                if (imgAry[i][j] == 0)
                                        print.print(".");
                                else {
                                        print.print("9");
                                }
                        }
                        print.println();
                }
                print.println();
        }

        public static void main(String[] args) {

                try {

                        String inputone = args[0];
                        String outputone = args[1];
                        String outputtwo = args[2];
                        String outputthree = args[3];

                        fInput = new FileInputStream(inputone);
                        fOutputone = new FileOutputStream(outputone);
                        fOutputtwo = new FileOutputStream(outputtwo);
                        fOutputthree = new FileOutputStream(outputthree);

                } catch (IOException e) {
                        System.out.println("one of the arguments in missing or wrong");
                }


                inputfile = new Scanner(fInput);
                numRows = inputfile.nextInt();
                numCols = inputfile.nextInt();
                minVal = inputfile.nextInt();
                maxVal = inputfile.nextInt();

                zeroFramedAry = new double[numRows+2][numCols+2];
                skeletonAry = new int[numRows+2][numCols+2];

                zeroFrame(zeroFramedAry);
                loadImage(zeroFramedAry);
```

```
                fistPass_EuclidianDistance(zeroFramedAry);
                prettyPrintDistance(zeroFramedAry, "pass-1");
                secondPass_EuclidianDistance(zeroFramedAry);
                printImage(zeroFramedAry, fOutputone);
                prettyPrintDistance(zeroFramedAry, "pass-2");
                compute_localMaxima(zeroFramedAry, skeletonAry);
                printSkeleton(skeletonAry, fOutputtwo);
                prettyPrintSkeleton(skeletonAry);

                inputfile.close();
                try {
                        fInput.close();
                } catch (IOException e) {

                        e.printStackTrace();
                }
        }

}
```

INPUT 1

20 40 0 1

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0
0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0
0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0
0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0

INPUT 2

38 31 0 1

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

OUTPUT FOR INPUT 1

OUTPUT1

```
20 40 0.0 6.64
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.41 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.41 2.41 1.41 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.41 2.41 2.82 2.41 1.41 1.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.41 2.41 2.82 3.82 2.82 2.41 1.41 1.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.41 2.41 2.82 3.82 4.23 3.82 2.82 2.41 1.41 1.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.41 2.41 2.82 3.82 4.23 5.23 4.23 3.82 2.82 2.41 1.41 1.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.41 2.41 2.82 3.82 4.23 5.23 5.64 5.23 4.23 3.82 2.82 2.41 1.41 1.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.41 2.41 2.82 3.82 4.23 5.23 5.64 6.64 5.64 5.23 4.23 3.82 2.82 2.41 1.41 1.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.41 2.41 2.82 3.82 4.23 5.23 5.64 6.23 6.64 6.23 5.64 5.23 4.23 3.82 2.82 2.41 1.41 1.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.41 2.41 2.82 3.82 4.23 4.82 5.23 5.64 5.23 4.82 4.23 3.82 2.82 2.41 1.41 1.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.41 2.41 2.82 3.41 3.82 4.23 5.23 4.23 3.82 3.41 2.82 2.41 1.41 1.0 0.0 0.0
0.0 0.0 0.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 0.0 0.0 0.0 0.0 0.0 1.0 1.41 2.0 2.41 2.82 3.82 4.23 3.82 2.82 2.41 2.0 1.41 1.0 0.0 0.0 0.0
0.0 0.0 0.0 1.0 2.0 2.0 2.0 2.0 2.0 2.0 2.0 2.0 2.0 2.0 2.0 2.0 2.0 2.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.0 1.41 2.41 2.82 3.82 2.82 2.41 1.41 1.0 1.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 1.0 2.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 2.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.41 2.41 2.82 2.41 1.41 1.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 1.0 2.0 3.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 3.0 2.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.41 2.41 1.41 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 1.0 2.0 3.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 3.0 2.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.41 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 1.0 2.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 2.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 1.0 2.0 2.0 2.0 2.0 2.0 2.0 2.0 2.0 2.0 2.0 2.0 2.0 2.0 2.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
```

OUTPUT2

```
20 40 0.0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
```

OUTPUT3

```
pass-1
                                                   1.0
                                                1.01.41.0
                                             1.01.42.41.41.0
                                          1.01.42.42.82.41.41.0
                                       1.01.42.42.83.82.82.41.41.0
                                    1.01.42.42.83.84.23.82.82.41.41.0
                                 1.01.42.42.83.84.25.24.23.82.82.41.41.0
                              1.01.42.42.83.84.25.25.65.24.23.82.82.41.41.0
                           1.01.42.42.83.84.25.25.66.65.65.24.23.82.82.41.41.0
                        1.01.42.42.83.84.25.25.66.67.16.65.65.24.23.82.82.41.41.0
                           1.02.03.04.05.05.66.67.18.17.16.65.65.24.23.82.82.4
                              1.02.03.04.05.06.07.08.08.17.16.65.65.24.23.8
1.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.0 1.02.03.04.05.06.07.08.08.17.16.65.65.2
1.02.02.02.02.02.02.02.02.02.02.02.02.02.02.01.4 1.02.03.04.05.06.07.08.08.17.16.6
```

