Cover Sheet

CV Project5: Image Compression via Distance Transform Java

Student: Fengzhang Du

Project Due Date: 03/30/2021

Algorithm Steps for Compute Image Compression:

```
step 0: inFile <- open input file
        numRows, numCols, minVal, maxVal <- read from inFile
        dynamically allocate zeroFramedAry with extra 2 rows and 2 cols
        dynamically allocate skeletonAry with extra 2 rows and 2 cols
        open outFile_1, outFile_2
Step 1: skeletonFileName <- args[0] + "_skeleton.txt"
Step 2: skeletonFile <- open( skeletonFileName )
Step 3: decompressedFileName <- args[0] + "_decompressed.txt"
Step 4: decompressFile <- open (decompressedFileName)
Step 5: loadImage (inFile, zeroFramedAry)
Step 6: compute8Distance (zeroFramedAry, outFile1)
Step 7: skeletonExtraction (zeroFramedAry, skeletonAry, skeletonFile,
        outFile1) // perform lossless compression
Step 8: skeletonExpansion (zeroFramedAry, skeletonFile, outFile2)
        // perform decompression
step 9: Output numRows, numCols, newMinVal, newMaxVal to decompressFile
Step 10: ary2File (zeroFramedAry, decompressFile)
Step 11: close all files
```

Source Code

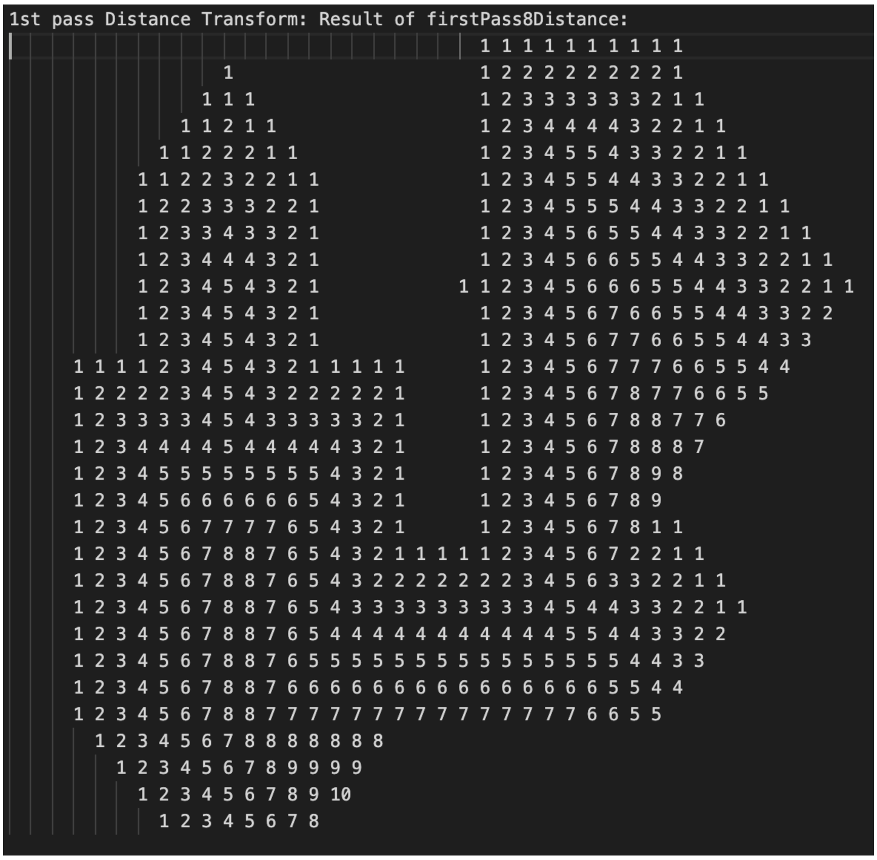
```
import java.io.*;
import java.util.Scanner;

public class Main{
   public static void main(String[] args) throws IOException, InterruptedException{
       String fileName = args[0].replace(".txt", "");
       String skeleton_name = fileName+"_skeleton.txt";
       String decompressed_name = fileName+"_decompressed.txt";
       try(
           Scanner input = new Scanner(new BufferedReader(new FileReader(args[0])));
           // 2 output files
           BufferedWriter output1 = new BufferedWriter(new FileWriter(args[1], true));
           BufferedWriter output2 = new BufferedWriter(new FileWriter(args[2]));


           BufferedWriter skeletonFile = new BufferedWriter(new FileWriter(skeleton_name, true));
           BufferedWriter decompressedFile = new BufferedWriter(new
FileWriter(decompressed_name));
           // open the compressed skeleton
           Scanner skeletonFileReader = new Scanner(new BufferedReader(new
FileReader(skeleton_name)));
       ){
           // Read and store image header.
           int header[] = new int[4];
           for (int i=0; i<4; i++){
               if (input.hasNextInt()) header[i] = input.nextInt();
           }
           ImageProcessing img = new ImageProcessing(header[0], header[1], header[2], header[3]);
           img.loadImg(input);
           img.compute8Distance(output1);
           img.skeletonExtraction(output1, skeletonFile);
           img.skeletonExpansion(output2, skeletonFileReader);
           img.ary2File(decompressedFile, output2);
       }
   }
}

class ImageProcessing{
   // field
   int numRows=0, numCols=0, minVal=0, maxVal=0, newMin=0, newMax=0;
   int[][] zeroFramedAry;
   int[][] skeletonAry;
   int f = 1; // frame size

   // constructor
   ImageProcessing(int numRows, int numCols, int minVal, int maxVal){
       this.numRows = numRows;
       this.numCols = numCols;
```

```
        this.minVal = minVal;
        this.maxVal = maxVal;
    }

    // methods
    void loadImg(Scanner input){
        this.zeroFramedAry = new int[this.numRows+2][this.numCols+2];
        this.skeletonAry = new int[this.numRows+2][this.numCols+2];
        for(int i=f; i<numRows+f; i++){
            for(int j=f; j<numCols+f; j++){
                if(input.hasNextInt()) zeroFramedAry[i][j] = input.nextInt();
                else{
                    System.out.println( "Corrupted Image input data!");
                    System.exit(0);
                }
            }
        }
    }

    void compute8Distance(BufferedWriter output) throws IOException{
        // all 4 methods only involve zeroFramedAry
        firstPass8Distance();
        reformatPrettyPrint("1st pass Distance Transform: Result of firstPass8Distance: ",
zeroFramedAry, output);
        secondPass8Distance();
        reformatPrettyPrint("2nd pass Distance Transform: Result of secondPass8Distance: ",
zeroFramedAry, output);
    }

    void firstPass8Distance(){
        for (int i=f; i<numRows+f; i++){
            for (int j=f; j<numCols+f; j++){
                int tempMin = 10000;
                if (zeroFramedAry[i][j] > 0){
                    // loop through all the neighbors
                    for (int k=i-1; k<=i; k++){
                        for (int d=j-1; d<=j+1; d++){
                            if (k >= i && d >=j) break;
                            else{
                                tempMin = Math.min(tempMin, zeroFramedAry[k][d]);
                            }
                        }
                    }
                    zeroFramedAry[i][j] = tempMin+1;
                }
            }
        }
    }

    void secondPass8Distance(){
        newMax = 0;
        for (int i=numRows; i>=f; i--){
            for (int j=numCols; j>=f; j--){
                if (zeroFramedAry[i][j] > 0){
                    // loop through all the neighbors
                    for (int k=i+1; k>=i; k--){
                        for(int d=j+1;d>=j-1; d--){
                            if(k<=i && d<=j) break;
                            else{
                                zeroFramedAry[i][j] = Math.min(zeroFramedAry[i][j],
zeroFramedAry[k][d]+1);
                                newMin = Math.min(newMin, zeroFramedAry[k][d]);
                                newMax = Math.max(newMax, zeroFramedAry[k][d]);
                            }
                        }
                    }
                }
            }
        }
    }

    boolean isLocalMaxima(int i, int j){
        // loop through all the neighbors
        for (int k=i-1; k<=i+1; k++){
            for (int d=j-1; d<=j+1; d++){
                if(zeroFramedAry[i][j]<zeroFramedAry[k][d]) {
                    return false;
                }
            }
        }
        return true;
    }

    void computeLocalMaxima() throws IOException{
        for (int i=f; i<numRows+f; i++){
            for (int j=f; j<numCols+f; j++){
                if (isLocalMaxima(i, j)){
                    skeletonAry[i][j] = zeroFramedAry[i][j];
                }else{
                    skeletonAry[i][j] = 0;
                }
            }
        }
    }

    void extractLocalMaxima(BufferedWriter output) throws IOException{
        output.write(Integer.toString(numRows) + " " + Integer.toString(numCols) + " ");
        output.write(Integer.toString(newMin) + " " + Integer.toString(newMax) + "\n");
        for (int i=f; i<=numRows; i++){
            for(int j=f; j<=numCols; j++){
                if (skeletonAry[i][j] > 0){
                    output.write(i+" " + j+" "+skeletonAry[i][j]+"\n");
                }
            }
        }
        output.close();

    }

    void skeletonExtraction(BufferedWriter output1, BufferedWriter skeletonFile) throws
IOException{
        computeLocalMaxima();
        reformatPrettyPrint("Local Maxima: Result of  computeLocalMaxima;", skeletonAry, output1);
        extractLocalMaxima(skeletonFile);
    }

    void skeletonExpansion(BufferedWriter output2, Scanner skeletonFileReader) throws IOException,
InterruptedException{
        // set array to all zeros.
        this.zeroFramedAry = new int[this.numRows+2][this.numCols + 2];
        this.skeletonAry = new int[this.numRows+2][this.numCols + 2];
        loadSkeleton(output2, skeletonFileReader);
        firstPassExpension();
        reformatPrettyPrint("1st pass Expansion: Result of firstPassExpension:", zeroFramedAry,
output2);
        secondPassExpension();
        reformatPrettyPrint("2nd pass Expansion: Result of secondPassExpension:", zeroFramedAry,
output2);
    }

    void loadSkeleton(BufferedWriter output2, Scanner skeletonFileReader) throws IOException{
        // load header from compressed skeletonFileReader
        output2.write("Compressed Skeleton: \n");
        int newHeader[] = new int[4];
        for (int i=0; i<4; i++){
            if (skeletonFileReader.hasNextInt()) {
                newHeader[i] = skeletonFileReader.nextInt();
                output2.write(newHeader[i] + " ");
            }
        }
        output2.write("\n");

        while(skeletonFileReader.hasNextInt()){
            int i = skeletonFileReader.nextInt();
            int j = skeletonFileReader.nextInt();
            zeroFramedAry[i][j] = skeletonFileReader.nextInt();
            output2.write(i +" " + j + " " + zeroFramedAry[i][j] + "\n");
        }
        output2.write("\n");
    }

    void firstPassExpension(){
        for(int i=f; i<=numRows; i++){
            for(int j=f; j<=numCols; j++){
                if (zeroFramedAry[i][j] == 0){
                    // loop through all neighbors.
                    for (int k=i-1; k<=i+1; k++){
                        for (int d=j-1; d<=j+1; d++){
                            if (k==i && d==j) continue;
                            else{
                                zeroFramedAry[i][j] = Math.max(zeroFramedAry[i][j], 
 zeroFramedAry[k][d]-1);
                            }
                        }
                    }
                }
            }
        }
    }

    void secondPassExpension(){
        for(int i=numRows; i>=f; i--){
            for(int j=numCols; j>=f; j--){
                // loop through all neighbors for all pixels.
                int tempMax = 0;
                for (int k=i+1; k>=i-1; k--){
                    for (int d=j+1; d>=j-1; d--){
                        if (k==i && d==j) continue;
                        else{
                            tempMax = Math.max(tempMax, zeroFramedAry[k][d]);
                        }
                    }
                }
                if(zeroFramedAry[i][j]<tempMax) zeroFramedAry[i][j] = tempMax-1;
            }
        }
    }

    void ary2File(BufferedWriter decompressedFile, BufferedWriter output2) throws IOException{ // 
 to decompressed file
        decompressedFile.write(numRows + " " + numCols + " " + minVal + " " + maxVal + "\n");
        output2.write("\nDecompressed File:\n");
        output2.write(numRows + " " + numCols + " " + minVal + " " + maxVal + "\n");
        for(int i=f; i<=numRows; i++){
            for (int j=f; j<=numCols; j++){
                if (zeroFramedAry[i][j] >= 1){
                    decompressedFile.write("1 ");
                    output2.write("1 ");
                }else{
                    decompressedFile.write("0 ");
                    output2.write("0 ");
                }
            }
            decompressedFile.write("\n");
```

