Name: Jingshi Liu

Section: Image Processing

Project: **Project 2 - Averaging and Gaussian Filter**

Due Date: Sept 2nd

Algorithm Steps

step 0:

inFile, maskFile, outFile1, deBugFile <— via argv[]

choice <— argv [3]

numRows, numCols, minVal, maxVal <— read from inFile

maskRows, maskCols, maskMin, maskMax <— read from maskFile

mirrorFramedAry, avgAry, GaussAry <— dynamically allocate with numRows + 4 by numCols + 4 cols

histAvgAry <— dynamically allocate with maxVal+1 and initialize to zero (Must do in C++)

histGaussAry <— dynamically allocate with maxVal+1 and initialize to zero (Must do in C++)

step 1: loadImage (inFile, mirrorFramedAry)

step 2: mirrorFraming (mirrorFramedAry)

step 3: imgReformat (mirrorFramedAry, outFile1)

step 4:

if choice == 1:

computeAvg5x5 (mirrorFramedAry, avgAry, deBugFile)

computeHist (avgAry, histAvgAry, deBugFile)

nameAvg <— argv[1] + "_Avg5x5.txt" // nameAvg is a string type

avgFile <— open nameAvg for write // avgFile is a file stream type

imgReformat (avgAry, outFile1) // print reformat to avgAry to outFile1

avgFile <— output numRows, numCols, minVal, maxVal

avgFile <— output avgAry to avgFile

avgHist <— argv[1] + "_Avg5x5_hist.txt" // avgHist is a string type

histAvgFile <— open avgHist for write // histAvgFile is a file stream type

printHist (histAvgAry, histAvgFile, maxVal, deBugFile)

Step 5:

if choice == 2:

maskWeight <— loadMaskAry (maskFile, maskAry)

computeGauss5x5 (mirrorFramedAry, GaussAry, maskAry, maskWeight, deBugFile)

computeHist (GaussAry, histGaussAry, deBugFile)

nameGauss <— argv[1] + "_Gauss5x5.txt"

GaussFile <— open nameGauss for write

imgReformat (GaussAry, outFile1)

GaussFile <— output numRows, numCols, minVal, maxVal

GaussFile <— output GaussAry to GaussFile

GaussHist <— argv[1] + "_Gauss5x5_hist.txt"

histGaussFile <— open GaussHist for write

printHist (histGaussAry, histGaussFile, maxVal, deBugFile)

