# CV
# Project 6: Image Compression and Decompression via Distance Transform

## Adrian Noa
## Due 11/2/2022

### My Algorithms:

**firstPass8Distance():**
This method perform the first pass of the Distance Transform algorithm on the binary input image.

```
1) Loop: i → [1 to numRows]
   a) Loop: j → [1 to numCols]
      i) pixel ← ZFAry[i][j]
      ii) if pixel > 0:
          (1) ZFAry[i][j] ← checkNeighbors(ZFAry, “min”, 1, i, j) + 1
```

**int checkNeighbors(Ary, string “min” or “max”, set 1 or -1, i, j):**
This function returns and integer with the min or max of all the neighbors of the current pixel

```
1) Arr[] ← {Ary[i -1*set][j-1], Ary[i -1*set][j],Ary[i -1*set][j+1],Ary[i][j -1*set]}
2) If min : return ← *min_element(begin(arr), end(arr))
3) return ← *max_element(begin(arr), end(arr))
```

**secondPass8Distance():**
Similar to pass1, loops from bottom to top, right to left. Keeps track of new min and max values

```
1) newMinVal = 999
2) newMaxVal = -1
3) flag ← true
4) Loop: i → [numRows-1 → 1]
   a) Loop: j → [numCols-1 → 1]
      i) pixel ← ZFAry[i][j]
      ii) if pixel > 0:
          (1) ZFAry[i][j] ← min( checkNeighbors(ZFAry, “min”, -1, i, j) + 1, pixel )
          (2) If ZFAry[i][j] < newMinVal :
              (a) newMinVal = ZFAry[i][j]
          (3) If ZFAry[i][j] > newMinVal :
              (a) NewMaxVal = ZFAry[i][j]
```

**computeLocalMaxima():**
This Algorithm creates the Skeleton Image by adding pixels into the skelton array iff it’s a local maxima

```
1) Loop: i → [1 to numRows]
   a) Loop: j → [1 to numCols]
```

```
        i)  if isLocalMaxima(i,j):
            (1) SkeletonAry[i][j] ← ZFAry[i][j]
```

**Bool isLocalMaxima(i, j):**

A pixel is a local maxima if its greater than all its 8 neighbors

```
1) Pixel ← ZFAry[i][j]
2) If pixel >= checkNeighbors(ZFAry, "max", 1, i, j) && pixel >= checkNeighbors(ZFAry, "max", -1, i, j) :
        a.  Return ← true
3) Return ← false
```

**loadSkeleton(ifstream& inFile):**

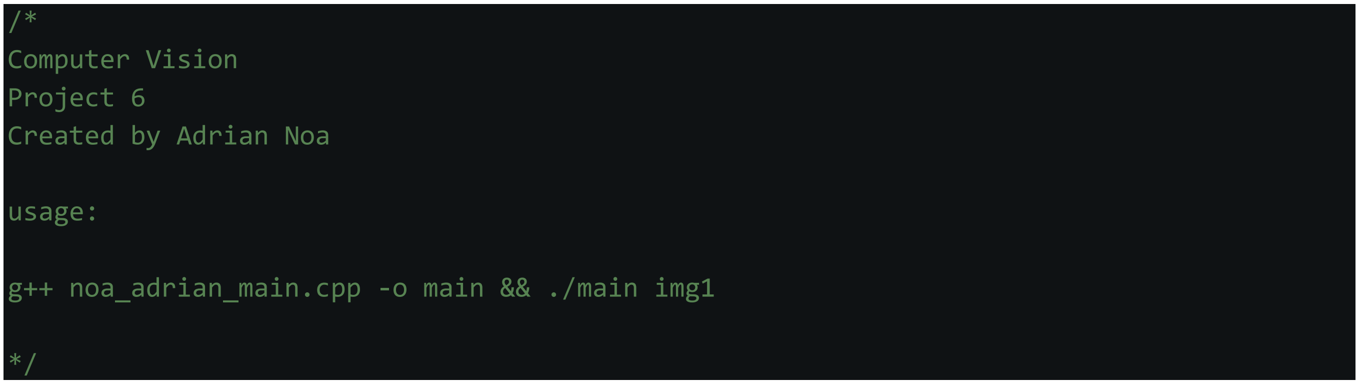
```
1) Zero2D(ZFAry)
2) i, j, val
3) inFile >> numRows >> numCols >> newMinVal >> newMaxVal
4) WHILE inFile >> i >> j >> val:
        a.  ZFAry[i][j] = val
```

**firstExpansion():**

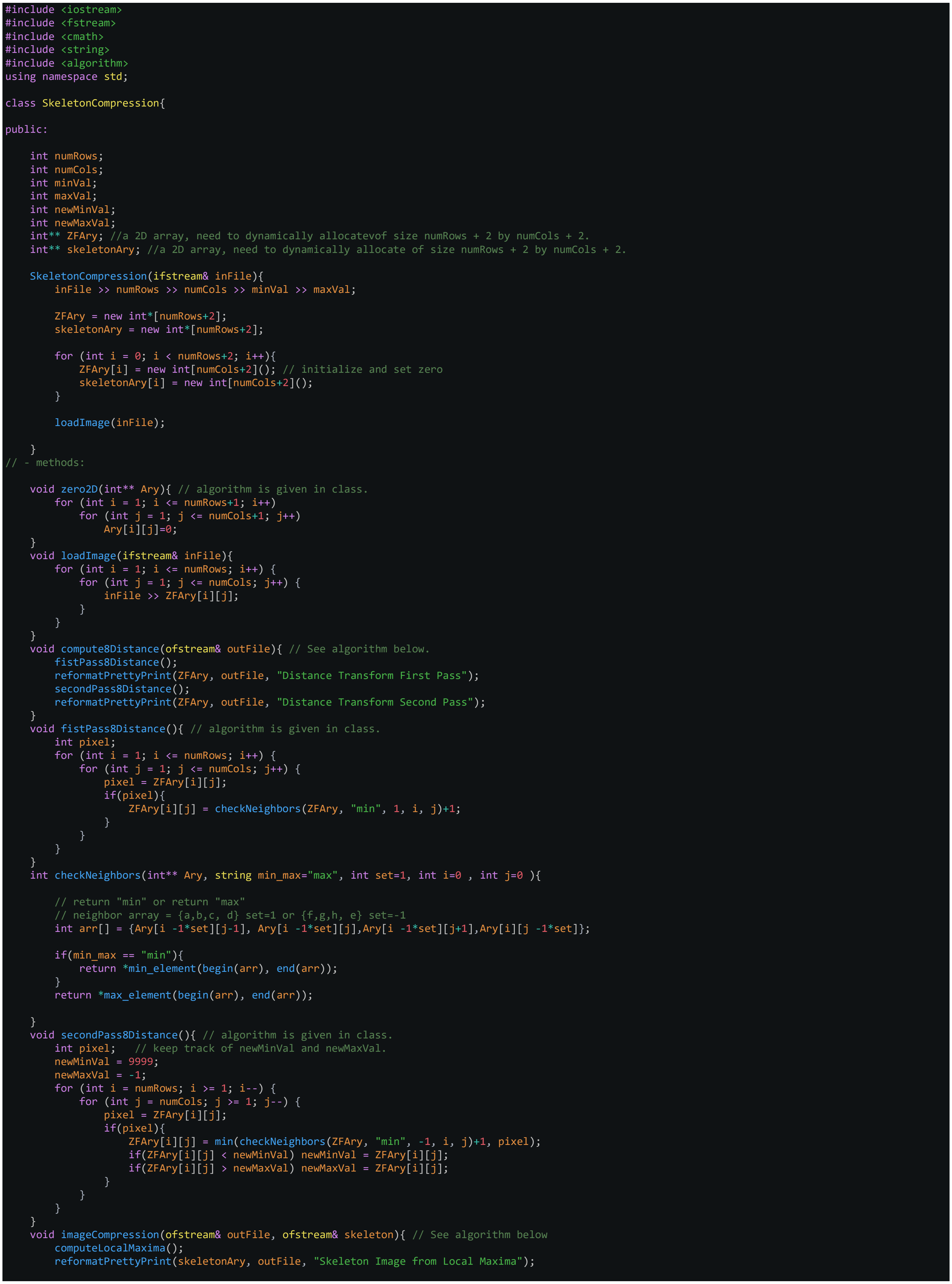
```
1) Loop: i → [1 to numRows]
    a) Loop: j → [1 to numCols]
        i)  pixel ← ZFAry[i][j]
        ii) if pixel == 0:
            (1) ZFAry[i][j] ← max(max(checkNeighbors(ZFAry, "max", 1, i, j) -1, checkNeighbors(ZFary, "max", -1, i, j)-1), pixel
```