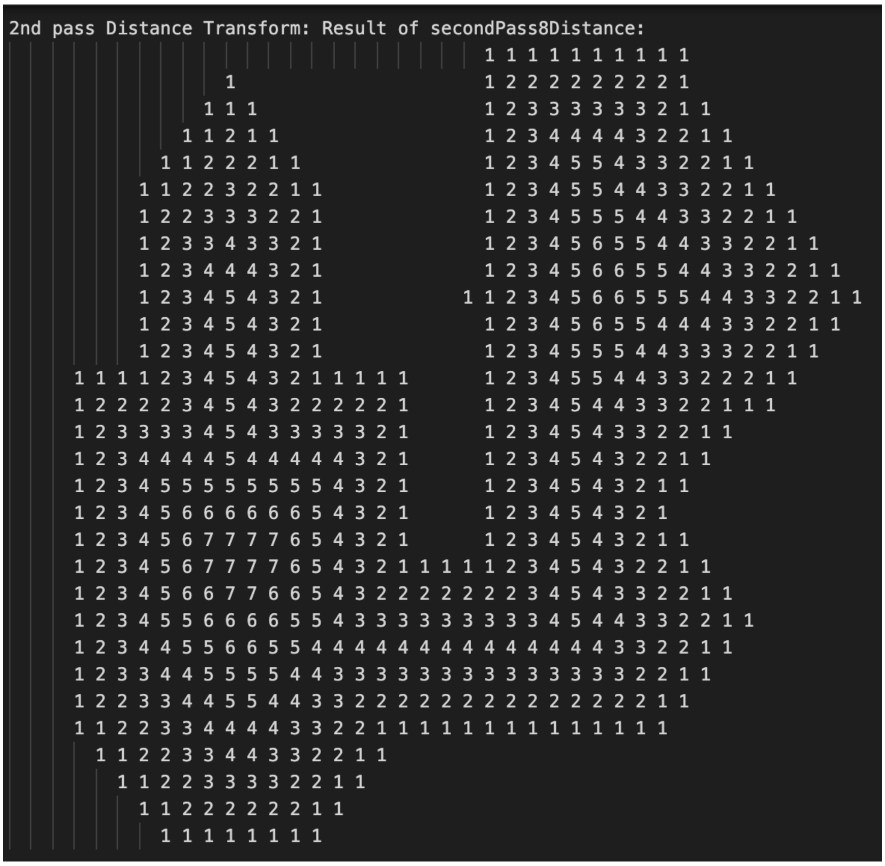
```
            output2.write("\n");
        }
    }

    void reformatPrettyPrint(String title, int [][] arr, BufferedWriter output) throws IOException
{
        output.write(title + "\n");
        for(int i=f; i<numRows+f; i++){
            for(int j=f; j<numCols+f; j++){
                if(arr[i][j] == 0){
                    output.write(" " + " ");
                }else{
                    output.write(Integer.toString(arr[i][j]) + " ");
                }
            }
            output.write("\n");
        }
        output.write("\n");
    }
}
```

