CSC 381-34: Proj4B (C++)

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

```
// Author: Swrajit Paul

#include <iostream>
#include <fstream>

using namespace std;

ifstream inFile;
ofstream outFile;
ofstream outFiletwo;
ofstream outFilethree;

class imageProcessing {

	public:

		int numRows;
		int numCols;
		int minVal;
		int maxVal;
		int newMinVal = 0;
		int newMaxVal = 0;
		int** zeroFramedAry;
		int** skeletonAry;

	    imageProcessing(string in, string out, string outtwo, string outthree) {

		inFile.open(in.c_str());
			outFile.open(out.c_str());
			outFiletwo.open(outtwo.c_str());
			outFilethree.open(outthree.c_str());
			inFile >> numRows;
			inFile >> numCols;
			inFile >> minVal;
			inFile >> maxVal;

			zeroFramedAry = new int*[numRows+2];
			for(int i = 0; i < numRows+2; i++){
				zeroFramedAry[i] = new int[numCols+2]; }// set up the array with proper rows and
cols
			for(int i = 0; i < numRows+2; i++) {
				for(int j = 0; j < numCols+2; j++) {
					zeroFramedAry[i][j] = 0; } }// initialize the array

			skeletonAry = new int*[numRows+2];
			for(int i = 0; i < numRows+2; i++){
				skeletonAry[i] = new int[numCols+2];
			}// set up the array with proper rows and cols
			for(int i = 0; i < numRows+2; i++) {
				for(int j = 0; j < numCols+2; j++) {
					skeletonAry[i][j] = 0; } }// initialize the array
	    }
```

```
void loadImage(int** FramedAry) {
        // reads line by line from the input into zeroFramedAry
        for(int i = 1; i < numRows+1; i++) {
                for(int j = 1; j < numCols+1; j++) {
                        inFile >> FramedAry[i][j]; } }
}

void zeroFrame(int** 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; }
}

void fistPass_4Distance (int** imgAry){
        for(int i = 1; i < numRows+1; i++) {
                for(int j = 1; j < numCols+1; j++) {
                        if (imgAry[i][j] > 0){
                                int tempAry[2];
                                tempAry[0] = imgAry[i-1][j];
                                tempAry[1] = imgAry[i][j-1];
                                int min = 20000000;

                                for(int k =0; k < 2; k++){
                                        if(tempAry[k] + 1 < min){
                                        min = tempAry[k] + 1; } }

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

void secondPass_4Distance (int** imgAry){
        for(int i = numRows+1; i > 0; i--) {
                for(int j = numCols+1; j > 0; j--) {
                        if (zeroFramedAry[i][j] > 0){
                                int tempAry[2];
                                tempAry[0] = imgAry[i+1][j];
                                tempAry[1] = imgAry[i][j+1];

                                int min = 20000000;

                                for(int k =0; k < 2; k++){
                                        if(tempAry[k] + 1 < min){
                                                min = tempAry[k] + 1;
                                        }
                                }
                                if (imgAry[i][j] > min){

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

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


int is_maxima (int** imgAry, int i, int j){

        int tempAry[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;
}

void compute_localMaxima(int** 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;
                                }
                        }
                }
        }
}

void printImage(int** imgAry, ofstream& oFile) {
        for(int i = 1; i < numRows+1; i++) {
                for(int j = 1; j < numCols+1; j++) {
                        oFile << imgAry[i][j] << " ";
                }
                oFile << endl;
        }
}

void prettyPrintDistance (int** imgAry, string pass) {
```

```
                outFilethree << pass << endl;
                for(int i = 1; i < numRows+1; i++) {
                        for(int j = 1; j < numCols+1; j++) {
                                if (imgAry[i][j] == 0)
                                        outFilethree << "  ";
                                else {
                                        if(imgAry[i][j] / 10 == 0)
                                                outFilethree << imgAry[i][j] << " ";
                                        else
                                                outFilethree << imgAry[i][j];
                                }
                        }
                        outFilethree << endl;
                }
                outFilethree << endl;
        }
        void prettyPrintSkeleton (int** imgAry) {
                for(int i = 1; i < numRows+1; i++) {
                        for(int j = 1; j < numCols+1; j++) {
                                if (imgAry[i][j] == 0)
                                        outFilethree << ".";
                                else {
                                        outFilethree << "9";
                                }
                        }
                        outFilethree << endl;
                }
                outFilethree << endl;
        }

};

int main(int argc, char *argv[]) {

        imageProcessing img (argv[1],argv[2],argv[3],argv[4]);
        img.zeroFrame(img.zeroFramedAry);
        img.loadImage(img.zeroFramedAry);
        img.fistPass_4Distance(img.zeroFramedAry);
        img.prettyPrintDistance(img.zeroFramedAry, "Pass-1");
        img.secondPass_4Distance(img.zeroFramedAry);
        outFile << img.numRows << " " << img.numCols << " " << img.newMinVal << " " << img.newMaxVal <<
endl;
        img.printImage(img.zeroFramedAry, outFile);
        img.prettyPrintDistance(img.zeroFramedAry, "Pass-2");
        img.compute_localMaxima(img.zeroFramedAry, img.skeletonAry);
        img.newMaxVal = 1;
        outFiletwo << img.numRows << " " << img.numCols << " " << img.newMinVal << " " << img.newMaxVal
<< endl;
        img.printImage(img.skeletonAry, outFiletwo);
        img.prettyPrintSkeleton(img.skeletonAry);
        inFile.close();
        outFile.close();
        outFiletwo.close();
        outFilethree.close();
        return 0;
```