Step 6: close all files

Source Code:

```
//
//  main.cpp
//  CS381_Image_Processing_Project2_Averaing_and_Gaussian_Filter
```

```
//
//  Created by Jingshi Liu on 9/16/2023.
//



#include <iostream>
#include <fstream>


using namespace std;


// ----------------------------------------Util Functions Declaration
-------------------------------------------------//
// Util functions are declared at the top the file to make sure all other functions can access
the Util Functions


int** getArray(int rows, int cols);


int* getArray(int length);


void printArray(int** array, int row, int col);



// ------------------------------------ Class Enhancement
-----------------------------------------------------------//



class Enhancement{
public:
    int numRows, numCols, minVal, maxVal, maskRows, maskCols, maskMin, maskMax, maskWeight;
    int** mirroredFramedArray;
    int** averagingArray;
    int** gaussArray;

    int* neighborArray;
    int* maskArray;
    int* histogramAveragingArray;
    int* histogramGaussianArray;

    Enhancement(ifstream& inFile, ifstream& maskFile){
        inFile >> numRows >> numCols >> minVal >> maxVal;
```

```
    maskFile >> maskRows >> maskCols >> maskMin >> maskMax;
    mirroredFramedArray = getArray(numRows + 4, numCols + 4);
    averagingArray = getArray(numRows + 4, numCols + 4);
    gaussArray = getArray(numRows + 4, numCols + 4);

    histogramAveragingArray = getArray(maxVal + 1);
    histogramGaussianArray = getArray(maxVal + 1);
    maskArray = getArray(25);
    neighborArray = getArray(25);

    loadImage(inFile);
    mirrorFraming();
}

void loadImage(ifstream& inFile){
    int num;
    for(int i = 2; i < numRows + 2; i++){
        for(int j = 2; j < numCols + 2; j++){
            inFile >> num;
            this->mirroredFramedArray[i][j] = num;
        }
    }
}

void mirrorFraming(){
    for(int i = 0; i < numRows + 4; i++){
        // mirror row 2
        mirroredFramedArray[i][0] = mirroredFramedArray[i][3];
        // mirror row 1
        mirroredFramedArray[i][1] = mirroredFramedArray[i][2];

        // mirror last row
        mirroredFramedArray[i][numRows+2] = mirroredFramedArray[i][numRows+1];
        // mirror last second row
        mirroredFramedArray[i][numRows+3] = mirroredFramedArray[i][numRows];
    }
    for(int i = 0; i < numCols + 4; i++){
        // mirror col 2
```

```
        mirroredFramedArray[0][i] = mirroredFramedArray[3][i];
        //mirror col 1
        mirroredFramedArray[1][i] = mirroredFramedArray[2][i];

        // mirror last col
        mirroredFramedArray[numCols+2][i] = mirroredFramedArray[numCols+1][i];
        // mirror last second col
        mirroredFramedArray[numCols+3][i] = mirroredFramedArray[numCols][i];
    }
}

int loadMaskArray(ifstream& maskFile){
    maskFile >> maskRows >> maskCols >> maskMin >> maskMax;
    int totalWeight = 0, index = 0, curWeight;
    for (int i = 0; i < maskRows; i++){
        for(int j = 0; j < maskCols; j++){
            maskFile >> curWeight;
            maskArray[index++] = curWeight;
            totalWeight += curWeight;
        }
    }
    this->maskWeight = totalWeight;
    return totalWeight;
}

void loadNeighborArray(int i, int j){
    int index = 0;
    for(int r = i - 2; r <= i + 2; r++){
        for(int c = j - 2; c <= j + 2; c++){
            neighborArray[index++] = mirroredFramedArray[r][c];
        }
    }
}

void computeAverage5x5(ofstream& debugFile){
    debugFile << "Entering computeAverage5x5 method\n";
    for(int i = 2; i < numRows + 2; i++){
        for(int j = 2; j < numCols + 2; j++){
```

```
            averagingArray[i][j] = average5x5(i, j);
        }
    }
    debugFile << "Leaving computeAverage5x5 method\n";
}

int average5x5(int i, int j){
    int sum = 0;
    for(int r = i - 2; r <= i + 2; r++){
        for(int c = j - 2; c <= j + 2; c++){
            sum += mirroredFramedArray[r][c];
        }
    }
    return sum / 25;
}

void computeGaussian5x5(ofstream& debugFile){
    debugFile << "Entering computeGaussian5x5 method\n";
    for(int i = 2; i < numRows + 2; i++){
        for(int j = 2; j < numCols + 2; j++){
            loadNeighborArray(i, j);
            gaussArray[i][j] = convolution(debugFile);
        }
    }
    debugFile << "Leaving computeGaussian5x5 method\n";
}

int convolution(ofstream& debugFile){
    debugFile << "Entering convolution method\n";
    int result = 0;
    for(int i = 0; i < 25; i++){
        result += (neighborArray[i] * maskArray[i]);
    }
    debugFile << "In convolution method, result is: " << result << '\n';
    debugFile << "Leaving convolution method\n";
    return result / maskWeight;
}
```

```
void computeHistogram(int** imageArray, int* histogramArray, ofstream& debugFile){
    debugFile << "Entering computeHistogram method\n";
    for(int i = 2; i < numRows + 2; i++){
        for(int j = 2; j < numCols + 2; j++){
            histogramArray[imageArray[i][j]]++;
        }
    }
    debugFile << "Leaving computeHistogram method\n";
}

void imageReformat(int** imageArray, ofstream& outFile){
    outFile << numRows << " " << numCols << " " << minVal << " " << maxVal << '\n';
    string str;
    int curWidth, pixelWidth = to_string(maxVal).length();

    for(int r = 2; r < numRows + 2; r++){
        for(int c = 2; c < numCols + 2; c++){
            outFile << imageArray[r][c];
            str = to_string(imageArray[r][c]);
            curWidth = str.length();
            while(curWidth < pixelWidth){
                outFile<<' ';
                curWidth++;
            }
        }
        outFile << '\n';
    }
}

void printHistogram(int* histogramArray, ofstream& outFile, ofstream& debugFile){
    debugFile << "Entering printHistogram method\n";
    outFile << numCols << " " << numCols << " " << minVal << " " << maxVal << '\n';
    for(int i = 0; i <= maxVal; i++){
        outFile << i << " " << histogramArray[i] << '\n';
    }
    debugFile << "Leaving printHistogram method\n";
}
```

```
    void outputImage(int** imageArray, ofstream& outFile){
        outFile << numRows << " " << numCols << " " << minVal << " " << maxVal <<'\n';
        for(int i = 2; i < numRows + 2; i++){
            for(int j = 2; j < numCols + 2; j++){
                outFile << imageArray[i][j] << " ";
            }
            outFile << "\n";
        }
    }
};

// ---------------------------------- Until Functions Implementation
----------------------------------------------------//

int** getArray(int rows, int cols){
    int** array = new int*[rows];
    for(int i = 0; i < rows; i++){
        array[i] = new int[cols];
        for(int j = 0; j < cols; j++){
            array[i][j] = 0;
        }
    }
    return array;
}

int* getArray(int length){
    int* array = new int[length];
    for(int i = 0; i < length; i++){
        array[i] = 0;
    }
    return array;
}

void useAverageFilter(const char* argv[], Enhancement* enhancement, ofstream& outFile, ofstream&
debugFile){
    enhancement->computeAverage5x5(debugFile);
    enhancement->computeHistogram(enhancement->averagingArray, enhancement-
>histogramAveragingArray ,debugFile);

    ofstream averageFile("./" + (string)argv[1] + "_Avg5x5.txt");
```

```
    enhancement->imageReformat(enhancement->averagingArray, outFile);
    enhancement->outputImage(enhancement->averagingArray, averageFile);

    ofstream histAvgFile("./" + (string)argv[1] + "_Avg5x5_hist.txt");
    enhancement->printHistogram(enhancement->histogramAveragingArray, histAvgFile, debugFile);

    averageFile.close();
    histAvgFile.close();
}

void useGaussianFilter(const char* argv[], Enhancement* enhancement, ifstream&
maskFile ,ofstream& outFile, ofstream& debugFile){
    enhancement->loadMaskArray(maskFile);
    enhancement->computeGaussian5x5(debugFile);
    enhancement->computeHistogram(enhancement->gaussArray, enhancement->histogramGaussianArray,
debugFile);

    ofstream gaussFile("./" + (string)argv[1] + "_Gauss5x5.txt");
    enhancement->imageReformat(enhancement->gaussArray, outFile);
    enhancement->outputImage(enhancement->gaussArray, gaussFile);

    ofstream histGaussFile("./" + (string)argv[1] + "_Gauss5x5_hist.txt");
    enhancement->printHistogram(enhancement->histogramGaussianArray, histGaussFile, debugFile);

    gaussFile.close();
    histGaussFile.close();
}

void printArray(int** array, int row, int col){
    for(int i = 0; i < row; i++){
        for(int j = 0; j < col; j++){
            cout<<array[i][j]<<" ";
        }
        cout<<endl;
    }
    cout<<endl<<endl;
}
```

```
// ------------------------------------ Main Function
-------------------------------------------------------//


int main(int argc, const char* argv[]){
    ifstream inFile, maskFile;
    ofstream outFile("./output.txt"), debugFile("./debugFile.txt");

    inFile.open(argv[1]);
    maskFile.open(argv[2]);
    string choice = argv[3];

    Enhancement* enhancement = new Enhancement(inFile, maskFile);
    enhancement->imageReformat(enhancement->mirroredFramedArray, outFile);
    if(choice == "1"){
        useAverageFilter(argv, enhancement, outFile, debugFile);
    }else if(choice == "2"){
        useGaussianFilter(argv, enhancement, maskFile, outFile, debugFile);
    }else{
        cout<< "Unknown choice argument entered, please enter either '1' or '2'\n";
    }

    inFile.close();
    maskFile.close();
    outFile.close();
    debugFile.close();
}
```