Program Output

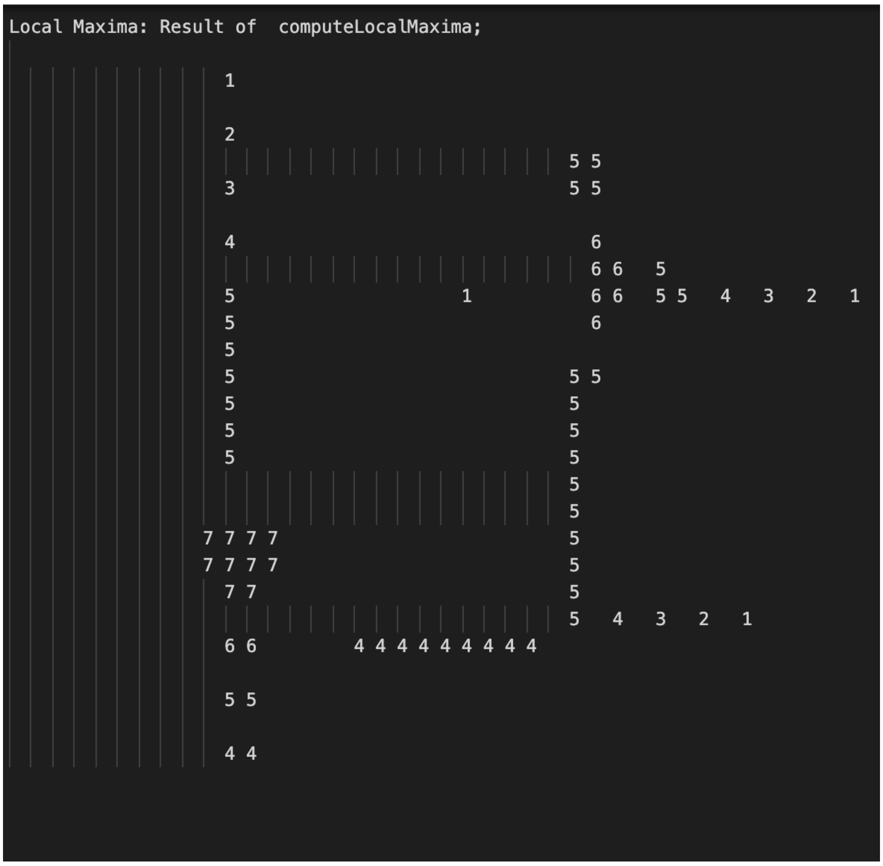
image1_output_1.txt



```
1st pass Distance Transform: Result of firstPass8Distance:
                                            1 1 1 1 1 1 1 1 1 1
                    1                       1 2 2 2 2 2 2 2 2 1
                  1 1 1                     1 2 3 3 3 3 3 3 2 1 1
                1 1 2 1 1                   1 2 3 4 4 4 4 3 2 2 1 1
              1 1 2 2 2 1 1                 1 2 3 4 5 5 4 3 3 2 2 1 1
            1 1 2 2 3 2 2 1 1               1 2 3 4 5 5 4 4 3 3 2 2 1 1
            1 2 2 3 3 3 2 2 1               1 2 3 4 5 5 5 4 4 3 3 2 2 1 1
            1 2 3 3 4 3 3 2 1               1 2 3 4 5 6 5 5 4 4 3 3 2 2 1 1
            1 2 3 4 4 4 3 2 1               1 2 3 4 5 6 6 5 5 4 4 3 3 2 2 1 1
            1 2 3 4 5 4 3 2 1             1 1 2 3 4 5 6 6 6 5 5 4 4 3 3 2 2 1 1
            1 2 3 4 5 4 3 2 1               1 2 3 4 5 6 7 6 6 5 5 4 4 3 3 2 2
            1 2 3 4 5 4 3 2 1               1 2 3 4 5 6 7 7 6 6 5 5 4 4 3 3
      1 1 1 1 2 3 4 5 4 3 2 1 1 1 1 1       1 2 3 4 5 6 7 7 7 6 6 5 5 4 4
      1 2 2 2 2 3 4 5 4 3 2 2 2 2 2 1       1 2 3 4 5 6 7 8 7 7 6 6 5 5
      1 2 3 3 3 3 4 5 4 3 3 3 3 3 2 1       1 2 3 4 5 6 7 8 8 7 7 6
      1 2 3 4 4 4 4 5 4 4 4 4 4 3 2 1       1 2 3 4 5 6 7 8 8 8 7
      1 2 3 4 5 5 5 5 5 5 5 5 4 3 2 1       1 2 3 4 5 6 7 8 9 8
      1 2 3 4 5 6 6 6 6 6 6 5 4 3 2 1       1 2 3 4 5 6 7 8 9
      1 2 3 4 5 6 7 7 7 7 6 5 4 3 2 1       1 2 3 4 5 6 7 8 1 1
      1 2 3 4 5 6 7 8 8 7 6 5 4 3 2 1 1 1 1 1 2 3 4 5 6 7 2 2 1 1
      1 2 3 4 5 6 7 8 8 7 6 5 4 3 2 2 2 2 2 2 2 3 4 5 6 3 3 2 2 1 1
      1 2 3 4 5 6 7 8 8 7 6 5 4 3 3 3 3 3 3 3 3 3 4 5 4 4 3 3 2 2 1 1
      1 2 3 4 5 6 7 8 8 7 6 5 4 4 4 4 4 4 4 4 4 4 4 5 5 4 4 3 3 2 2
      1 2 3 4 5 6 7 8 8 7 6 5 5 5 5 5 5 5 5 5 5 5 5 5 5 4 4 3 3
      1 2 3 4 5 6 7 8 8 7 6 6 6 6 6 6 6 6 6 6 6 6 6 6 5 5 4 4
      1 2 3 4 5 6 7 8 8 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 6 6 5 5
        1 2 3 4 5 6 7 8 8 8 8 8 8 8
          1 2 3 4 5 6 7 8 9 9 9 9
            1 2 3 4 5 6 7 8 9 10
              1 2 3 4 5 6 7 8
```



```
2nd pass Distance Transform: Result of secondPass8Distance:
                                            1 1 1 1 1 1 1 1 1 1
                    1                       1 2 2 2 2 2 2 2 2 1
                  1 1 1                     1 2 3 3 3 3 3 3 2 1 1
                1 1 2 1 1                   1 2 3 4 4 4 4 3 2 2 1 1
              1 1 2 2 2 1 1                 1 2 3 4 5 5 4 3 3 2 2 1 1
            1 1 2 2 3 2 2 1 1               1 2 3 4 5 5 4 4 3 3 2 2 1 1
            1 2 2 3 3 3 2 2 1               1 2 3 4 5 5 5 4 4 3 3 2 2 1 1
            1 2 3 3 4 3 3 2 1               1 2 3 4 5 6 5 5 4 4 3 3 2 2 1 1
            1 2 3 4 4 4 3 2 1               1 2 3 4 5 6 6 5 5 4 4 3 3 2 2 1 1
            1 2 3 4 5 4 3 2 1             1 1 2 3 4 5 6 6 5 5 5 4 4 3 3 2 2 1 1
            1 2 3 4 5 4 3 2 1               1 2 3 4 5 6 5 5 4 4 4 3 3 2 2 1 1
            1 2 3 4 5 4 3 2 1               1 2 3 4 5 5 5 4 4 3 3 3 2 2 1 1
      1 1 1 1 2 3 4 5 4 3 2 1 1 1 1 1       1 2 3 4 5 5 4 4 3 3 2 2 2 1 1
      1 2 2 2 2 3 4 5 4 3 2 2 2 2 2 1       1 2 3 4 5 4 4 3 3 2 2 1 1 1
      1 2 3 3 3 3 4 5 4 3 3 3 3 3 2 1       1 2 3 4 5 4 3 3 2 2 1 1
      1 2 3 4 4 4 4 5 4 4 4 4 4 3 2 1       1 2 3 4 5 4 3 2 2 1 1
      1 2 3 4 5 5 5 5 5 5 5 5 4 3 2 1       1 2 3 4 5 4 3 2 1 1
      1 2 3 4 5 6 6 6 6 6 6 5 4 3 2 1       1 2 3 4 5 4 3 2 1
      1 2 3 4 5 6 7 7 7 7 6 5 4 3 2 1       1 2 3 4 5 4 3 2 1 1
      1 2 3 4 5 6 7 7 7 7 6 5 4 3 2 1 1 1 1 1 2 3 4 5 4 3 2 2 1 1
      1 2 3 4 5 6 6 7 7 6 6 5 4 3 2 2 2 2 2 2 2 3 4 5 4 3 3 2 2 1 1
      1 2 3 4 5 5 6 6 6 6 5 5 4 3 3 3 3 3 3 3 3 3 4 5 4 4 3 3 2 2 1 1
      1 2 3 4 4 5 5 6 6 5 5 4 4 4 4 4 4 4 4 4 4 4 4 4 4 3 3 2 2 1 1
      1 2 3 3 4 4 5 5 5 5 4 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 1 1
      1 2 2 3 3 4 4 5 5 4 4 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1
      1 1 2 2 3 3 4 4 4 4 3 3 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1
        1 1 2 2 3 3 4 4 3 3 2 2 1 1
          1 1 2 2 3 3 3 3 2 2 1 1
            1 1 2 2 2 2 2 2 1 1
              1 1 1 1 1 1 1 1
```

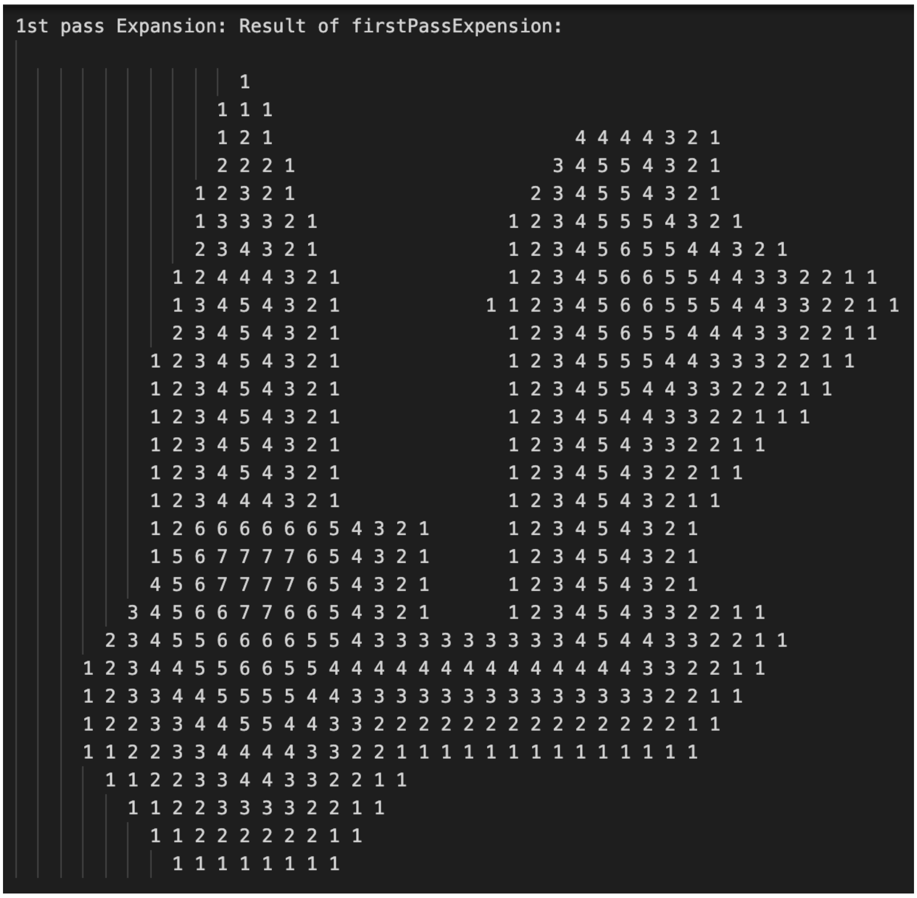


```
Local Maxima: Result of  computeLocalMaxima;

                    1

                    2
                                                    5 5
                    3                               5 5

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

                    5 5

                    4 4
```