```
1.02.03.03.03.03.03.03.03.03.03.03.03.03.02.81.4          1.02.03.04.05.06.07.0
1.02.03.04.04.04.04.04.04.04.04.04.04.04.02.81.4          1.02.03.04.05.0
1.02.03.04.05.05.05.05.05.05.05.05.05.04.22.81.4          1.02.03.0
1.02.03.04.05.06.06.06.06.06.06.06.05.64.22.81.4          1.0
1.02.03.04.05.06.07.07.07.07.07.07.05.64.22.81.4          1.0
1.02.03.04.05.06.07.08.08.08.08.07.15.64.22.81.4          1.0
```

pass-2

```
                                                          1.0
                                                          1.01.41.0
                                                          1.01.42.41.41.0
                                                          1.01.42.42.82.41.41.0
                                                          1.01.42.42.83.82.82.41.41.0
                                                          1.01.42.42.83.84.23.82.82.41.41.0
                                                          1.01.42.42.83.84.25.24.23.82.82.41.41.0
                                                          1.01.42.42.83.84.25.25.65.24.23.82.82.41.41.0
                                                          1.01.42.42.83.84.25.25.66.65.65.24.23.82.82.41.41.0
                                                          1.01.42.42.83.84.25.25.66.26.66.25.65.24.23.82.82.41.41.0
                                                          1.01.42.42.83.84.24.85.25.65.24.84.23.82.82.41.41.0
                                                          1.01.42.42.83.43.84.25.24.23.83.42.82.41.41.0
1.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.0          1.01.42.02.42.83.84.23.82.82.42.01.41.0
1.02.02.02.02.02.02.02.02.02.02.02.02.02.02.01.0          1.01.01.42.42.83.82.82.41.41.01.0
1.02.03.03.03.03.03.03.03.03.03.03.03.03.02.01.0          1.01.42.42.82.41.41.0
1.02.03.04.04.04.04.04.04.04.04.04.04.03.02.01.0          1.01.42.41.41.0
1.02.03.04.04.04.04.04.04.04.04.04.04.03.02.01.0          1.01.41.0
1.02.03.03.03.03.03.03.03.03.03.03.03.03.02.01.0          1.0
1.02.02.02.02.02.02.02.02.02.02.02.02.02.02.01.0          1.0
1.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.0          1.0
```

```
........................................
........................................
........................................
........................................
........................................
........................................
........................................
........................................
..............................9.........
..............................9.........
........................................
........................................
........................................
........................................
........................................
......9999999999........................
......9999999999........................
........................................
..............................9.........
..............................9.........
```

OUTPUT FOR INPUT 2

OUTPUT1

```
38 31 0.0 6.64
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.41 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.41 2.41 1.41 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.41 2.41 2.82 2.41 1.41 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.41 2.41 2.82 3.82 2.82 2.41 1.41 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.41 2.41 2.82 3.82 4.23 3.82 2.82 2.41 1.41 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.41 2.41 2.82 3.82 4.23 5.23 4.23 3.82 2.82 2.41 1.41 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 2.0 2.82 3.82 4.23 5.23 5.64 5.23 4.23 3.82 2.82 2.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 2.0 3.0 4.0 5.0 5.64 6.64 5.64 5.0 4.0 3.0 2.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 2.0 3.0 4.0 5.0 6.0 6.64 6.0 5.0 4.0 3.0 2.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 2.0 3.0 4.0 4.82 5.23 5.64 5.23 4.82 4.0 3.0 2.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 2.0 2.82 3.41 3.82 4.23 5.23 4.23 3.82 3.41 2.82 2.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.41 2.0 2.41 2.82 3.82 4.23 3.82 2.82 2.41 2.0 1.41 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.0 1.41 2.41 2.82 3.82 2.82 2.41 1.41 1.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.41 2.41 2.82 2.41 1.41 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.41 2.41 1.41 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 2.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 2.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 2.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.41 2.41 1.41 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.41 2.41 2.82 2.41 1.41 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.41 2.41 2.82 3.82 2.82 2.41 1.41 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.41 2.41 2.82 3.82 4.23 3.82 2.82 2.41 1.41 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.41 2.41 2.82 3.82 4.23 5.23 4.23 3.82 2.82 2.41 1.41 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 2.0 2.82 3.82 4.23 5.23 5.64 5.23 4.23 3.82 2.82 2.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 2.0 3.0 4.0 5.0 5.64 6.64 5.64 5.0 4.0 3.0 2.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 2.0 3.0 4.0 5.0 6.0 6.64 6.0 5.0 4.0 3.0 2.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 2.0 3.0 4.0 4.82 5.23 5.64 5.23 4.82 4.0 3.0 2.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
```