**secondExpansion():**

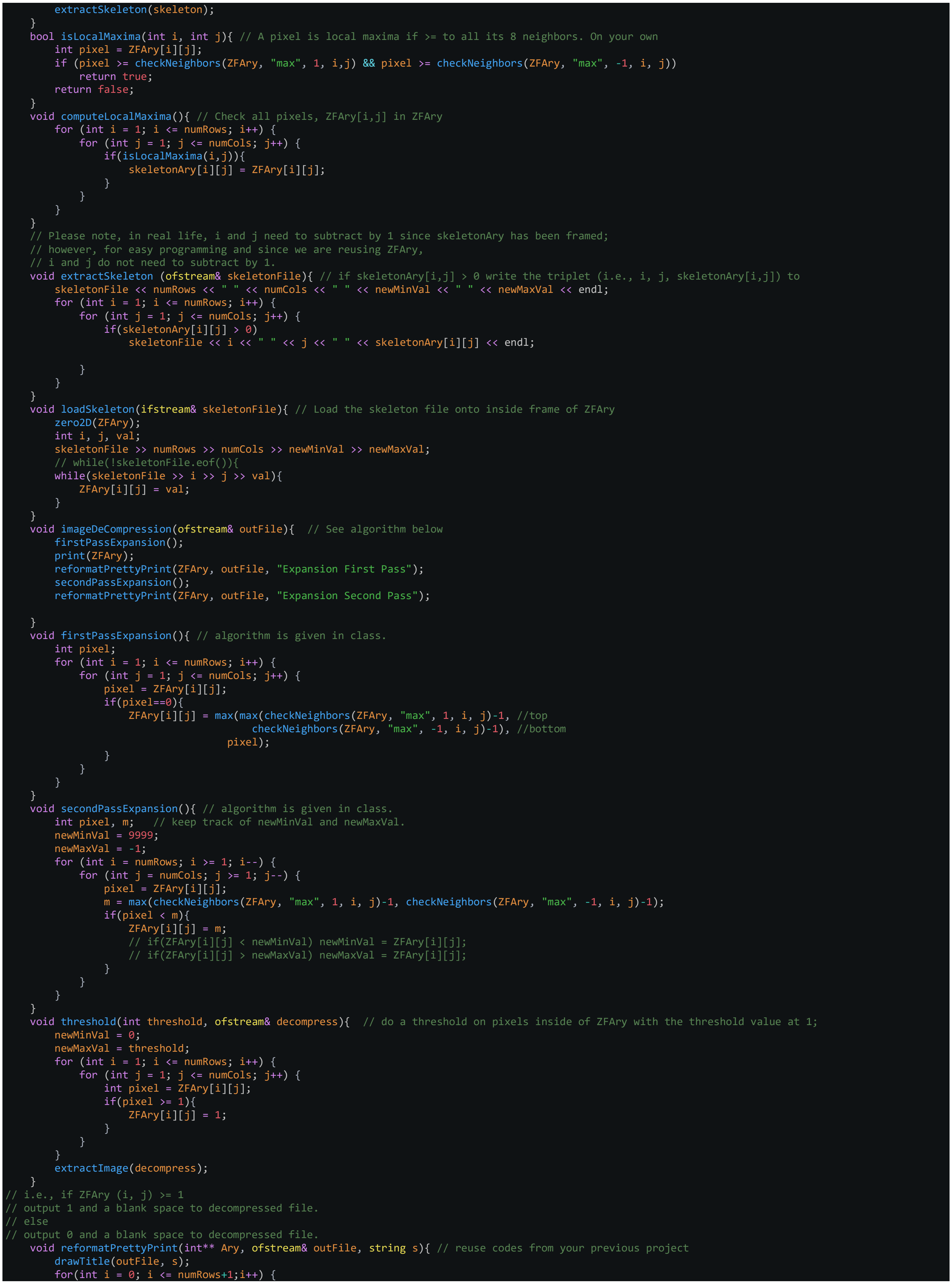
```
1) Loop: i → [numRows-1 → 1]
    a) Loop: j → [numCols-1 → 1]
        i)  pixel ← ZFAry[i][j]
            (1) m ← max(checkNeighbors(ZFAry, "max", 1, i, j) -1, checkNeighbors(ZFary, "max", -1, i, j)-1)
            (2) if pixel < m:
                (a) ZFAry[i][j] ← m
```



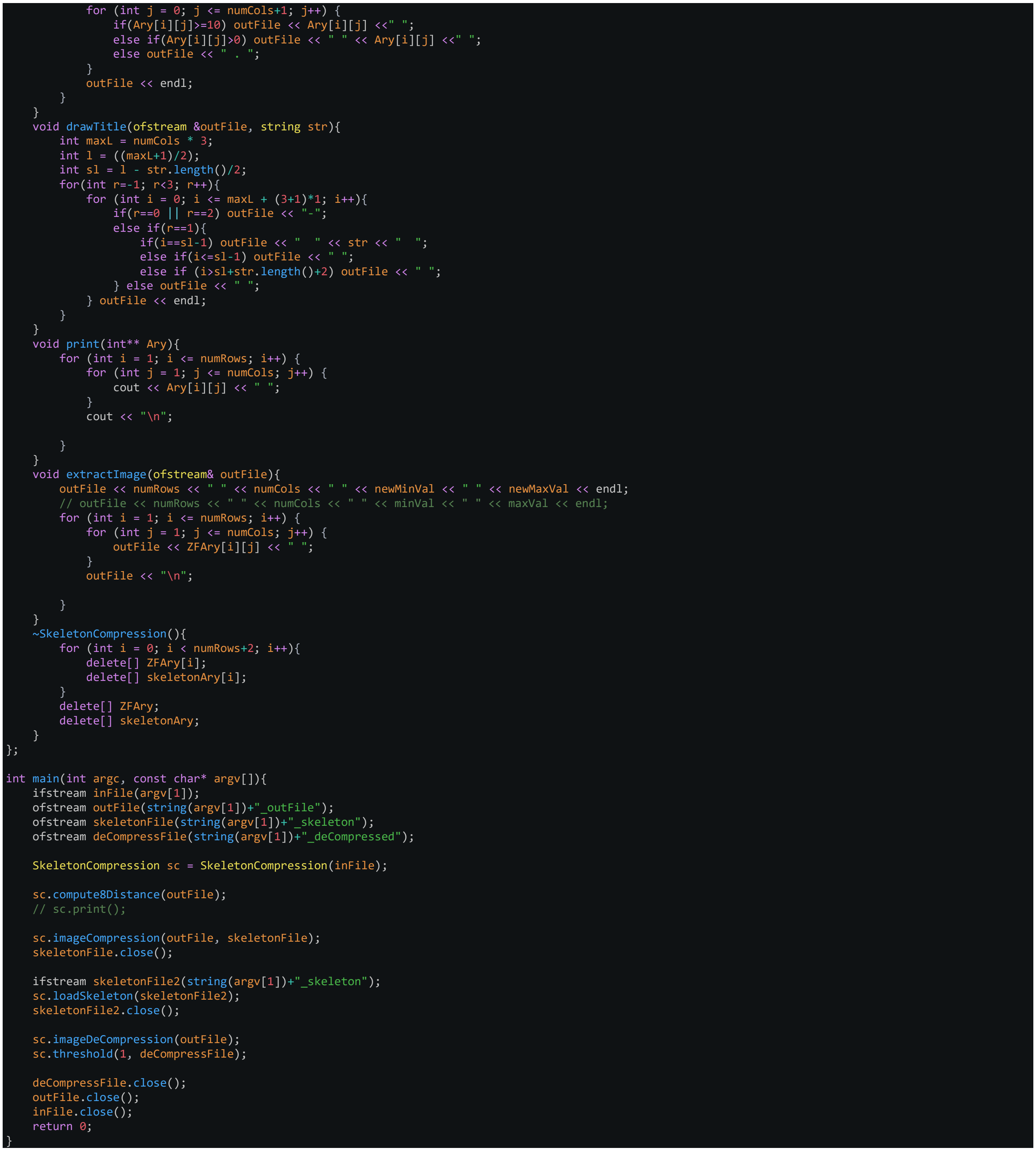
```
/*
Computer Vision
Project 6
Created by Adrian Noa

usage:

g++ noa_adrian_main.cpp -o main && ./main img1

*/
```



```
#include <iostream>
#include <fstream>
#include <cmath>
#include <string>
#include <algorithm>
using namespace std;

class SkeletonCompression{

public:

    int numRows;
    int numCols;
    int minVal;
    int maxVal;
    int newMinVal;
    int newMaxVal;
    int** ZFAry; //a 2D array, need to dynamically allocatevof size numRows + 2 by numCols + 2.
    int** skeletonAry; //a 2D array, need to dynamically allocate of size numRows + 2 by numCols + 2.

    SkeletonCompression(ifstream& inFile){
        inFile >> numRows >> numCols >> minVal >> maxVal;

        ZFAry = new int*[numRows+2];
        skeletonAry = new int*[numRows+2];

        for (int i = 0; i < numRows+2; i++){
            ZFAry[i] = new int[numCols+2](); // initialize and set zero
            skeletonAry[i] = new int[numCols+2]();
        }

        loadImage(inFile);

    }
// - methods:

    void zero2D(int** Ary){ // algorithm is given in class.
        for (int i = 1; i <= numRows+1; i++)
            for (int j = 1; j <= numCols+1; j++)
                Ary[i][j]=0;
    }
    void loadImage(ifstream& inFile){
        for (int i = 1; i <= numRows; i++) {
            for (int j = 1; j <= numCols; j++) {
                inFile >> ZFAry[i][j];
            }
        }
    }
    void compute8Distance(ofstream& outFile){ // See algorithm below.
        fistPass8Distance();
        reformatPrettyPrint(ZFAry, outFile, "Distance Transform First Pass");
        secondPass8Distance();
        reformatPrettyPrint(ZFAry, outFile, "Distance Transform Second Pass");
    }
    void fistPass8Distance(){ // algorithm is given in class.
        int pixel;
        for (int i = 1; i <= numRows; i++) {
            for (int j = 1; j <= numCols; j++) {
                pixel = ZFAry[i][j];
                if(pixel){
                    ZFAry[i][j] = checkNeighbors(ZFAry, "min", 1, i, j)+1;
                }
            }
        }
    }
    int checkNeighbors(int** Ary, string min_max="max", int set=1, int i=0 , int j=0 ){

        // return "min" or return "max"
        // neighbor array = {a,b,c, d} set=1 or {f,g,h, e} set=-1
        int arr[] = {Ary[i -1*set][j-1], Ary[i -1*set][j],Ary[i -1*set][j+1],Ary[i][j -1*set]};

        if(min_max == "min"){
            return *min_element(begin(arr), end(arr));
        }
        return *max_element(begin(arr), end(arr));

    }
    void secondPass8Distance(){ // algorithm is given in class.
        int pixel;   // keep track of newMinVal and newMaxVal.
        newMinVal = 9999;
        newMaxVal = -1;
        for (int i = numRows; i >= 1; i--) {
            for (int j = numCols; j >= 1; j--) {
                pixel = ZFAry[i][j];
                if(pixel){
                    ZFAry[i][j] = min(checkNeighbors(ZFAry, "min", -1, i, j)+1, pixel);
                    if(ZFAry[i][j] < newMinVal) newMinVal = ZFAry[i][j];
                    if(ZFAry[i][j] > newMaxVal) newMaxVal = ZFAry[i][j];
                }
            }
        }
    }
    void imageCompression(ofstream& outFile, ofstream& skeleton){ // See algorithm below
        computeLocalMaxima();
        reformatPrettyPrint(skeletonAry, outFile, "Skeleton Image from Local Maxima");
```