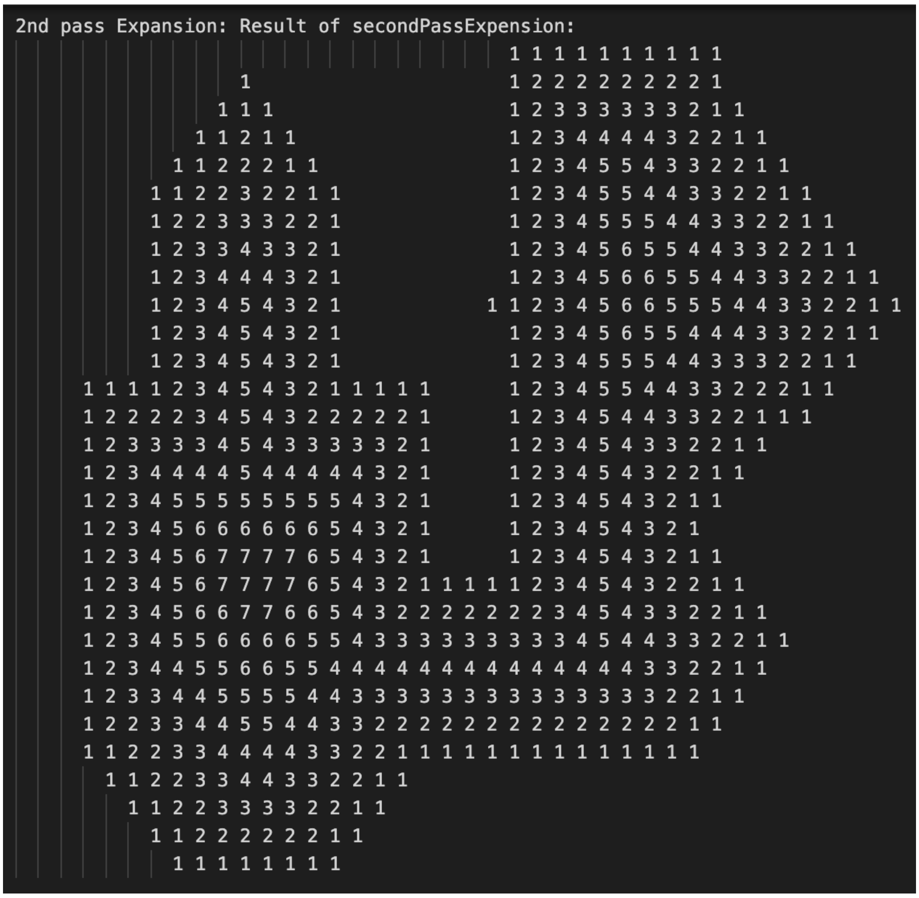
image1_output_2.txt

Compressed Skeleton:

```
30 40 0 7
2 11 1
4 11 2
5 27 5
5 28 5
6 11 3
6 27 5
6 28 5
8 11 4
8 28 6
9 28 6
9 29 6
9 31 5
10 11 5
10 22 1
10 28 6
10 29 6
10 31 5
10 32 5
10 34 4
10 36 3
10 38 2
10 40 1
11 11 5
11 28 6
12 11 5
13 11 5
13 27 5
13 28 5
14 11 5
14 27 5
15 11 5
15 27 5
16 11 5
16 27 5
17 27 5
18 27 5
19 10 7
19 11 7
19 12 7
19 13 7
19 27 5
20 10 7
20 11 7
20 12 7
20 13 7
20 27 5
21 11 7
21 12 7
21 27 5
22 27 5
22 29 4
22 31 3
22 33 2
22 35 1
23 11 6
```

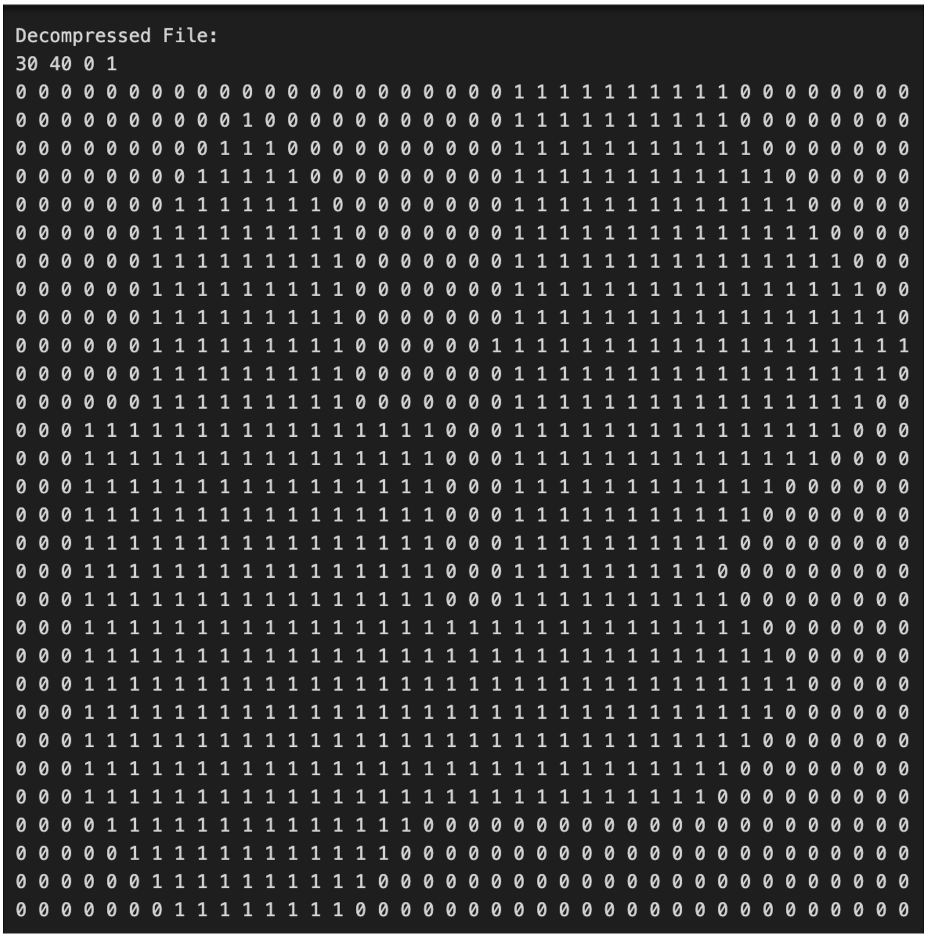


```
1st pass Expansion: Result of firstPassExpension:

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