Program Output

Output 1

```
46 46 1 63
1 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5
2 1 2 3 4 551 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 434 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5
3 1 2 3 445 1 423 4 45512 3 4 5 1 2 3 4 5 1 2 584 5 1 2 534 5 1 2 3 4 45112 434 5 412 3 4 5
4 1 2 3 4 5 1 2 3 4 5 512 3 4 5 1 2 3 4 5 1 2 584 5 1 2 634 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5
5 1 623 4 5 1 2 434 5 1 2 3 4 5 1 2 3 4 5 1 2 5 4 5 1 2 534 351 2 3 4 5 412 3 4 5 1 2 3 4 5
6 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 414 5 1 2 3 4 5 1 2 3 4 5 512 3 4 55512 3 4 5
7 1 2 3 4 5 1 2 3 445 1 2 3 4 5 8 2 3 4 5 1 2 384 5 1 123 445 1 2 3 4 5 1 2 614 5 1 2 3 4 5
8 1 2 3 4 5 1 2 534 5 1 2 3 4 5 1 2 3 4 5 1 4253445 1 2 3 4 4 1 2 3 4 551 2 3 4 5 1 2 3 4 5
9 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 4838485 1 2 3 445 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5
10112 434 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 484 4848481 2 434 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5
1 1 2 3 4 5 112 3 4 5 1 2 3 4 5 1 2 3 4 4833414 4148482 3 445 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5
2 1 2 3 4 4 1 2 3 4 5 1 2 3 4 5 1 2 3 4848484 8 484848483 4 5 192 3 4 5 1 2 3 145 1 2 3 4 5
3 1 2 3 4 5 5 2 3 4 5 1 2 3 4 5 1 2 4 38448 8 34414 38373841351 2 3 4 5 1 2 3 445 1 2 3 4 5
4 413233343738393130323435343538404860636048413835343231302825282422201 88 6 134 5 1 2 3 145
5 1 213 4 5 1 2 3 4 5 1 2 3 4 5 1 484848481048484834484848485 1 2 3 4 5 1 2 3 4 5 1 323 4 5
6 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 484848484848484848484848484 48481 2 3 4 5 1 2 3 4 551 2 3 4 5
7 1 2 3 145 1 2 3 4 5 1 2 3 4 4848484142434142434 484846484848482 3 4 511 2 3 4 5 1 2 3 4 5
8 1 2 3 4 5 1 2 3 4 5 1 2 3 48414844488 45484 484848484848484 4 483 4 5 1 2 3 4 5 1 2 3 4 5
9 1 2 3 4 151 123 4 5 1 2 484848486048484848486162484848488 7 4848484 5 1 2 3 4 5 1 2 134 5
101 2 3 4 5 1 2 3 4 5 1 4848485 4848483 48484848486 484847488 484848485 1 123 4 5 1 2 3 4 5
1 1 523 4 5 1 123 4 5 484858484848484848283848484848488 484828283828181 2 3 4 5 112 3 4 5
2 1 2 3 4 5 1 2 3 4 484858484848404847484848414842485248 4 385 484848383828183 4 5 1 2 3 145
3 612223242738293130323435343538404860636048413845343931302825282422201818161 34 5 1 2 3 4 5
4 1 2 3 4 5 1 2 4848484848484 484848485858583838584858582824444848483838432818 4 5 1 2 3 4 5
5 1 2 4841484248438 486048484848414248434846484548404 84 3 48304848488 484838384 2 8 8 8 4 5
6 1 2 3 4 5 1 2 4848484848488 48486348634 4848484 4848488 4 4848484848184848484 5 1 2 3 4 5
7 1 2 3 4 5 1 2 3 48488 484842484818484848486348484848488 8 4848485 4848483 4 5 1 2 3 4 5
8 1 2 3 4 5 132 3 4 48486248554848484 378 484848544858484 84 28 48484848482 3 4 5 112 3 4 5
9 1 2 3 4 5 1 2 3 4 5 4848484848488 283848484 4848481 48486 4 384 4848481 2 3 4 5 1 2 3 4 5
101 2 3 4 5 1 2 3 4 5 1 484848483 484848484848184848484848488 4 48485 1 2 3 4 5 1 123 4 5
1 214253242728293130323435343538404860636048413835343231302825282432201 88 6 3 4 5 1 2 3 145
2 1 2 3 4 5 1 2 3 4 5 1 2 484848484818484848484848484488 48488 4 484 5 1 2 3 14151 2 3 4 5
3 112 3 4 5 1 2 3 4 5 1 2 3 484848484848484848484848484848488 4 3 4 5 1 2 3 4 5 1 2 3 4 5
```