```
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 2.0 2.82 3.41 3.82 4.23 5.23 4.23 3.82 3.41 2.82 2.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.41 2.0 2.41 2.82 3.82 4.23 3.82 2.82 2.41 2.0 1.41 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.0 1.41 2.41 2.82 3.82 2.82 2.41 1.41 1.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.41 2.41 2.82 2.41 1.41 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.41 2.41 1.41 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 2.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.41 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
```

OUTPUT2

```
38 31 0.0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

OUTPUT3

pass-1

```
                1.0
              1.01.41.0
            1.01.42.41.41.0
          1.01.42.42.82.41.41.0
        1.01.42.42.83.82.82.41.41.0
      1.01.42.42.83.84.23.82.82.41.41.0
    1.01.42.42.83.84.25.24.23.82.82.41.41.0
    1.02.02.83.84.25.25.65.24.23.82.82.41.4
    1.02.03.04.05.05.66.65.65.24.23.82.81.4
    1.02.03.04.05.06.07.06.65.65.24.22.81.4
```

1.02.03.04.05.06.07.07.16.65.64.22.81.4
1.02.03.04.05.06.07.08.07.15.64.22.81.4
1.02.03.04.05.06.07.08.07.15.64.22.81.4
1.02.03.04.05.06.07.07.15.64.22.8
1.02.03.04.05.06.05.6
1.02.03.04.05.0
1.02.03.0
1.02.01.4
1.02.01.4
1.01.42.41.41.0
1.01.42.42.82.41.41.0
1.01.42.42.83.82.82.41.41.0
1.01.42.42.83.84.23.82.82.41.41.0
1.01.42.42.83.84.25.24.23.82.82.41.41.0
1.02.02.83.84.25.25.65.24.23.82.82.41.4
1.02.03.04.05.05.66.65.65.24.23.82.81.4
1.02.03.04.05.06.07.06.65.65.24.22.81.4
1.02.03.04.05.06.07.07.16.65.64.22.81.4
1.02.03.04.05.06.07.08.07.15.64.22.81.4
1.02.03.04.05.06.07.08.07.15.64.22.81.4
1.02.03.04.05.06.07.07.15.64.22.8
1.02.03.04.05.06.05.6
1.02.03.04.05.0
1.02.03.0
1.02.01.4
1.0

pass-2

1.0
1.01.41.0
1.01.42.41.41.0
1.01.42.42.82.41.41.0
1.01.42.42.83.82.82.41.41.0
1.01.42.42.83.84.23.82.82.41.41.0
1.01.42.42.83.84.25.24.23.82.82.41.41.0
1.02.02.83.84.25.25.65.24.23.82.82.01.0
1.02.03.04.05.05.66.65.65.04.03.02.01.0
1.02.03.04.05.06.06.66.05.04.03.02.01.0
1.02.03.04.04.85.25.65.24.84.03.02.01.0
1.02.02.83.43.84.25.24.23.83.42.82.01.0
1.01.42.02.42.83.84.23.82.82.42.01.41.0
1.01.01.42.42.83.82.82.41.41.01.0
1.01.42.42.82.41.41.0
1.01.42.41.41.0
1.02.01.0
1.02.01.0
1.02.01.0
1.01.42.41.41.0
1.01.42.42.82.41.41.0
1.01.42.42.83.82.82.41.41.0
1.01.42.42.83.84.23.82.82.41.41.0
1.01.42.42.83.84.25.24.23.82.82.41.41.0
1.02.02.83.84.25.25.65.24.23.82.82.01.0
1.02.03.04.05.05.66.65.65.04.03.02.01.0
1.02.03.04.05.06.06.66.05.04.03.02.01.0
1.02.03.04.04.85.25.65.24.84.03.02.01.0
1.02.02.83.43.84.25.24.23.83.42.82.01.0
1.01.42.02.42.83.84.23.82.82.42.01.41.0
1.01.01.42.42.83.82.82.41.41.01.0

1.01.42.42.82.41.41.0
1.01.42.41.41.0
1.02.01.0
1.01.41.0
1.0