}

INPUT

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 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
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

OUTPUT FOR INPUT 1

OUTPUT1

20 40 0 9
0 0 0 0 0 0 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 2 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 2 3 2 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 2 3 4 3 2 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 2 3 4 5 4 3 2 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 2 3 4 5 6 5 4 3 2 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 2 3 4 5 6 7 6 5 4 3 2 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 2 3 4 5 6 7 8 7 6 5 4 3 2 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 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 2 3 4 5 5 6 7 8 9 8 7 6 5 5 4 3 2 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 2 3 4 4 5 6 7 8 7 6 5 4 4 3 2 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 2 3 3 4 5 6 7 6 5 4 3 3 2 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 2 2 3 4 5 6 5 4 3 2 2 1 0 0 0
0 0 0 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 0 0 0 0 0 0 1 1 2 3 4 5 4 3 2 1 1 0 0 0 0
0 0 0 1 2 3 3 3 3 3 3 3 3 3 3 3 3 2 1 0 0 0 0 0 0 0 0 1 2 3 4 3 2 1 0 0 0 0 0 0
0 0 0 1 2 3 4 4 4 4 4 4 4 4 4 4 3 2 1 0 0 0 0 0 0 0 0 0 1 2 3 2 1 0 0 0 0 0 0 0
0 0 0 1 2 3 4 4 4 4 4 4 4 4 4 4 3 2 1 0 0 0 0 0 0 0 0 0 0 1 2 1 0 0 0 0 0 0 0 0
0 0 0 1 2 3 3 3 3 3 3 3 3 3 3 3 3 2 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 2 2 2 2 2 2 2 2 2 2 2 2 2 2 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

OUTPUT2

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 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 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 1 0 0 0 0 1 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 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
                                                          1 2 1
                                                        1 2 3 2 1
                                                      1 2 3 4 3 2 1
                                                    1 2 3 4 5 4 3 2 1
                                                  1 2 3 4 5 6 5 4 3 2 1
                                                1 2 3 4 5 6 7 6 5 4 3 2 1
                                              1 2 3 4 5 6 7 8 7 6 5 4 3 2 1
                                            1 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 1
                                          1 2 3 4 5 6 7 8 9 109 8 7 6 5 4 3 2 1
                                            1 2 3 4 5 6 7 8 9 109 8 7 6 5 4 3
                                              1 2 3 4 5 6 7 8 9 109 8 7 6 5
      1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1           1 2 3 4 5 6 7 8 9 109 8 7
      1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2             1 2 3 4 5 6 7 8 9 109
      1 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3                 1 2 3 4 5 6 7
      1 2 3 4 4 4 4 4 4 4 4 4 4 4 4 4                   1 2 3 4 5
      1 2 3 4 5 5 5 5 5 5 5 5 5 5 5 5                     1 2 3
      1 2 3 4 5 6 6 6 6 6 6 6 6 6 6 6                       1
      1 2 3 4 5 6 7 7 7 7 7 7 7 7 7 7                       1
      1 2 3 4 5 6 7 8 8 8 8 8 8 8 8 8                       1
```

```
Pass-2
                                                            1
                                                          1 2 1
                                                        1 2 3 2 1
                                                      1 2 3 4 3 2 1
                                                    1 2 3 4 5 4 3 2 1
                                                  1 2 3 4 5 6 5 4 3 2 1
                                                1 2 3 4 5 6 7 6 5 4 3 2 1
                                              1 2 3 4 5 6 7 8 7 6 5 4 3 2 1
                                            1 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 1
                                          1 2 3 4 5 5 6 7 8 9 8 7 6 5 5 4 3 2 1
                                            1 2 3 4 4 5 6 7 8 7 6 5 4 4 3 2 1
                                              1 2 3 3 4 5 6 7 6 5 4 3 3 2 1
      1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1           1 2 2 3 4 5 6 5 4 3 2 2 1
      1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1             1 1 2 3 4 5 4 3 2 1 1
      1 2 3 3 3 3 3 3 3 3 3 3 3 3 2 1                 1 2 3 4 3 2 1
      1 2 3 4 4 4 4 4 4 4 4 4 4 3 2 1                   1 2 3 2 1
      1 2 3 4 4 4 4 4 4 4 4 4 4 3 2 1                     1 2 1
      1 2 3 3 3 3 3 3 3 3 3 3 3 3 2 1                       1
      1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1                       1
      1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1                       1
```

....

....

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

OUTPUT FOR INPUT 2

OUTPUT1