```
2nd pass Expansion: Result of secondPassExpension:
                                            1 1 1 1 1 1 1 1 1 1
                    1                       1 2 2 2 2 2 2 2 2 1
                  1 1 1                     1 2 3 3 3 3 3 3 2 1 1
                1 1 2 1 1                   1 2 3 4 4 4 4 3 2 2 1 1
              1 1 2 2 2 1 1                 1 2 3 4 5 5 4 3 3 2 2 1 1
            1 1 2 2 3 2 2 1 1               1 2 3 4 5 5 4 4 3 3 2 2 1 1
            1 2 2 3 3 3 2 2 1               1 2 3 4 5 5 5 4 4 3 3 2 2 1 1
            1 2 3 3 4 3 3 2 1               1 2 3 4 5 6 5 5 4 4 3 3 2 2 1 1
            1 2 3 4 4 4 3 2 1               1 2 3 4 5 6 6 5 5 4 4 3 3 2 2 1 1
            1 2 3 4 5 4 3 2 1             1 1 2 3 4 5 6 6 5 5 5 4 4 3 3 2 2 1 1
            1 2 3 4 5 4 3 2 1               1 2 3 4 5 6 5 5 4 4 4 3 3 2 2 1 1
            1 2 3 4 5 4 3 2 1               1 2 3 4 5 5 5 4 4 3 3 3 2 2 1 1
      1 1 1 1 2 3 4 5 4 3 2 1 1 1 1 1       1 2 3 4 5 5 4 4 3 3 2 2 2 1 1
      1 2 2 2 2 3 4 5 4 3 2 2 2 2 2 1       1 2 3 4 5 4 4 3 3 2 2 1 1 1
      1 2 3 3 3 3 4 5 4 3 3 3 3 3 2 1       1 2 3 4 5 4 3 3 2 2 1 1
      1 2 3 4 4 4 4 5 4 4 4 4 4 3 2 1       1 2 3 4 5 4 3 2 2 1 1
      1 2 3 4 5 5 5 5 5 5 5 5 4 3 2 1       1 2 3 4 5 4 3 2 1 1
      1 2 3 4 5 6 6 6 6 6 6 5 4 3 2 1       1 2 3 4 5 4 3 2 1
      1 2 3 4 5 6 7 7 7 7 6 5 4 3 2 1       1 2 3 4 5 4 3 2 1 1
      1 2 3 4 5 6 7 7 7 7 6 5 4 3 2 1 1 1 1 1 2 3 4 5 4 3 2 2 1 1
      1 2 3 4 5 6 6 7 7 6 6 5 4 3 2 2 2 2 2 2 2 3 4 5 4 3 3 2 2 1 1
      1 2 3 4 5 5 6 6 6 6 5 5 4 3 3 3 3 3 3 3 3 3 4 5 4 4 3 3 2 2 1 1
      1 2 3 4 4 5 5 6 6 5 5 4 4 4 4 4 4 4 4 4 4 4 4 4 4 3 3 2 2 1 1
      1 2 3 3 4 4 5 5 5 5 4 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 1 1
      1 2 2 3 3 4 4 5 5 4 4 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1
      1 1 2 2 3 3 4 4 4 4 3 3 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1
        1 1 2 2 3 3 4 4 3 3 2 2 1 1
          1 1 2 2 3 3 3 3 2 2 1 1
            1 1 2 2 2 2 2 2 1 1
              1 1 1 1 1 1 1 1
```

```
23 12 6
23 17 4
23 18 4
23 19 4
23 20 4
23 21 4
23 22 4
23 23 4
23 24 4
23 25 4
25 11 5
25 12 5
27 11 4
27 12 4
```

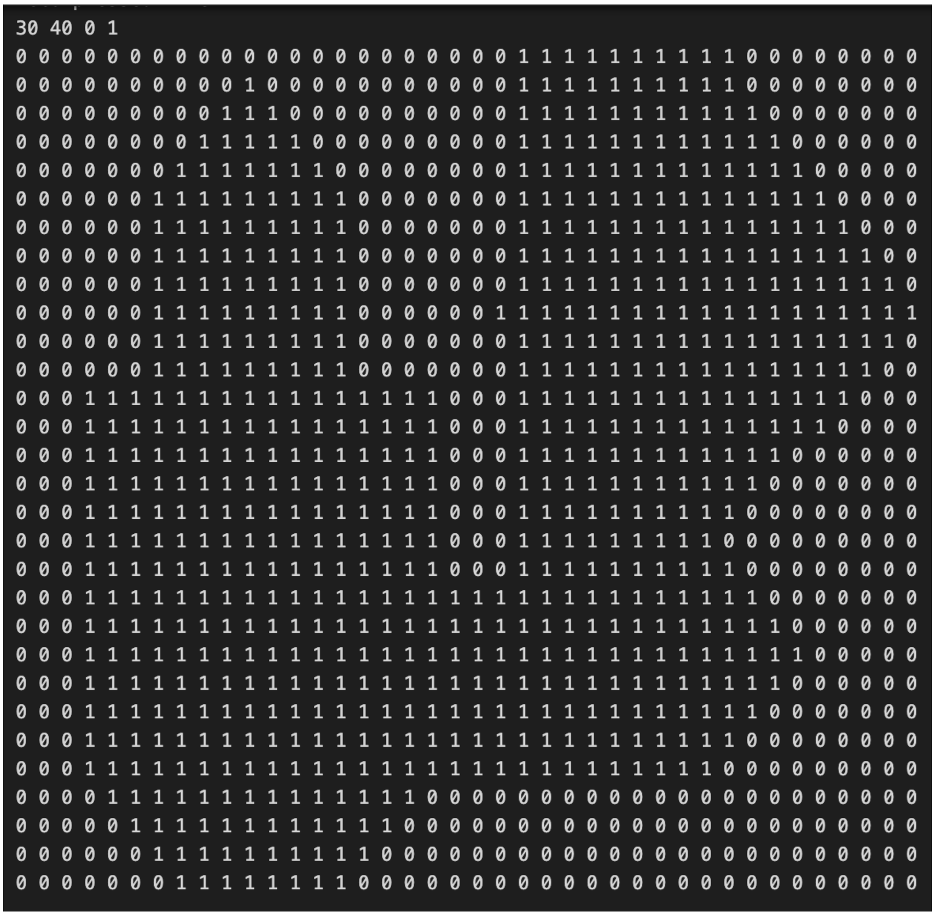


```
Decompressed File:
30 40 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 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 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 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 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 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 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 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 1 0
0 0 0 0 0 0 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 1 1 1
0 0 0 0 0 0 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 1 0
0 0 0 0 0 0 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 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 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 1 1 1 1 1 0 0 0 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 1 1 1 1 1 1 1 0 0 0 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 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 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 1 1 1 1 1 1 1 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 1 1 1 1 1 1 1 1 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 1 1 1 1 1 1 1 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 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 1 1 1 1 1 1 1 1 1 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 1 1 1 1 1 1 1 1 1 1 1 1 1 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 1 1 1 1 1 1 1 1 1 1 1 1 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 1 1 1 1 1 1 1 1 1 1 1 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 1 1 1 1 1 1 1 1 1 1 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 1 1 1 1 1 1 1 1 1 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 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 0 0 0 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 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 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
```

image1_ decompressed .txt

```
30 40 0 7
2 11 1
4 11 2
5 27 5
5 28 5
6 11 3
6 27 5
6 28 5
8 11 4
8 28 6
9 28 6
9 29 6
9 31 5
10 11 5
10 22 1
10 28 6
10 29 6
10 31 5
10 32 5
10 34 4
10 36 3
10 38 2
10 40 1
11 11 5
11 28 6
```

image1_decompressed.txt

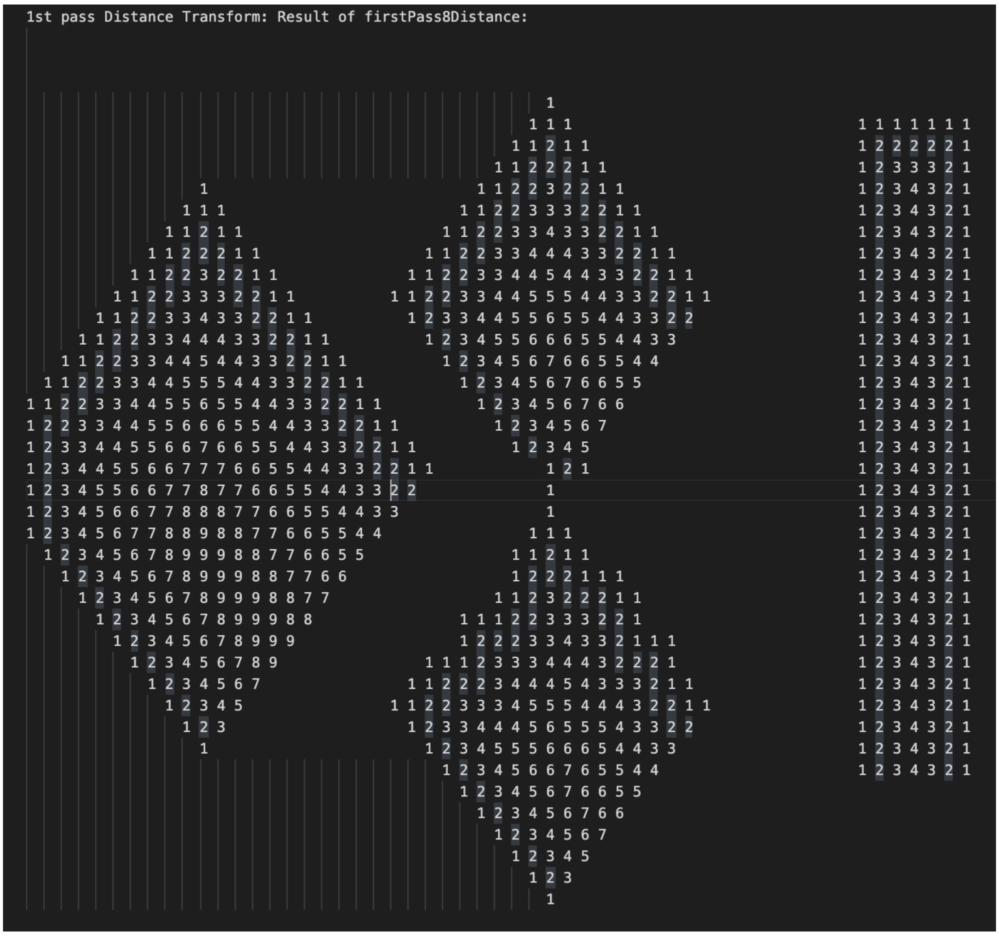