4 1 2 3 4 151 2 3 4 5 1 2 3 4 484841424348404842484348444828484 82 3 4 5 1 2 3 4 551 2 3 4 5

5 1 2 3 42551 423 4 5 1 2 3 4 5 3444413424343441343442343424 4 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5

6 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 4848584 1 28411 482 4 8 485 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5

7 1 2 3 4 5 1 2 3 4 5 132 3 4 5 1 2 48488 4834354148488 484 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5

8 1 2 3 4 511 2 3 4 5 1 2 3 4 5 1 2 3 483848388 1 4838483 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5

9 1 2 3 4 5 1 123 4 5 1 2 3 4 5 1 2 3 4 484848484848482 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5

101 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 48481848481 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5

1 1 2 3 445 1 2 3 4 5 1 123 4 5 1 2 3 4 5 1 4848485 1 2 3 4 5 1 2 3 4 551 2 3 4 551 2 3 4 5

2 1 2 3 485 1 2 3 4 5551123 4 5 1 2 3 4 5 1 42484 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5

3 1 2 3 4 45512 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 484 5 1 2 3 4 5 1 2 3 635 1 2 3 4 5 1 2 3 4 5

4 1 2 3 4 5 1 2 3 4 5 1 2 3 145 1 2 3 4 5 1 2 484 5 1 2 3 4 5 1 2 595 5 1 2 434 5 1 2 334 5

5 1 2 3 4 5 112 3 445 1 2 3 4 5 1 2 3 4 5 1 2 484 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5

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

46 46 1 63

1 1 3 8 8 10108 6 8 6 6 6 5 3 3 3 3 3 3 3 5 5 5 5 5 8 8 8 8 8 3 3 4 5 5 6 6 5 6 6 4 4 4 3 4

1 2 4 6 6 8 8 6 6 108 8 8 7 3 3 3 3 3 3 3 7 7 7 7 7 9 9 9 9 9 3 3 4 5 5 6 6 5 6 6 4 4 4 3 4

4 4 6 9 9 8 9 8 7 11108 8 7 3 3 3 3 3 3 3 7 7 7 7 7 11111212124 4 4 6 6 8 8 6 6 6 4 4 4 3 4

4 5 6 9 9 8 9 8 7 11108 8 7 3 3 3 3 3 3 3 9 9 9 9 9 11111212124 4 4 8 8 10101010108 8 6 3 4

5 5 7 7 7 6 7 7 9 1311108 7 3 3 3 3 3 3 3 10101010109 111212125 4 4 8 8 12121212128 8 6 3 4

5 6 5 5 5 3 6 8 8 10106 5 5 3 3 3 3 3 3 4 11131313129 9 1010105 4 5 8 8 1010109 9 7 7 5 3 4

6 6 5 5 5 3 6 8 8 8 8 4 3 3 3 3 3 3 3 6 12161616138 8 9 9 9 7 4 5 8 8 1010109 9 7 7 5 3 4

4 6 5 5 4 4 5 6 6 6 6 4 3 3 3 3 3 3 3 4 8 1621232118119 8 8 7 6 2 5 7 7 9 9 9 9 9 7 7 5 3 4