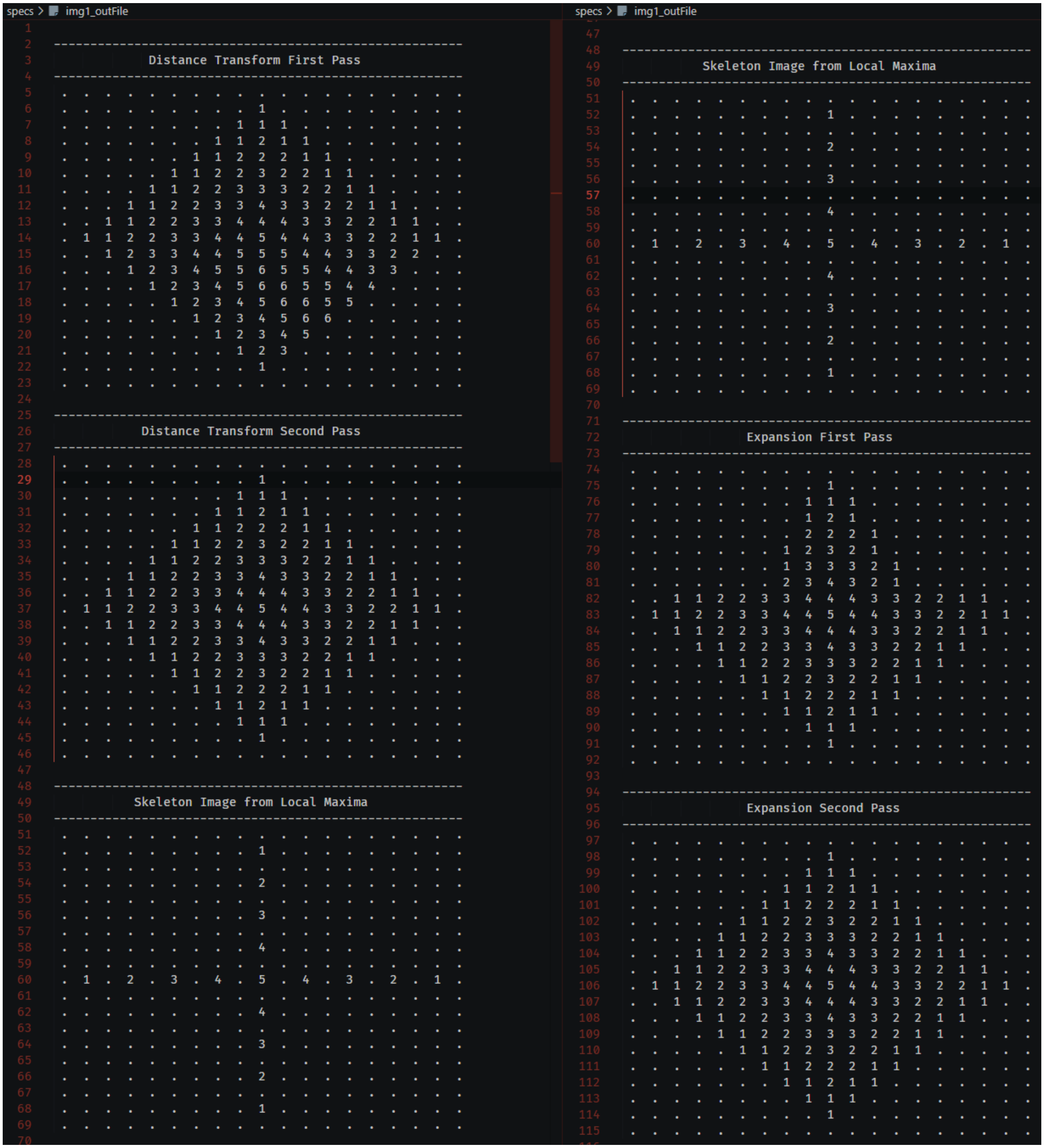


```
        extractSkeleton(skeleton);
    }
    bool isLocalMaxima(int i, int j){ // A pixel is local maxima if >= to all its 8 neighbors. On your own
        int pixel = ZFAry[i][j];
        if (pixel >= checkNeighbors(ZFAry, "max", 1, i,j) && pixel >= checkNeighbors(ZFAry, "max", -1, i, j))
            return true;
        return false;
    }
    void computeLocalMaxima(){ // Check all pixels, ZFAry[i,j] in ZFAry
        for (int i = 1; i <= numRows; i++) {
            for (int j = 1; j <= numCols; j++) {
                if(isLocalMaxima(i,j)){
                    skeletonAry[i][j] = ZFAry[i][j];
                }
            }
        }
    }
    // Please note, in real life, i and j need to subtract by 1 since skeletonAry has been framed;
    // however, for easy programming and since we are reusing ZFAry,
    // i and j do not need to subtract by 1.
    void extractSkeleton (ofstream& skeletonFile){ // if skeletonAry[i,j] > 0 write the triplet (i.e., i, j, skeletonAry[i,j]) to
        skeletonFile << numRows << " " << numCols << " " << newMinVal << " " << newMaxVal << endl;
        for (int i = 1; i <= numRows; i++) {
            for (int j = 1; j <= numCols; j++) {
                if(skeletonAry[i][j] > 0)
                    skeletonFile << i << " " << j << " " << skeletonAry[i][j] << endl;

            }
        }
    }
    void loadSkeleton(ifstream& skeletonFile){ // Load the skeleton file onto inside frame of ZFAry
        zero2D(ZFAry);
        int i, j, val;
        skeletonFile >> numRows >> numCols >> newMinVal >> newMaxVal;
        // while(!skeletonFile.eof()){
        while(skeletonFile >> i >> j >> val){
            ZFAry[i][j] = val;
        }
    }
    void imageDeCompression(ofstream& outFile){  // See algorithm below
        firstPassExpansion();
        print(ZFAry);
        reformatPrettyPrint(ZFAry, outFile, "Expansion First Pass");
        secondPassExpansion();
        reformatPrettyPrint(ZFAry, outFile, "Expansion Second Pass");

    }
    void firstPassExpansion(){ // algorithm is given in class.
        int pixel;
        for (int i = 1; i <= numRows; i++) {
            for (int j = 1; j <= numCols; j++) {
                pixel = ZFAry[i][j];
                if(pixel==0){
                    ZFAry[i][j] = max(max(checkNeighbors(ZFAry, "max", 1, i, j)-1, //top
                                        checkNeighbors(ZFAry, "max", -1, i, j)-1), //bottom
                                    pixel);
                }
            }
        }
    }
    void secondPassExpansion(){ // algorithm is given in class.
        int pixel, m;   // keep track of newMinVal and newMaxVal.
        newMinVal = 9999;
        newMaxVal = -1;
        for (int i = numRows; i >= 1; i--) {
            for (int j = numCols; j >= 1; j--) {
                pixel = ZFAry[i][j];
                m = max(checkNeighbors(ZFAry, "max", 1, i, j)-1, checkNeighbors(ZFAry, "max", -1, i, j)-1);
                if(pixel < m){
                    ZFAry[i][j] = m;
                    // if(ZFAry[i][j] < newMinVal) newMinVal = ZFAry[i][j];
                    // if(ZFAry[i][j] > newMaxVal) newMaxVal = ZFAry[i][j];
                }
            }
        }
    }
    void threshold(int threshold, ofstream& decompress){  // do a threshold on pixels inside of ZFAry with the threshold value at 1;
        newMinVal = 0;
        newMaxVal = threshold;
        for (int i = 1; i <= numRows; i++) {
            for (int j = 1; j <= numCols; j++) {
                int pixel = ZFAry[i][j];
                if(pixel >= 1){
                    ZFAry[i][j] = 1;
                }
            }
        }
        extractImage(decompress);
    }
// i.e., if ZFAry (i, j) >= 1
// output 1 and a blank space to decompressed file.
// else
// output 0 and a blank space to decompressed file.
    void reformatPrettyPrint(int** Ary, ofstream& outFile, string s){ // reuse codes from your previous project
        drawTitle(outFile, s);
        for(int i = 0; i <= numRows+1;i++) {
```



```
            for (int j = 0; j <= numCols+1; j++) {
                if(Ary[i][j]>=10) outFile << Ary[i][j] <<" ";
                else if(Ary[i][j]>0) outFile << " " << Ary[i][j] <<" ";
                else outFile << " . ";
            }
            outFile << endl;
        }
    }
    void drawTitle(ofstream &outFile, string str){
        int maxL = numCols * 3;
        int l = ((maxL+1)/2);
        int sl = l - str.length()/2;
        for(int r=-1; r<3; r++){
            for (int i = 0; i <= maxL + (3+1)*1; i++){
                if(r==0 || r==2) outFile << "-";
                else if(r==1){
                    if(i==sl-1) outFile << "  " << str << "  ";
                    else if(i<=sl-1) outFile << " ";
                    else if (i>sl+str.length()+2) outFile << " ";
                } else outFile << " ";
            } outFile << endl;
        }
    }
    void print(int** Ary){
        for (int i = 1; i <= numRows; i++) {
            for (int j = 1; j <= numCols; j++) {
                cout << Ary[i][j] << " ";
            }
            cout << "\n";

        }
    }
    void extractImage(ofstream& outFile){
        outFile << numRows << " " << numCols << " " << newMinVal << " " << newMaxVal << endl;
        // outFile << numRows << " " << numCols << " " << minVal << " " << maxVal << endl;
        for (int i = 1; i <= numRows; i++) {
            for (int j = 1; j <= numCols; j++) {
                outFile << ZFAry[i][j] << " ";
            }
            outFile << "\n";

        }
    }
    ~SkeletonCompression(){
        for (int i = 0; i < numRows+2; i++){
            delete[] ZFAry[i];
            delete[] skeletonAry[i];
        }
        delete[] ZFAry;
        delete[] skeletonAry;
    }
};

int main(int argc, const char* argv[]){
    ifstream inFile(argv[1]);
    ofstream outFile(string(argv[1])+"_outFile");
    ofstream skeletonFile(string(argv[1])+"_skeleton");
    ofstream deCompressFile(string(argv[1])+"_deCompressed");

    SkeletonCompression sc = SkeletonCompression(inFile);

    sc.compute8Distance(outFile);
    // sc.print();

    sc.imageCompression(outFile, skeletonFile);
    skeletonFile.close();

    ifstream skeletonFile2(string(argv[1])+"_skeleton");
    sc.loadSkeleton(skeletonFile2);
    skeletonFile2.close();

    sc.imageDeCompression(outFile);
    sc.threshold(1, deCompressFile);

    deCompressFile.close();
    outFile.close();
    inFile.close();
    return 0;
}
```