```
30 40 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 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 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 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 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 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 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 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 1 0
0 0 0 0 0 0 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 1 1 1
0 0 0 0 0 0 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 1 0
0 0 0 0 0 0 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 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 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 1 1 1 1 1 0 0 0 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 1 1 1 1 1 1 1 0 0 0 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 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 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 1 1 1 1 1 1 1 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 1 1 1 1 1 1 1 1 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 1 1 1 1 1 1 1 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 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 1 1 1 1 1 1 1 1 1 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 1 1 1 1 1 1 1 1 1 1 1 1 1 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 1 1 1 1 1 1 1 1 1 1 1 1 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 1 1 1 1 1 1 1 1 1 1 1 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 1 1 1 1 1 1 1 1 1 1 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 1 1 1 1 1 1 1 1 1 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 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 0 0 0 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 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 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
```

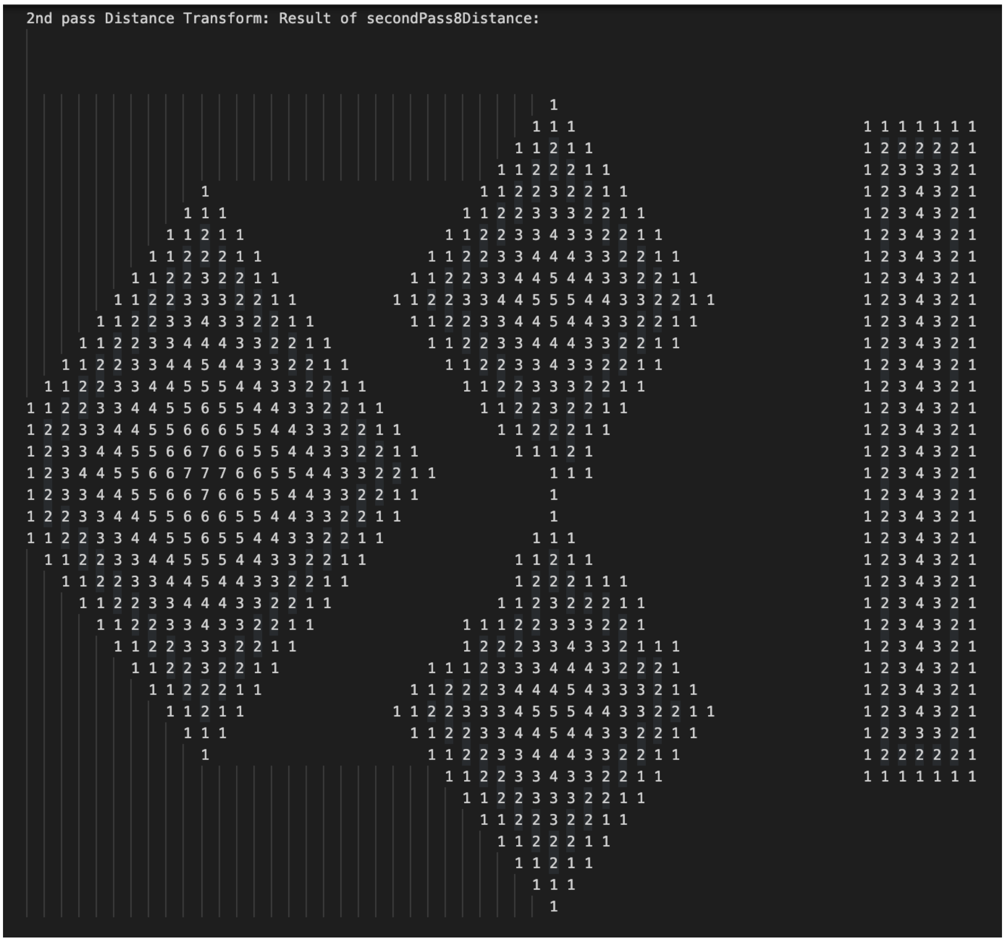
```
12 11 5
13 11 5
13 27 5
13 28 5
14 11 5
14 27 5
15 11 5
15 27 5
16 11 5
16 27 5
17 27 5
18 27 5
19 10 7
19 11 7
19 12 7
19 13 7
19 27 5
20 10 7
20 11 7
20 12 7
20 13 7
20 27 5
21 11 7
21 12 7
21 27 5
22 27 5
22 29 4
22 31 3
22 33 2
22 35 1
23 11 6
23 12 6
23 17 4
23 18 4
23 19 4
23 20 4
23 21 4
23 22 4
23 23 4
23 24 4
23 25 4
25 11 5
25 12 5
27 11 4
27 12 4
```

Next page for image 2.

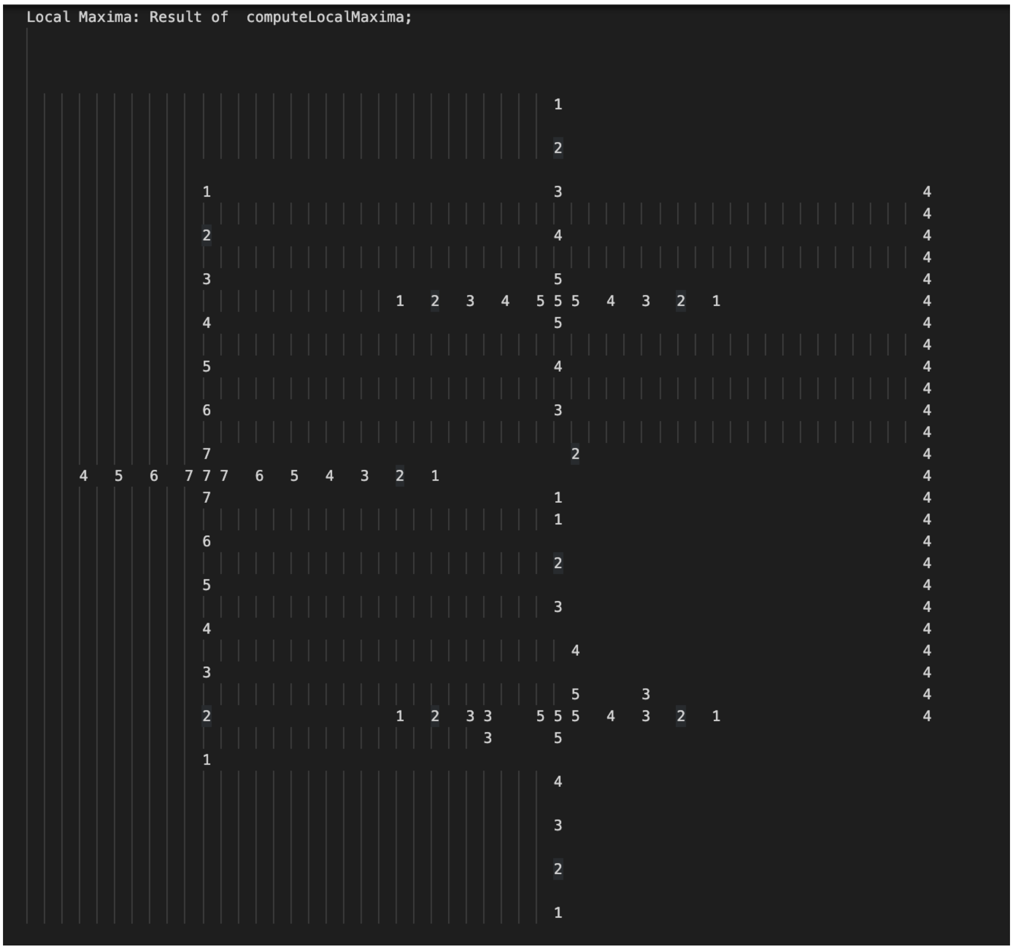
image2_output_1.txt



```
1st pass Distance Transform: Result of firstPass8Distance:



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



```
2nd pass Distance Transform: Result of secondPass8Distance:



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

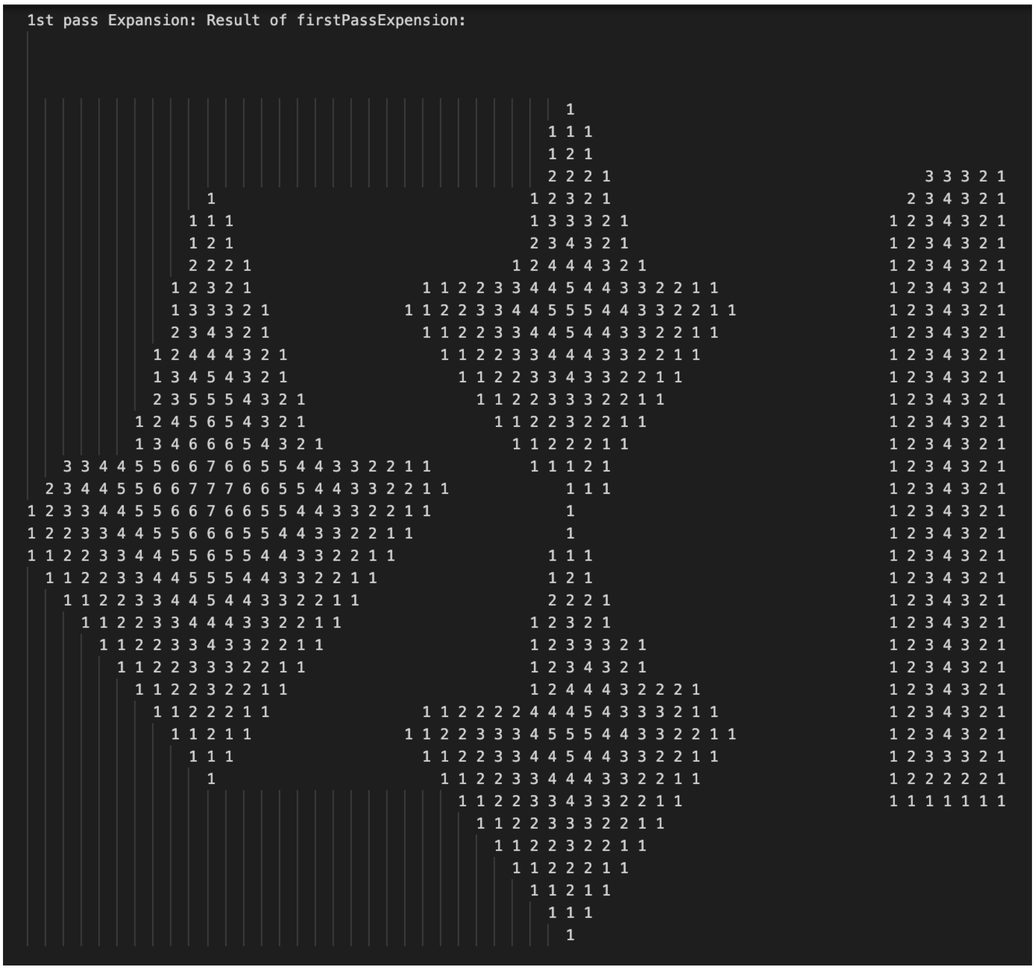


```
Local Maxima: Result of  computeLocalMaxima;



                                                            1

                                                            2

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

                                                            3

                                                            2

                                                            1
```

image2_output_2.txt

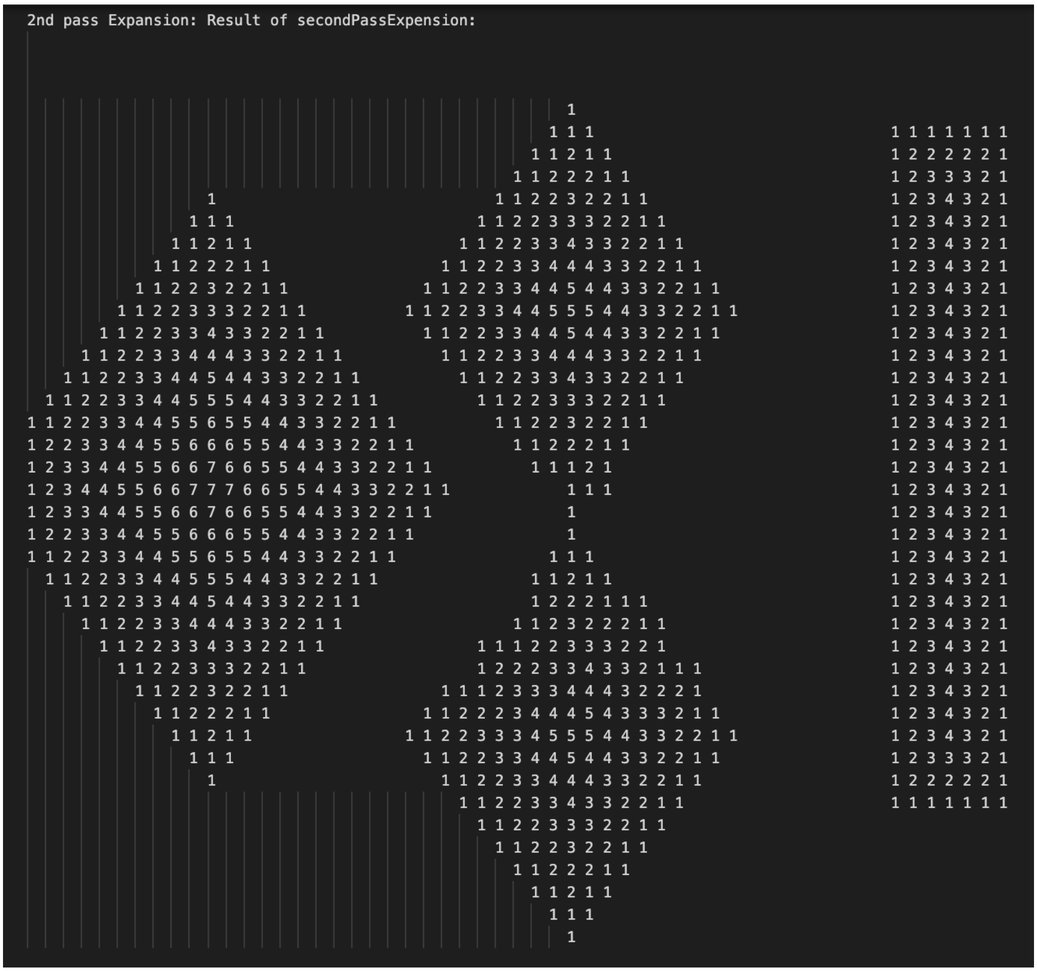
Compressed Skeleton:

```
45 64 0 7
4 31 1
6 31 2
8 11 1
8 31 3
8 52 4
9 52 4
10 11 2
10 31 4
10 52 4
11 52 4
12 11 3
12 31 5
12 52 4
13 22 1
13 24 2
13 26 3
13 28 4
13 30 5
13 31 5
13 32 5
13 34 4
13 36 3
13 38 2
13 40 1
13 52 4
```

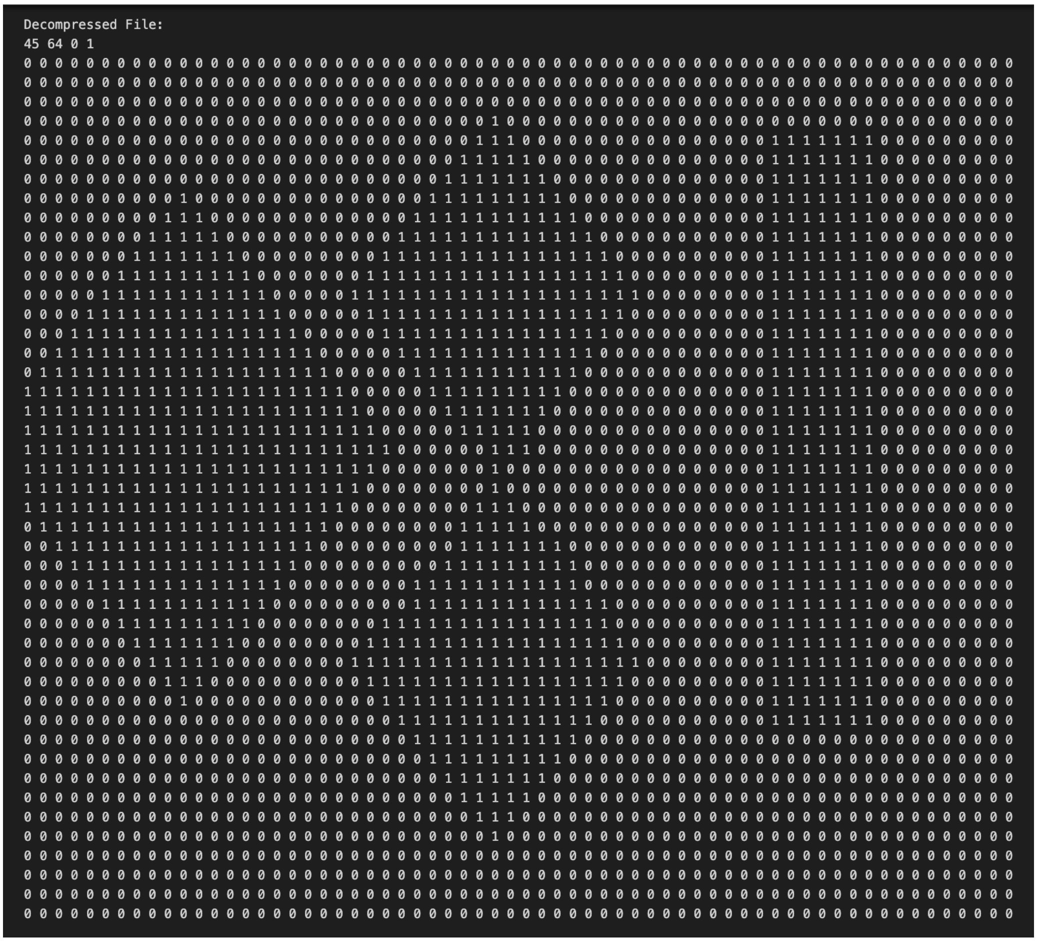