4 6 5 5 5 5 5 7 7 6 6 4 3 3 3 3 3 3 5 7 1219262726211615119 9 7 2 5 5 5 7 7 5 5 5 3 3 3 3 4

4 5 5 4 4 4 5 5 5 5 5 3 3 3 3 3 3 4 8 131823303230272318131 08 6 3 5 5 5 5 5 3 3 3 3 3 3 3 4

3 5 5 4 5 5 3 3 3 3 3 3 3 3 3 3 3 3 6 1116202430303029272420151 29 4 3 3 3 3 5 5 5 5 5 3 3 3 4

7 8 10111111109 9 9 8 9 9 9 9 10111521272831333231323128241915129 7 6 5 5 6 5 5 5 5 3 3 4 5

7 7 8 101010109 9 9 8 9 9 9 9 12142028323434353234353533292317149 7 6 5 5 6 5 5 6 6 4 4 5 5

7 7 8 10109 9 9 9 9 8 9 9 9 11152028353838393835373937353326191 6107 6 5 5 6 7 7 8 8 6 4 5 5

7 7 9 101010109 9 9 8 9 9 11152127344041414138353839373838332620131 07 7 6 7 7 7 8 7 6 4 5 5

8 8 9 1010109 9 9 9 8 9 10142027354044454342413941424042383327211 4129 7 6 6 5 5 6 6 6 4 5 5

4 4 4 4 4 4 4 3 3 3 3 4 8 1320303740444441424342444644443933282315141 06 4 4 5 5 6 6 6 4 5 4

4 4 4 3 3 4 4 3 3 3 4 8 1318283540404443414244434345434239353228232115107 5 5 5 5 5 5 3 4 4

5 6 5 5 5 4 4 4 3 5 8 1421283541444243424040434243454343393734322726191 38 5 3 3 3 3 3 3 4 4

5 5 5 5 5 4 4 4 5 9 1422303541444542434341414544454743413633323331302517116 5 4 3 3 4 4 4 5

109 9 11109 9 9 1114202835384245454446474545474545454037343231343332262113 9 6 5 3 3 4 4 4 5

108 9 109 8 1012162129353840424243444747464546444444403836343336363531261913107 4 3 3 3 4 5

9 1012151616181923293539434444444447484948464645454438363332323735353329241914 9 6 4 4 4 4 5

8 8 1013141620222834404242424242434848494846434343423634323132373838373430251912 7 3 4 4 4 5

8 9 1113141620242934404240404141424847484947444444423835323132363637363531272114 7 3 3 3 4 4

3 5 6 8 10111519263240444444444341454243444543464745413634323337394142393429221 37 4 4 4 4 4

3 5 6 8 10111315202735404445464341423839404241444342393430292933363939373126181 17 4 4 4 4 4

4 4 3 3 3 3 5 8 131929364043454239413739404240444243433834322830353636342619148 5 3 3 3 4 4

6 8 8 9 9 9 8 10141726333842444140403941434343444242433834312828313231292014 9 5 3 3 4 4 4 5

6 8 8 9 9 9 8 8 10142027354042434040394141434243414242383531262729272623159 6 4 4 4 4 4 4 5

6 8 8 9 9 8 7 8 8 10141927343841414243454545444340424139373326262420171 69 6 4 3 4 4 4 3 4 5

6 7 8 9 9 9 8 8 8 8 10141927333841444547474645444343424141362927221512116 4 6 5 6 6 6 3 4 5

6 7 9 131314131210108 10141927343943444444444342424240383732252217118 7 4 4 6 5 5 5 5 3 4 5

3 3 4 7 6 8 8 7 4 4 3 4 8 132029343940403940373940393637342923191 27 4 4 3 3 5 5 5 5 5 3 3 4

3 3 4 7 6 8 8 7 4 5 3 3 5 8 132028353637383734383937363530252013 8 5 3 3 3 3 5 5 5 5 5 3 3 4

3 3 4 8 8 10108 4 5 3 3 3 3 5 7 131928303335343034343332312520151 06 4 3 3 3 3 5 5 5 5 5 3 3 4

3 4 4 8 8 10108 5 5 3 3 3 3 4 7 121923293233313535312827191 28 5 2 3 3 3 3 3 3 3 3 3 3 3 4

4 4 3 4 4 5 5 5 3 3 3 3 3 3 3 4 8 141724303231363429252214 8 6 4 3 3 3 3 3 3 3 3 3 3 3 3 4

3 4 5 6 6 6 6 5 3 3 3 3 3 3 3 3 4 8 111927313437352925191 26 4 3 3 5 5 5 5 5 5 5 5 5 3 3 4

3 3 6 8 8 8 8 5 5 7 7 7 7 5 3 3 3 4 7 132228313331241813 8 4 3 3 3 5 5 5 5 5 5 5 5 5 3 3 4

2 3 6 7 9 10107 7 7 7 7 7 5 3 3 3 3 4 8 152328302821138 4 3 3 3 5 7 7 7 7 5 5 5 5 5 5 3 3 4