# IMG 1 Output

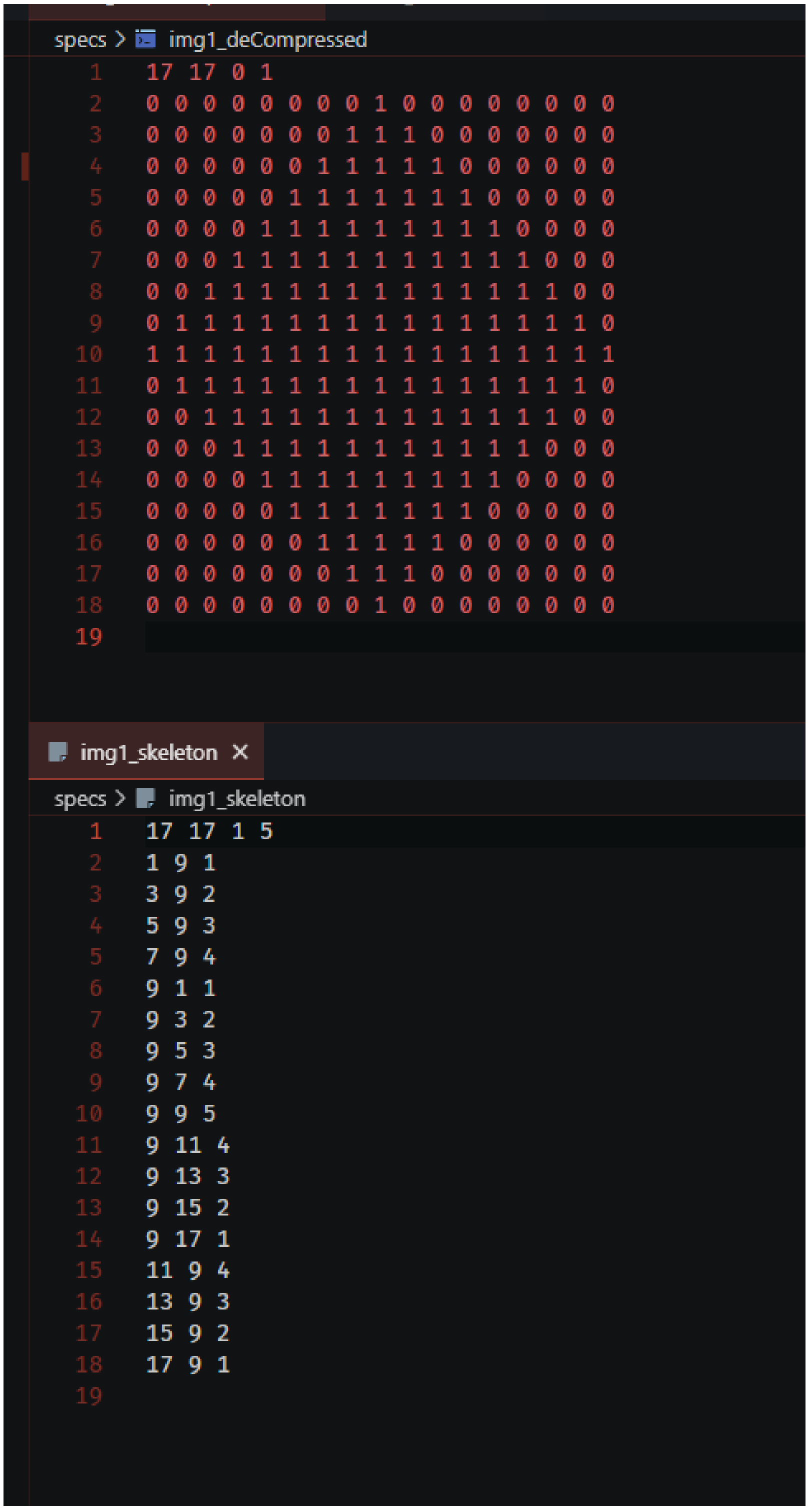


```

--------------------------------------------------------
             Distance Transform First Pass
--------------------------------------------------------
 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  1  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  1  1  1  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  1  1  2  1  1  .  .  .  .  .  .  .
 .  .  .  .  .  .  1  1  2  2  2  1  1  .  .  .  .  .  .
 .  .  .  .  .  1  1  2  2  3  2  2  1  1  .  .  .  .  .
 .  .  .  .  1  1  2  2  3  3  3  2  2  1  1  .  .  .  .
 .  .  .  1  1  2  2  3  3  4  3  3  2  2  1  1  .  .  .
 .  .  1  1  2  2  3  3  4  4  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  3  4  4  5  5  5  4  4  3  3  2  2  .  .
 .  .  .  1  2  3  4  5  5  6  5  5  4  4  3  3  .  .  .
 .  .  .  .  1  2  3  4  5  6  6  5  5  4  4  .  .  .  .
 .  .  .  .  .  1  2  3  4  5  6  6  5  5  .  .  .  .  .
 .  .  .  .  .  .  1  2  3  4  5  6  6  .  .  .  .  .  .
 .  .  .  .  .  .  .  1  2  3  4  5  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  1  2  3  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  1  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .


--------------------------------------------------------
            Distance Transform Second Pass
--------------------------------------------------------
 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  1  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  1  1  1  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  1  1  2  1  1  .  .  .  .  .  .  .
 .  .  .  .  .  .  1  1  2  2  2  1  1  .  .  .  .  .  .
 .  .  .  .  .  1  1  2  2  3  2  2  1  1  .  .  .  .  .
 .  .  .  .  1  1  2  2  3  3  3  2  2  1  1  .  .  .  .
 .  .  .  1  1  2  2  3  3  4  3  3  2  2  1  1  .  .  .
 .  .  1  1  2  2  3  3  4  4  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  1  2  2  3  3  4  4  4  3  3  2  2  1  1  .  .
 .  .  .  1  1  2  2  3  3  4  3  3  2  2  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  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .


--------------------------------------------------------
           Skeleton Image from Local Maxima
--------------------------------------------------------
 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  1  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  2  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  3  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  4  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
 .  1  .  2  .  3  .  4  .  5  .  4  .  3  .  2  .  1  .
 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  4  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  3  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  2  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  1  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