```
1st pass Expansion: Result of firstPassExpension:



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

```
14 11 4
14 31 5
14 52 4
15 52 4
16 11 5
16 31 4
16 52 4
17 52 4
18 11 6
18 31 3
18 52 4
19 52 4
20 11 7
20 32 2
20 52 4
21 4 4
21 6 5
21 8 6
21 10 7
21 11 7
21 12 7
21 14 6
21 16 5
21 18 4
21 20 3
21 22 2
21 24 1
21 52 4
22 11 7
22 31 1
22 52 4
23 31 1
23 52 4
24 11 6
24 52 4
25 31 2
25 52 4
26 11 5
26 52 4
27 31 3
27 52 4
28 11 4
28 52 4
29 32 4
29 52 4
30 11 3
30 52 4
31 32 5
31 36 3
31 52 4
32 11 2
32 22 1
32 24 2
32 26 3
32 27 3
32 30 5
32 31 5
32 32 5
32 34 4
32 36 3
32 38 2
32 40 1
```



```
2nd pass Expansion: Result of secondPassExpension:



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



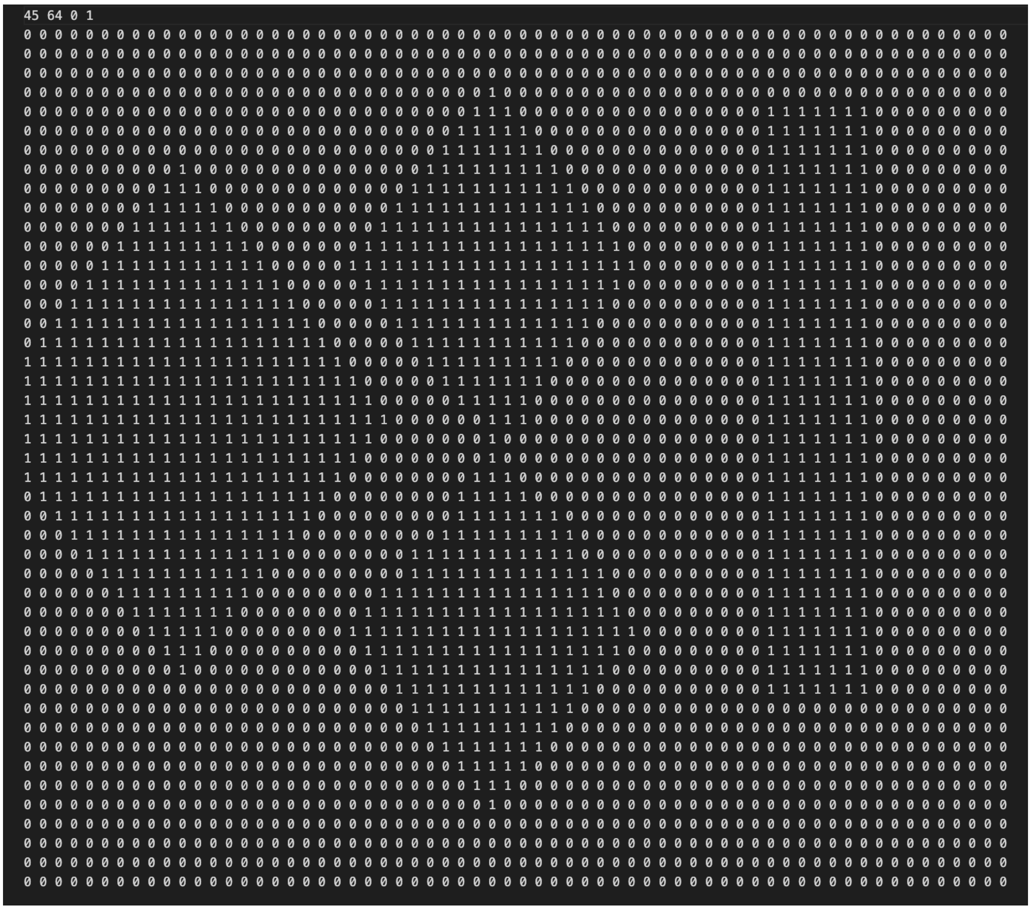
```
Decompressed File:
45 64 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 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 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 1 1 1 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 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 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 1 1 1 1 1 1 1 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 1 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 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 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 1 1 1 1 1 1 1 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 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 1 1 1 1 1 1 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 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 1 1 1 1 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 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 1 1 1 1 1 1 1 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 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 1 1 1 1 1 1 1 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 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 1 1 1 1 1 1 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 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 1 1 1 1 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 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 0 0 0 0 0 1 1 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 1 1 1 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 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 1 1 1 1 1 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 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 1 1 1 1 1 1 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 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 1 1 1 1 1 1 1 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 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 1 1 1 1 1 1 1 1 0 0 0 0 0 0 1 1 1 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
1 1 1 1 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 1 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
1 1 1 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 1 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
1 1 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 1 1 1 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 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 1 1 1 1 1 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 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 1 1 1 1 1 1 1 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 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 1 1 1 1 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 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 1 1 1 1 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 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 1 1 1 1 1 1 1 1 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 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 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 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 1 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 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 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
0 0 0 0 0 0 0 0 0 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 1 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 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 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 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 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 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
0 0 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 0 0 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 0 0 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 0 0 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 0 0 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
```

```
32 52 4
33 27 3
33 31 5
34 11 1
35 31 4
37 31 3
39 31 2
41 31 1
```

image2_ decompressed .txt

```
45 64 0 7
4 31 1
6 31 2
8 11 1
8 31 3
8 52 4
9 52 4
10 11 2
10 31 4
10 52 4
11 52 4
12 11 3
12 31 5
12 52 4
13 22 1
13 24 2
13 26 3
13 28 4
13 30 5
13 31 5
13 32 5
13 34 4
13 36 3
13 38 2
13 40 1
13 52 4
14 11 4
14 31 5
14 52 4
15 52 4
16 11 5
16 31 4
16 52 4
17 52 4
18 11 6
18 31 3
18 52 4
19 52 4
20 11 7
20 32 2
20 52 4
21 4 4
21 6 5
```

image2_decompressed.txt



```
45 64 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 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 1 1 1 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 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 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 1 1 1 1 1 1 1 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 1 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 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 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 1 1 1 1 1 1 1 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 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 1 1 1 1 1 1 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 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 1 1 1 1 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 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 1 1 1 1 1 1 1 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 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 1 1 1 1 1 1 1 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 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 1 1 1 1 1 1 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 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 1 1 1 1 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 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 0 0 0 0 0 1 1 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 1 1 1 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 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 1 1 1 1 1 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 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 1 1 1 1 1 1 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 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 1 1 1 1 1 1 1 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 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 1 1 1 1 1 1 1 1 0 0 0 0 0 0 1 1 1 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
1 1 1 1 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 1 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
1 1 1 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 1 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
1 1 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 1 1 1 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 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 1 1 1 1 1 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 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 1 1 1 1 1 1 1 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 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 1 1 1 1 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 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 1 1 1 1 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 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 1 1 1 1 1 1 1 1 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 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 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 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 1 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 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 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
0 0 0 0 0 0 0 0 0 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 1 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 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 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 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 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 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
0 0 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 0 0 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 0 0 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 0 0 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 0 0 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
```

```
21 8 6
21 10 7
21 11 7
21 12 7
21 14 6
21 16 5
21 18 4
21 20 3
21 22 2
21 24 1
21 52 4
22 11 7
22 31 1
22 52 4
23 31 1
23 52 4
24 11 6
24 52 4
25 31 2
25 52 4
26 11 5
26 52 4
27 31 3
27 52 4
28 11 4
28 52 4
29 32 4
29 52 4
30 11 3
30 52 4
31 32 5
31 36 3
31 52 4
32 11 2
32 22 1
32 24 2
32 26 3
32 27 3
32 30 5
32 31 5
32 32 5
32 34 4
32 36 3
32 38 2
32 40 1
32 52 4
33 27 3
33 31 5
34 11 1
35 31 4
37 31 3
39 31 2
41 31 1
```