2 2 6 7 9 9 9 6 7 7 7 7 8 6 4 3 3 3 3 4 1017212321168 4 3 3 3 5 7 9 9 9 9 6 6 6 6 6 6 4 5 5

2 2 5 7 1010108 9 8 9 9 8 6 4 3 3 3 3 3 3 6 15171717134 3 3 3 3 5 7 9 9 9 9 6 6 6 6 6 6 4 5 5

2 2 4 6 8 8 8 8 9 8 9 9 7 5 3 3 3 3 3 3 3 4 11111111103 3 3 3 3 5 7 7 7 7 7 4 4 4 4 4 4 4 5 5

2 3 2 4 7 7 7 8 7 4 4 4 3 3 3 3 3 3 3 3 3 3 8 8 8 8 8 3 3 3 3 3 5 7 7 7 7 7 4 4 4 4 4 4 4 5 5

2 3 3 3 3 3 3 7 7 6 6 6 3 3 3 3 3 3 3 3 3 3 8 8 8 8 8 3 3 3 3 3 5 5 5 5 5 4 4 4 4 4 4 4 4 5 5

Average File

46 46 1 63

1 1 3 8 8 10 10 8 6 8 6 6 6 5 3 3 3 3 3 3 3 5 5 5 5 5 8 8 8 8 8 3 3 4 5 5 6 6 5 6 6 4 4 4 3 4

1 2 4 6 6 8 8 6 6 10 8 8 8 7 3 3 3 3 3 3 3 7 7 7 7 7 9 9 9 9 9 3 3 4 5 5 6 6 5 6 6 4 4 4 3 4

4 4 6 9 9 8 9 8 7 11 10 8 8 7 3 3 3 3 3 3 3 7 7 7 7 7 11 11 12 12 12 4 4 4 6 6 8 8 6 6 6 4 4 4 3 4

4 5 6 9 9 8 9 8 7 11 10 8 8 7 3 3 3 3 3 3 3 9 9 9 9 9 11 11 12 12 12 4 4 4 8 8 10 10 10 10 10 8 8 6 3 4

5 5 7 7 7 6 7 7 9 13 11 10 8 7 3 3 3 3 3 3 3 10 10 10 10 10 9 11 12 12 12 5 4 4 8 8 12 12 12 12 12 8 8 6 3 4

5 6 5 5 5 3 6 8 8 10 10 6 5 5 3 3 3 3 3 3 3 4 11 13 13 13 12 9 9 10 10 10 5 4 5 8 8 10 10 10 9 9 7 7 5 3 4

6 6 5 5 5 3 6 8 8 8 8 4 3 3 3 3 3 3 3 3 6 12 16 16 16 13 8 8 9 9 9 7 4 5 8 8 10 10 10 9 9 7 7 5 3 4

4 6 5 5 4 4 5 6 6 6 6 4 3 3 3 3 3 3 3 4 8 16 21 23 21 18 11 9 8 8 7 6 2 5 7 7 9 9 9 9 9 7 7 5 3 4

4 6 5 5 5 5 5 7 7 6 6 4 3 3 3 3 3 3 5 7 12 19 26 27 26 21 16 15 11 9 9 7 2 5 5 5 7 7 5 5 5 3 3 3 3 4

4 5 5 4 4 4 5 5 5 5 5 3 3 3 3 3 3 4 8 13 18 23 30 32 30 27 23 18 13 10 8 6 3 5 5 5 5 5 3 3 3 3 3 3 3 4

3 5 5 4 5 5 3 3 3 3 3 3 3 3 3 3 3 6 11 16 20 24 30 30 30 29 27 24 20 15 12 9 4 3 3 3 3 5 5 5 5 5 3 3 3 4

7 8 10 11 11 11 10 9 9 9 8 9 9 9 9 10 11 15 21 27 28 31 33 32 31 32 31 28 24 19 15 12 9 7 6 5 5 6 5 5 5 5 3 3 4 5

7 7 8 10 10 10 10 9 9 9 8 9 9 9 9 12 14 20 28 32 34 34 35 32 34 35 35 33 29 23 17 14 9 7 6 5 5 6 5 5 6 6 4 4 5 5

7 7 8 10 10 9 9 9 9 9 8 9 9 9 11 15 20 28 35 38 38 39 38 35 37 39 37 35 33 26 19 16 10 7 6 5 5 6 7 7 8 8 6 4 5 5

7 7 9 10 10 10 10 9 9 9 8 9 9 11 15 21 27 34 40 41 41 41 38 35 38 39 37 38 38 33 26 20 13 10 7 7 6 7 7 7 8 7 6 4 5 5

8 8 9 10 10 10 9 9 9 9 8 9 10 14 20 27 35 40 44 45 43 42 41 39 41 42 40 42 38 33 27 21 14 12 9 7 6 6 5 5 6 6 6 4 5 5

4 4 4 4 4 4 4 3 3 3 3 4 8 13 20 30 37 40 44 44 41 42 43 42 44 46 44 44 39 33 28 23 15 14 10 6 4 4 5 5 6 6 6 4 5 4