```

--------------------------------------------------------
           Skeleton Image from Local Maxima
--------------------------------------------------------
 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  1  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  2  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  3  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  4  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
 .  1  .  2  .  3  .  4  .  5  .  4  .  3  .  2  .  1  .
 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  4  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  3  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  2  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  1  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

```

--------------------------------------------------------
                 Expansion First Pass
--------------------------------------------------------
 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  1  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  1  1  1  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  1  2  1  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  2  2  2  1  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  1  2  3  2  1  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  1  3  3  3  2  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  5  4  4  3  3  2  2  1  1  .
 .  .  1  1  2  2  3  3  4  4  4  3  3  2  2  1  1  .  .
 .  .  .  1  1  2  2  3  3  4  3  3  2  2  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  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .


--------------------------------------------------------
                 Expansion Second Pass
--------------------------------------------------------
 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  1  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  1  1  1  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  1  1  2  1  1  .  .  .  .  .  .  .
 .  .  .  .  .  .  1  1  2  2  2  1  1  .  .  .  .  .  .
 .  .  .  .  .  1  1  2  2  3  2  2  1  1  .  .  .  .  .
 .  .  .  .  1  1  2  2  3  3  3  2  2  1  1  .  .  .  .
 .  .  .  1  1  2  2  3  3  4  3  3  2  2  1  1  .  .  .
 .  .  1  1  2  2  3  3  4  4  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  1  2  2  3  3  4  4  4  3  3  2  2  1  1  .  .
 .  .  .  1  1  2  2  3  3  4  3  3  2  2  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  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

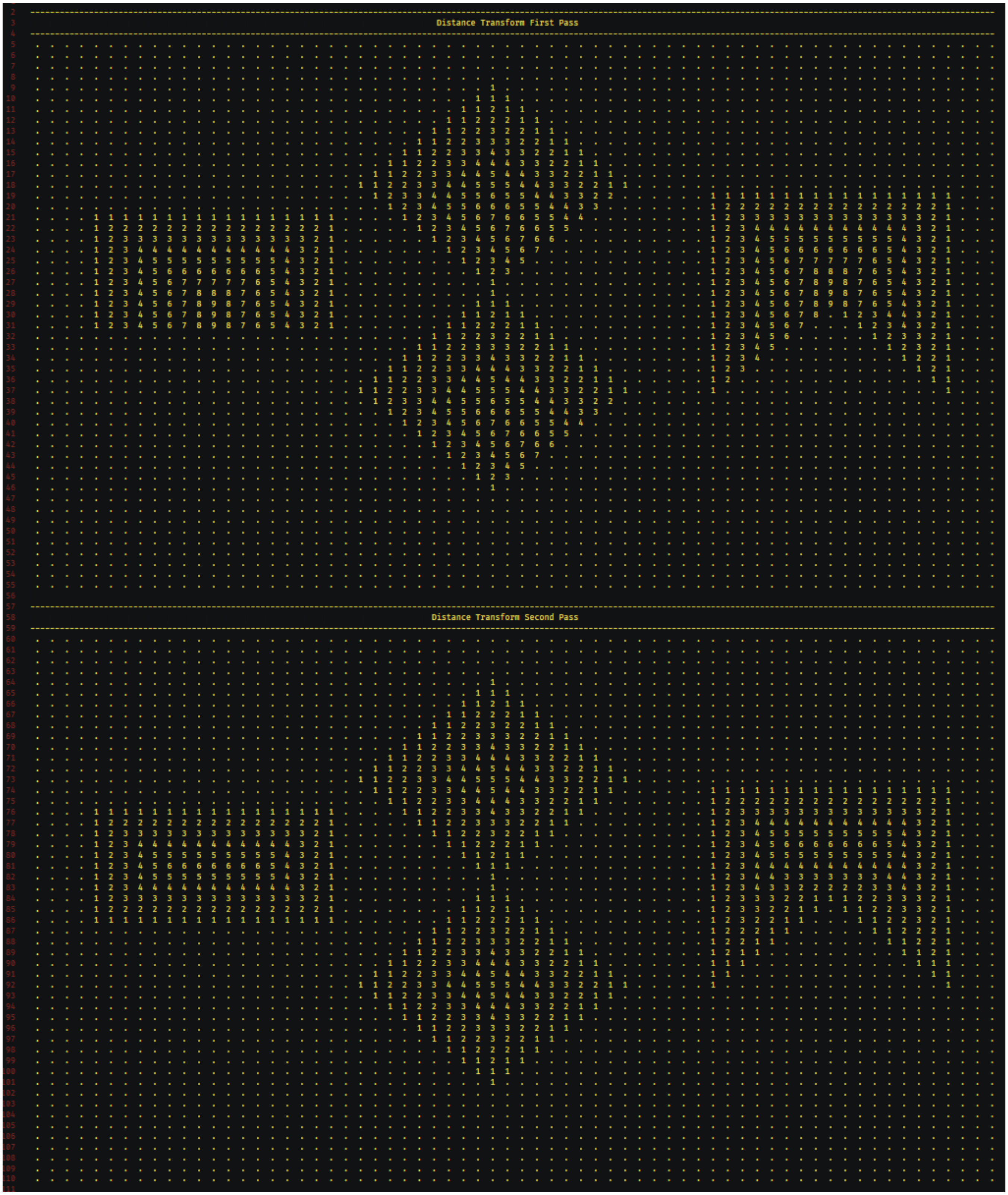
# IMG 1 Compression and Decompression files