```
38 31 0 7
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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 2 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 2 3 2 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 2 3 4 3 2 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 2 3 4 5 4 3 2 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 2 3 4 5 6 5 4 3 2 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 2 3 4 5 6 7 6 5 4 3 2 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 2 3 4 5 6 7 6 5 4 3 2 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 2 3 4 5 6 7 6 5 4 3 2 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 2 3 4 5 6 7 6 5 4 3 2 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 2 3 4 5 6 7 6 5 4 3 2 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 2 3 4 5 6 7 6 5 4 3 2 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 2 2 3 4 5 6 5 4 3 2 2 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 1 2 3 4 5 4 3 2 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 2 3 4 3 2 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 2 3 2 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 2 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 2 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 2 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 2 3 2 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 2 3 4 3 2 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 2 3 4 5 4 3 2 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 2 3 4 5 6 5 4 3 2 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 2 3 4 5 6 7 6 5 4 3 2 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 2 3 4 5 6 7 6 5 4 3 2 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 2 3 4 5 6 7 6 5 4 3 2 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 2 3 4 5 6 7 6 5 4 3 2 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 2 3 4 5 6 7 6 5 4 3 2 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 2 3 4 5 6 7 6 5 4 3 2 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 2 2 3 4 5 6 5 4 3 2 2 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 1 2 3 4 5 4 3 2 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 2 3 4 3 2 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 2 3 2 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 2 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 2 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

OUTPUT2

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 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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 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 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 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 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 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 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 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

OUTPUT3

Pass-1

1
1 2 1
1 2 3 2 1
1 2 3 4 3 2 1
1 2 3 4 5 4 3 2 1
1 2 3 4 5 6 5 4 3 2 1
1 2 3 4 5 6 7 6 5 4 3 2 1
1 2 3 4 5 6 7 7 6 5 4 3 2
1 2 3 4 5 6 7 8 7 6 5 4 3
1 2 3 4 5 6 7 8 8 7 6 5 4
1 2 3 4 5 6 7 8 9 8 7 6 5
1 2 3 4 5 6 7 8 9 9 8 7 6
1 2 3 4 5 6 7 8 9 109 8 7
1 2 3 4 5 6 7 8 9 109
1 2 3 4 5 6 7
1 2 3 4 5
1 2 3
1 2 3
1 2 3
1 2 3 4 1
1 2 3 4 5 2 1
1 2 3 4 5 6 3 2 1
1 2 3 4 5 6 7 4 3 2 1
1 2 3 4 5 6 7 8 5 4 3 2 1
1 2 3 4 5 6 7 8 6 5 4 3 2
1 2 3 4 5 6 7 8 7 6 5 4 3
1 2 3 4 5 6 7 8 8 7 6 5 4
1 2 3 4 5 6 7 8 9 8 7 6 5
1 2 3 4 5 6 7 8 9 9 8 7 6
1 2 3 4 5 6 7 8 9 109 8 7
1 2 3 4 5 6 7 8 9 109
1 2 3 4 5 6 7
1 2 3 4 5
1 2 3
1 2 3
1

Pass-2

1
1 2 1
1 2 3 2 1

1 2 3 4 3 2 1
1 2 3 4 5 4 3 2 1
1 2 3 4 5 6 5 4 3 2 1
1 2 3 4 5 6 7 6 5 4 3 2 1
1 2 3 4 5 6 7 6 5 4 3 2 1
1 2 3 4 5 6 7 6 5 4 3 2 1
1 2 3 4 5 6 7 6 5 4 3 2 1
1 2 3 4 5 6 7 6 5 4 3 2 1
1 2 3 4 5 6 7 6 5 4 3 2 1
1 2 2 3 4 5 6 5 4 3 2 2 1
1 1 2 3 4 5 4 3 2 1 1
1 2 3 4 3 2 1
1 2 3 2 1
1 2 1
1 2 1
1 2 1
1 2 3 2 1
1 2 3 4 3 2 1
1 2 3 4 5 4 3 2 1
1 2 3 4 5 6 5 4 3 2 1
1 2 3 4 5 6 7 6 5 4 3 2 1
1 2 3 4 5 6 7 6 5 4 3 2 1
1 2 3 4 5 6 7 6 5 4 3 2 1
1 2 3 4 5 6 7 6 5 4 3 2 1
1 2 3 4 5 6 7 6 5 4 3 2 1
1 2 3 4 5 6 7 6 5 4 3 2 1
1 2 3 4 5 6 7 6 5 4 3 2 1
1 2 2 3 4 5 6 5 4 3 2 2 1
1 1 2 3 4 5 4 3 2 1 1
1 2 3 4 3 2 1
1 2 3 2 1
1 2 1
1 2 1
1

.....
.....
.....
.....
.....
.....
.....
.....9.....
.....9.....

.....9.....
.....9.....
.....9.....
.....9.....
.....
.....
.....
.....
.....
.....9.....
.....
.....
.....
.....
.....
.....9.....
.....9.....
.....9.....
.....9.....
.....9.....
.....9.....
.....
.....
.....
.....
.....
.....9.....
.....
.....