4 4 4 3 3 4 4 3 3 3 4 8 13 18 28 35 40 40 44 43 41 42 44 43 43 45 43 42 39 35 32 28 23 21 15 10 7 5 5 5 5 5 5 3 4 4

5 6 5 5 5 4 4 4 3 5 8 14 21 28 35 41 44 42 43 42 40 40 43 42 43 45 43 43 39 37 34 32 27 26 19 13 8 5 3 3 3 3 3 3 4 4

5 5 5 5 5 4 4 4 5 9 14 22 30 35 41 44 45 42 43 43 41 41 45 44 45 47 43 41 36 33 32 33 31 30 25 17 11 6 5 4 3 3 4 4 4 5

10 9 9 11 10 9 9 9 11 14 20 28 35 38 42 45 45 44 46 47 45 45 47 45 45 45 40 37 34 32 31 34 33 32 26 21 13 9 6 5 3 3 4 4 4 5

10 8 9 10 9 8 10 12 16 21 29 35 38 40 42 42 43 44 47 47 46 45 46 44 44 44 40 38 36 34 33 36 36 35 31 26 19 13 10 7 4 3 3 3 4 5

9 10 12 15 16 16 18 19 23 29 35 39 43 44 44 44 44 47 48 49 48 46 46 45 45 44 38 36 33 32 32 37 35 35 33 29 24 19 14 9 6 4 4 4 4 5

8 8 10 13 14 16 20 22 28 34 40 42 42 42 42 42 43 48 48 49 48 46 43 43 43 42 36 34 32 31 32 37 38 38 37 34 30 25 19 12 7 3 4 4 4 5

8 9 11 13 14 16 20 24 29 34 40 42 40 40 41 41 42 48 47 48 49 47 44 44 44 42 38 35 32 31 32 36 36 37 36 35 31 27 21 14 7 3 3 3 4 4

3 5 6 8 10 11 15 19 26 32 40 44 44 44 44 43 41 45 42 43 44 45 43 46 47 45 41 36 34 32 33 37 39 41 42 39 34 29 22 13 7 4 4 4 4 4

3 5 6 8 10 11 13 15 20 27 35 40 44 45 46 43 41 42 38 39 40 42 41 44 43 42 39 34 30 29 29 33 36 39 39 37 31 26 18 11 7 4 4 4 4 4

4 4 3 3 3 3 5 8 13 19 29 36 40 43 45 42 39 41 37 39 40 42 40 44 42 43 43 38 34 32 28 30 35 36 36 34 26 19 14 8 5 3 3 3 4 4

6 8 8 9 9 9 8 10 14 17 26 33 38 42 44 41 40 40 39 41 43 43 43 44 42 42 43 38 34 31 28 28 31 32 31 29 20 14 9 5 3 3 4 4 4 5

6 8 8 9 9 9 8 8 10 14 20 27 35 40 42 43 40 40 39 41 41 43 42 43 41 42 42 38 35 31 26 27 29 27 26 23 15 9 6 4 4 4 4 4 4 5

6 8 8 9 9 8 7 8 8 10 14 19 27 34 38 41 41 42 43 45 45 45 44 43 40 42 41 39 37 33 26 26 24 20 17 16 9 6 4 3 4 4 4 3 4 5

6 7 8 9 9 9 8 8 8 8 10 14 19 27 33 38 41 44 45 47 47 46 45 44 43 43 42 41 41 36 29 27 22 15 12 11 6 4 6 5 6 6 6 3 4 5

6 7 9 13 13 14 13 12 10 10 8 10 14 19 27 34 39 43 44 44 44 44 43 42 42 42 40 38 37 32 25 22 17 11 8 7 4 4 6 5 5 5 5 3 4 5

3 3 4 7 6 8 8 7 4 4 3 4 8 13 20 29 34 39 40 40 39 40 37 39 40 39 36 37 34 29 23 19 12 7 4 4 3 3 5 5 5 5 5 3 3 4

3 3 4 7 6 8 8 7 4 5 3 3 5 8 13 20 28 35 36 37 38 37 34 38 39 37 36 35 30 25 20 13 8 5 3 3 3 3 5 5 5 5 5 3 3 4

3 3 4 8 8 10 10 8 4 5 3 3 3 5 7 13 19 28 30 33 35 34 30 34 34 33 32 31 25 20 15 10 6 4 3 3 3 3 5 5 5 5 5 3 3 4

3 4 4 8 8 10 10 8 5 5 3 3 3 3 4 7 12 19 23 29 32 33 31 35 35 31 28 27 19 12 8 5 2 3 3 3 3 3 3 3 3 3 3 3 4

4 4 3 4 4 5 5 5 3 3 3 3 3 3 3 4 8 14 17 24 30 32 31 36 34 29 25 22 14 8 6 4 3 3 3 3 3 3 3 3 3 3 3 3 4

3 4 5 6 6 6 6 5 3 3 3 3 3 3 3 3 4 8 11 19 27 31 34 37 35 29 25 19 12 6 4 3 3 5 5 5 5 5 5 5 5 5 3 3 4