```
17 17 0 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 1 1 1 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 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 0 0 0
0 0 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
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
0 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 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 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 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 1 0 0 0 0 0 0 0 0

```

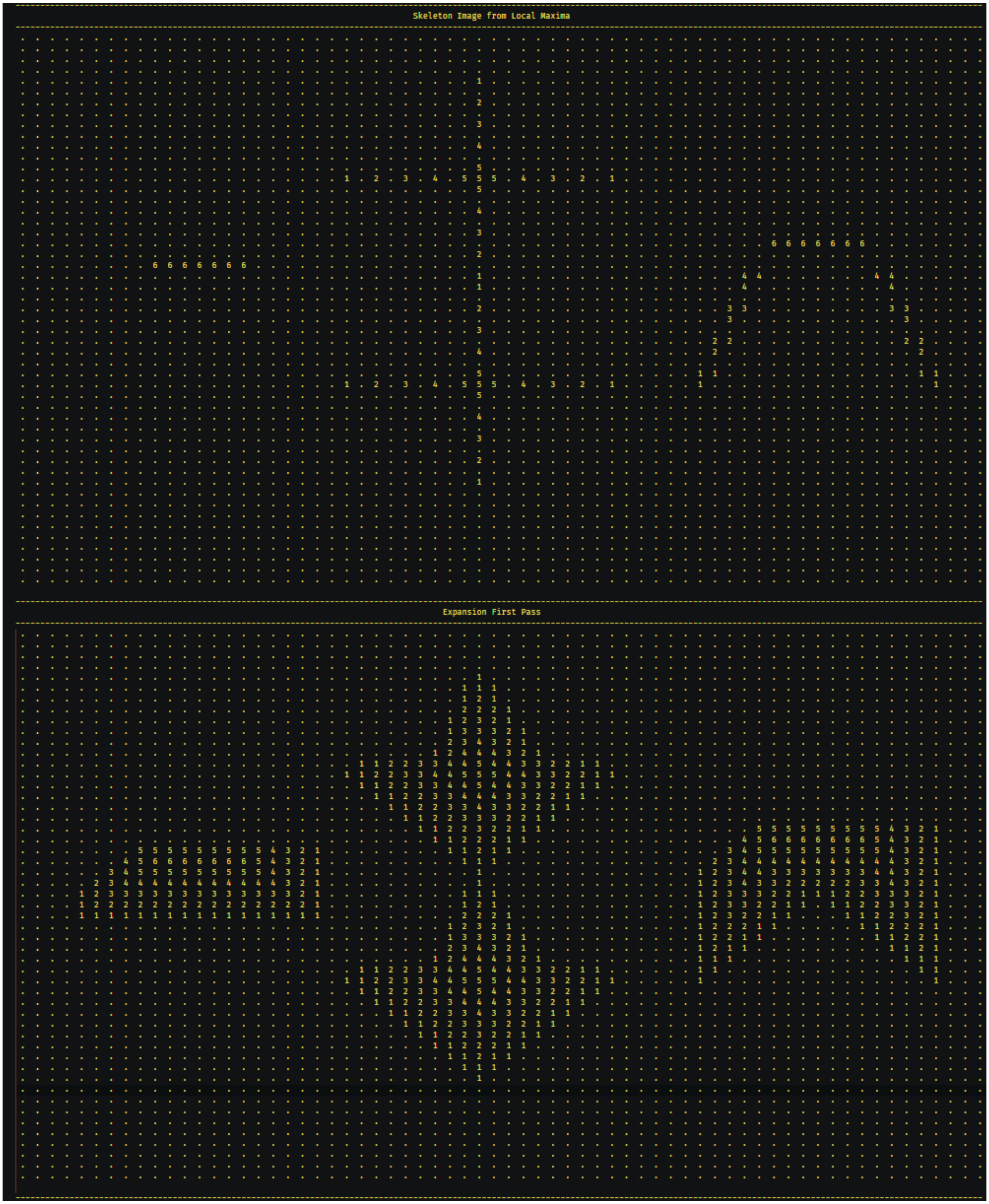
```
17 17 1 5
1 9 1
3 9 2
5 9 3
7 9 4
9 1 1
9 3 2
9 5 3
9 7 4
9 9 5
9 11 4
9 13 3
9 15 2
9 17 1
11 9 4
13 9 3
15 9 2
17 9 1

```

# IMG 2 Output



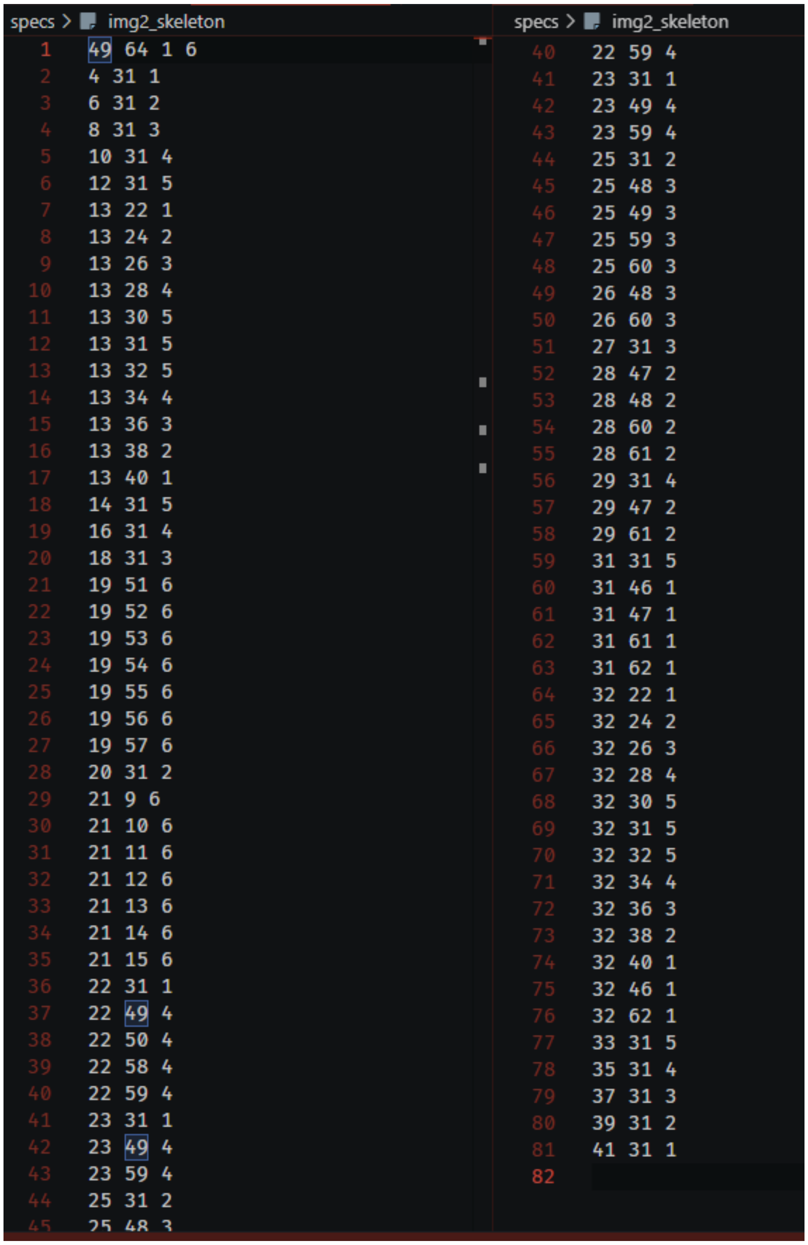
```
-----------------------------------------------------------------------------------------------------------------------------------
                                                   Distance Transform First Pass
-----------------------------------------------------------------------------------------------------------------------------------
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1 1 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1 1 2 1 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . 1 1 2 2 2 1 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . 1 1 2 2 3 2 2 1 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . 1 1 2 2 3 3 3 2 2 1 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . 1 1 2 2 3 3 4 3 3 2 2 1 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . 1 1 2 2 3 3 4 4 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 1 2 2 3 3 4 4 5 5 5 4 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 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 . . .
. . . . . . . . . . . . . . . . . . . . . . . . 1 2 3 4 5 5 6 6 6 5 5 4 4 3 3 . . . . . . . 1 2 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 2 3 4 5 6 7 6 6 5 5 4 4 . . . . . . . . 1 2 3 3 3 3 3 3 3 3 3 3 3 3 3 2 1 . . .
. . . . 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 . . . . . 1 2 3 4 5 6 7 6 6 5 5 . . . . . . . . . 1 2 3 4 4 4 4 4 4 4 4 4 4 4 3 2 1 . . .
. . . . 1 2 3 3 3 3 3 3 3 3 3 3 3 3 3 2 1 . . . . . . 1 2 3 4 5 6 7 6 6 . . . . . . . . . . 1 2 3 4 5 5 5 5 5 5 5 5 5 4 3 2 1 . . .
. . . . 1 2 3 4 4 4 4 4 4 4 4 4 4 4 3 2 1 . . . . . . . 1 2 3 4 5 6 7 . . . . . . . . . . . 1 2 3 4 5 6 6 6 6 6 6 6 5 4 3 2 1 . . .
. . . . 1 2 3 4 5 5 5 5 5 5 5 5 5 4 3 2 1 . . . . . . . . 1 2 3 4 5 . . . . . . . . . . . . 1 2 3 4 5 6 7 7 7 7 7 6 5 4 3 2 1 . . .
. . . . 1 2 3 4 5 6 6 6 6 6 6 6 5 4 3 2 1 . . . . . . . . . 1 2 3 . . . . . . . . . . . . . 1 2 3 4 5 6 7 8 8 8 7 6 5 4 3 2 1 . . .
. . . . 1 2 3 4 5 6 7 7 7 7 7 6 5 4 3 2 1 . . . . . . . . . . 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 8 8 7 6 5 4 3 2 1 . . . . . . . . . . 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 8 7 6 5 4 3 2 1 . . . . . . . . . 1 1 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 8 7 6 5 4 3 2 1 . . . . . . . . 1 1 2 1 1 . . . . . . . . . . . . 1 2 3 4 5 6 7 8 . 1 2 3 4 4 3 2 1 . . .
. . . . 1 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 1 . . . . . . . 1 1 2 2 2 1 1 . . . . . . . . . . . 1 2 3 4 5 6 7 . . . 1 2 3 4 3 2 1 . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . 1 1 2 2 3 2 2 1 1 . . . . . . . . . . 1 2 3 4 5 6 . . . . . 1 2 3 3 2 1 . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . 1 1 2 2 3 3 3 2 2 1 1 . . . . . . . . . 1 2 3 4 5 . . . . . . . 1 2 3 2 1 . . .
. . . . . . . . . . . . . . . . . . . . . . . . . 1 1 2 2 3 3 4 3 3 2 2 1 1 . . . . . . . . 1 2 3 4 . . . . . . . . . 1 2 2 1 . . .
. . . . . . . . . . . . . . . . . . . . . . . . 1 1 2 2 3 3 4 4 4 3 3 2 2 1 1 . . . . . . . 1 2 3 . . . . . . . . . . . 1 2 1 . . .
. . . . . . . . . . . . . . . . . . . . . . . 1 1 2 2 3 3 4 4 5 4 4 3 3 2 2 1 1 . . . . . . 1 2 . . . . . . . . . . . . . 1 1 . . .
. . . . . . . . . . . . . . . . . . . . . . 1 1 2 2 3 3 4 4 5 5 5 4 4 3 3 2 2 1 1 . . . . . 1 . . . . . . . . . . . . . . . 1 . . .
. . . . . . . . . . . . . . . . . . . . . . . 1 2 3 3 4 4 5 5 6 5 5 4 4 3 3 2 2 . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . 1 2 3 4 5 5 6 6 6 5 5 4 4 3 3 . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . 1 2 3 4 5 6 7 6 6 5 5 4 4 . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . 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 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

-----------------------------------------------------------------------------------------------------------------------------------
                                                   Distance Transform Second Pass
-----------------------------------------------------------------------------------------------------------------------------------
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1 1 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1 1 2 1 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . 1 1 2 2 2 1 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . 1 1 2 2 3 2 2 1 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . 1 1 2 2 3 3 3 2 2 1 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . 1 1 2 2 3 3 4 3 3 2 2 1 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . 1 1 2 2 3 3 4 4 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 1 2 2 3 3 4 4 5 5 5 4 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 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 . . .
. . . . . . . . . . . . . . . . . . . . . . . . 1 1 2 2 3 3 4 4 4 3 3 2 2 1 1 . . . . . . . 1 2 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 1 2 2 3 3 4 3 3 2 2 1 1 . . . . . . . . 1 2 3 3 3 3 3 3 3 3 3 3 3 3 3 2 1 . . .
. . . . 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 . . . . . 1 1 2 2 3 3 3 2 2 1 1 . . . . . . . . . 1 2 3 4 4 4 4 4 4 4 4 4 4 4 3 2 1 . . .
. . . . 1 2 3 3 3 3 3 3 3 3 3 3 3 3 3 2 1 . . . . . . 1 1 2 2 3 2 2 1 1 . . . . . . . . . . 1 2 3 4 5 5 5 5 5 5 5 5 5 4 3 2 1 . . .
. . . . 1 2 3 4 4 4 4 4 4 4 4 4 4 4 3 2 1 . . . . . . . 1 1 2 2 2 1 1 . . . . . . . . . . . 1 2 3 4 5 6 6 6 6 6 6 6 5 4 3 2 1 . . .
. . . . 1 2 3 4 5 5 5 5 5 5 5 5 5 4 3 2 1 . . . . . . . . 1 1 2 1 1 . . . . . . . . . . . . 1 2 3 4 5 5 5 5 5 5 5 5 5 4 3 2 1 . . .
. . . . 1 2 3 4 5 6 6 6 6 6 6 6 5 4 3 2 1 . . . . . . . . . 1 1 1 . . . . . . . . . . . . . 1 2 3 4 4 4 4 4 4 4 4 4 4 4 3 2 1 . . .
. . . . 1 2 3 4 5 5 5 5 5 5 5 5 5 4 3 2 1 . . . . . . . . . . 1 . . . . . . . . . . . . . . 1 2 3 4 4 3 3 3 3 3 3 3 4 4 3 2 1 . . .
. . . . 1 2 3 4 4 4 4 4 4 4 4 4 4 4 3 2 1 . . . . . . . . . . 1 . . . . . . . . . . . . . . 1 2 3 4 3 3 2 2 2 2 2 3 3 4 3 2 1 . . .
. . . . 1 2 3 3 3 3 3 3 3 3 3 3 3 3 3 2 1 . . . . . . . . . 1 1 1 . . . . . . . . . . . . . 1 2 3 3 3 2 2 1 1 1 2 2 3 3 3 2 1 . . .
. . . . 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 . . . . . . . . 1 1 2 1 1 . . . . . . . . . . . . 1 2 3 3 2 2 1 1 . 1 1 2 2 3 3 2 1 . . .
. . . . 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 . . . . . . . 1 1 2 2 2 1 1 . . . . . . . . . . . 1 2 3 2 2 1 1 . . . 1 1 2 2 3 2 1 . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . 1 1 2 2 3 2 2 1 1 . . . . . . . . . . 1 2 2 2 1 1 . . . . . 1 1 2 2 2 1 . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . 1 1 2 2 3 3 3 2 2 1 1 . . . . . . . . . 1 2 2 1 1 . . . . . . . 1 1 2 2 1 . . .
. . . . . . . . . . . . . . . . . . . . . . . . . 1 1 2 2 3 3 4 3 3 2 2 1 1 . . . . . . . . 1 2 1 1 . . . . . . . . . 1 1 2 1 . . .
. . . . . . . . . . . . . . . . . . . . . . . . 1 1 2 2 3 3 4 4 4 3 3 2 2 1 1 . . . . . . . 1 1 1 . . . . . . . . . . . 1 1 1 . . .
. . . . . . . . . . . . . . . . . . . . . . . 1 1 2 2 3 3 4 4 5 4 4 3 3 2 2 1 1 . . . . . . 1 1 . . . . . . . . . . . . . 1 1 . . .
. . . . . . . . . . . . . . . . . . . . . . 1 1 2 2 3 3 4 4 5 5 5 4 4 3 3 2 2 1 1 . . . . . 1 . . . . . . . . . . . . . . . 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 4 4 3 3 2 2 1 1 . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . 1 1 2 2 3 3 4 3 3 2 2 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 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
```

# IMG 2 Output



```
---------------------------------------------------------------------------------------------------------------------------------
                                                  Skeleton Image from Local Maxima
---------------------------------------------------------------------------------------------------------------------------------
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . 1 . 2 . 3 . 4 . 5 5 5 . 4 . 3 . 2 . 1 . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6 6 6 6 6 6 6 . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . 6 6 6 6 6 6 6 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1 . . . . . . . . . . . . . . . . . 4 4 . . . . . . . 4 4 . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1 . . . . . . . . . . . . . . . . . 4 . . . . . . . . . 4 . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2 . . . . . . . . . . . . . . . . 3 3 . . . . . . . . . 3 3 . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3 . . . . . . . . . . . 3 . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2 2 . . . . . . . . . . . 2 2 . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4 . . . . . . . . . . . . . . . 2 . . . . . . . . . . . . . 2 . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5 . . . . . . . . . . . . . . 1 1 . . . . . . . . . . . . . 1 1 . . .
. . . . . . . . . . . . . . . . . . . . . . 1 . 2 . 3 . 4 . 5 5 5 . 4 . 3 . 2 . 1 . . . . . 1 . . . . . . . . . . . . . . . 1 . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

---------------------------------------------------------------------------------------------------------------------------------
                                                        Expansion First Pass
---------------------------------------------------------------------------------------------------------------------------------
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1 1 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1 2 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2 2 2 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1 2 3 2 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1 3 3 3 2 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2 3 4 3 2 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . 1 2 4 4 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 4 5 5 5 4 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 1 2 2 3 3 4 4 4 3 3 2 2 1 1 . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . 1 1 2 2 3 3 4 3 3 2 2 1 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . 1 1 2 2 3 3 3 2 2 1 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . 1 1 2 2 3 2 2 1 1 . . . . . . . . . . . . . . 5 5 5 5 5 5 5 5 5 4 3 2 1 . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . 1 1 2 2 2 1 1 . . . . . . . . . . . . . . 4 5 6 6 6 6 6 6 6 5 4 3 2 1 . . .
. . . . . . . . 5 5 5 5 5 5 5 5 5 4 3 2 1 . . . . . . . . 1 1 2 1 1 . . . . . . . . . . . . . . 3 4 5 5 5 5 5 5 5 5 5 4 3 2 1 . . .
. . . . . . . 4 5 6 6 6 6 6 6 6 5 4 3 2 1 . . . . . . . . . 1 1 1 . . . . . . . . . . . . . . 2 3 4 4 4 4 4 4 4 4 4 4 4 3 2 1 . . .
. . . . . . 3 4 5 5 5 5 5 5 5 5 5 4 3 2 1 . . . . . . . . . . 1 . . . . . . . . . . . . . . 1 2 3 4 4 3 3 3 3 3 3 3 4 4 3 2 1 . . .
. . . . . 2 3 4 4 4 4 4 4 4 4 4 4 4 3 2 1 . . . . . . . . . . 1 . . . . . . . . . . . . . . 1 2 3 4 3 3 2 2 2 2 2 3 3 4 3 2 1 . . .
. . . . 1 2 3 3 3 3 3 3 3 3 3 3 3 3 3 2 1 . . . . . . . . . 1 1 1 . . . . . . . . . . . . . 1 2 3 3 3 2 2 1 1 1 2 2 3 3 3 2 1 . . .
. . . . 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 . . . . . . . . . 1 2 1 . . . . . . . . . . . . . 1 2 3 3 2 2 1 1 . 1 1 2 2 3 3 2 1 . . .
. . . . 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 . . . . . . . . . 2 2 2 1 . . . . . . . . . . . . 1 2 3 2 2 1 1 . . . 1 1 2 2 3 2 1 . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1 2 3 2 1 . . . . . . . . . . . . 1 2 2 2 1 1 . . . . . 1 1 2 2 2 1 . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1 3 3 3 2 1 . . . . . . . . . . . 1 2 2 1 1 . . . . . . . 1 1 2 2 1 . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2 3 4 3 2 1 . . . . . . . . . . . 1 2 1 1 . . . . . . . . . 1 1 2 1 . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . 1 2 4 4 4 3 2 1 . . . . . . . . . . 1 1 1 . . . . . . . . . . . 1 1 1 . . .
. . . . . . . . . . . . . . . . . . . . . . . 1 1 2 2 3 3 4 4 5 4 4 3 3 2 2 1 1 . . . . . . 1 1 . . . . . . . . . . . . . 1 1 . . .
. . . . . . . . . . . . . . . . . . . . . . 1 1 2 2 3 3 4 4 5 5 5 4 4 3 3 2 2 1 1 . . . . . 1 . . . . . . . . . . . . . . . 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 4 4 3 3 2 2 1 1 . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . 1 1 2 2 3 3 4 3 3 2 2 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 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
---------------------------------------------------------------------------------------------------------------------------------
```

# IMG 2 Compression file



specs > img2_skeleton

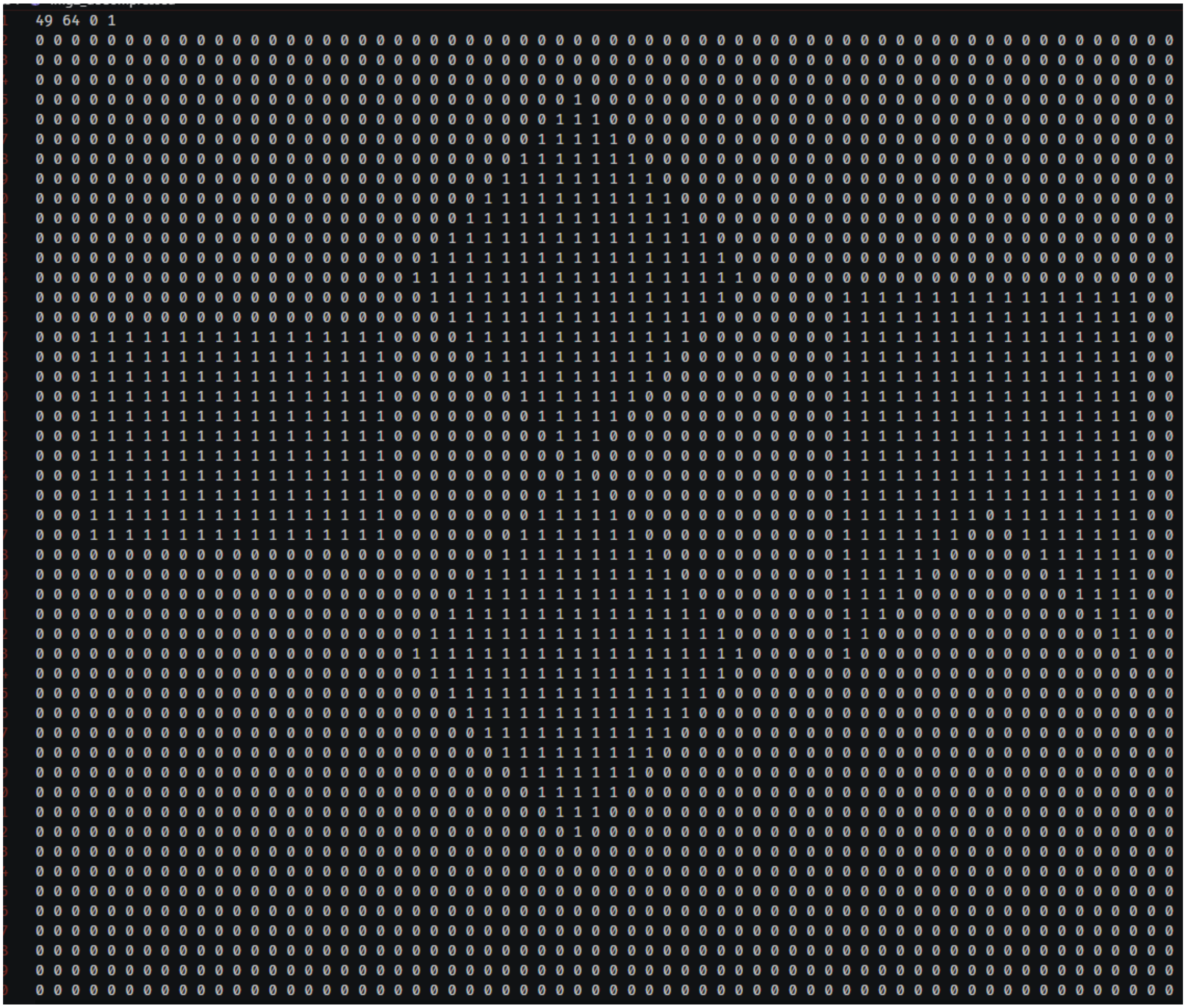
```
49 64 1 6
4 31 1
6 31 2
8 31 3
10 31 4
12 31 5
13 22 1
13 24 2
13 26 3
13 28 4
13 30 5
13 31 5
13 32 5
13 34 4
13 36 3
13 38 2
13 40 1
14 31 5
16 31 4
18 31 3
19 51 6
19 52 6
19 53 6
19 54 6
19 55 6
19 56 6
19 57 6
20 31 2
21 9 6
21 10 6
21 11 6
21 12 6
21 13 6
21 14 6
21 15 6
22 31 1
22 49 4
22 50 4
22 58 4
22 59 4
23 31 1
23 49 4
23 59 4
25 31 2
25 48 3
```

specs > img2_skeleton

```
22 59 4
23 31 1
23 49 4
23 59 4
25 31 2
25 48 3
```

```
25 49 3
25 59 3
25 60 3
26 48 3
26 60 3
27 31 3
28 47 2
28 48 2
28 60 2
28 61 2
29 31 4
29 47 2
29 61 2
31 31 5
31 46 1
31 47 1
31 61 1
31 62 1
32 22 1
32 24 2
32 26 3
32 28 4
32 30 5
32 31 5
32 32 5
32 34 4
32 36 3
32 38 2
32 40 1
32 46 1
32 62 1
33 31 5
35 31 4
37 31 3
39 31 2
41 31 1
```

# IMG 2 Decompression files



```
49 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 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 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 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 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 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 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
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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 0 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 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 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 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 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 0 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 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 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 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 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 0 0 0 0 0 0 0 0 0 1 1 1 1 1 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 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 0 0 0 0 0 0 0 0 0 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 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 1 1 1 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 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 0 0 0 0 0 0 0 0 0 0 0 0 0 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 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 1 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 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 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 0 0
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 0
0 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
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```