3 3 6 8 8 8 8 5 5 7 7 7 7 5 3 3 3 4 7 13 22 28 31 33 31 24 18 13 8 4 3 3 3 5 5 5 5 5 5 5 5 5 3 3 4

2 3 6 7 9 10 10 7 7 7 7 7 7 5 3 3 3 3 4 8 15 23 28 30 28 21 13 8 4 3 3 3 5 7 7 7 7 5 5 5 5 5 5 3 3 4

2 2 6 7 9 9 9 6 7 7 7 7 8 6 4 3 3 3 3 4 10 17 21 23 21 16 8 4 3 3 3 5 7 9 9 9 9 6 6 6 6 6 6 4 5 5

2 2 5 7 10 10 10 8 9 8 9 9 8 6 4 3 3 3 3 3 6 15 17 17 17 13 4 3 3 3 3 5 7 9 9 9 9 6 6 6 6 6 6 4 5 5

2 2 4 6 8 8 8 8 9 8 9 9 7 5 3 3 3 3 3 3 4 11 11 11 11 10 3 3 3 3 3 5 7 7 7 7 7 4 4 4 4 4 4 4 5 5

2 3 2 4 7 7 7 8 7 4 4 4 3 3 3 3 3 3 3 3 3 8 8 8 8 8 3 3 3 3 3 5 7 7 7 7 7 4 4 4 4 4 4 4 5 5

2 3 3 3 3 3 3 7 7 6 6 6 3 3 3 3 3 3 3 3 3 3 8 8 8 8 8 3 3 3 3 3 5 5 5 5 5 4 4 4 4 4 4 4 4 5 5

Histogram Average File

46 46 1 63

0 0

1 3

2 14

3 320

4 207

5 214

6 115

7 114

8 140

9 118

10 78

11 32

12 30

13 29

14 22

15 16

16 14

17 10

18 7

19 20

20 18

21 14

22 7

23 12

24 8

25 7

26 15

27 19

28 19

29 18

30 16

31 21

32 24

33 21

34 28

35 30

36 19

37 21

38 25

39 25

40 32

41 31

42 44

43 41

44 41

45 26

46 10

47 11

48 7

49 3

50 0

51 0

52 0

53 0

54 0

55 0

56 0

57 0

58 0

59 0

60 0

61 0

62 0

63 0

Project 1 Histogram Averaging

```
Ć0 (0):

1 (3):+++

2 (14):++++++++++++++

3 (320):++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

4 (207):+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

5 (214):++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

6 (115):+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

7 (114):++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

8 (140):++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

9 (118):++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

10 (78):++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

11 (32):++++++++++++++++++++++++++++++++

12 (30):++++++++++++++++++++++++++++++

13 (29):+++++++++++++++++++++++++++++

14 (22):++++++++++++++++++++++

15 (16):++++++++++++++++

16 (14):++++++++++++++

17 (10):++++++++++

18 (7):+++++++

19 (20):++++++++++++++++++++

20 (18):++++++++++++++++++

21 (14):++++++++++++++

22 (7):+++++++

23 (12):++++++++++++

24 (8):++++++++

25 (7):+++++++

26 (15):+++++++++++++++

27 (19):++++++++++++++++++

28 (19):++++++++++++++++++

29 (18):++++++++++++++++++

30 (16):++++++++++++++++

31 (21):++++++++++++++++++++

32 (24):++++++++++++++++++++++++

33 (21):++++++++++++++++++++

34 (28):++++++++++++++++++++++++++++

35 (30):++++++++++++++++++++++++++++++

36 (19):++++++++++++++++++

37 (21):++++++++++++++++++++

38 (25):+++++++++++++++++++++++++

39 (25):+++++++++++++++++++++++++

40 (32):++++++++++++++++++++++++++++++++

41 (31):+++++++++++++++++++++++++++++++

42 (44):++

43 (41):+++

44 (41):+++

45 (26):++++++++++++++++++++++++++

46 (10):++++++++++

47 (11):++++++++++

48 (7):+++++++

49 (3):+++

50 (0):

51 (0):

52 (0):

53 (0):

54 (0):

55 (0):

56 (0):

57 (0):

58 (0):

59 (0):

60 (0):

61 (0):

62 (0):

63 (0):

The Bi-Gaussian Value is 25

Project 0 Binary Average Image

46 46 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 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 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 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 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 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 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 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 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 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 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 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 1 1 1 1 0 0 0 0 0 0 0 0 0 0

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

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

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

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

Gauss File

46 46 1 63

1 1 2 9 13 10 6 4 3 3 3 2 2 3 3 3 2 2 3 3 3 2 2 3 3 3 8 10 8 6 4 2 2 3 3 3 2 2 3 3 3 2 2 3 3 4

1 1 2 3 3 3 2 2 3 3 3 2 2 3 3 3 2 2 3 3 3 2 2 3 3 3 2 2 3 3 3 2 2 3 3 3 2 2 3 3 3 2 2 3 3 4

1 1 2 3 3 13 9 6 3 3 3 2 2 3 3 3 2 2 3 3 3 2 2 3 3 3 2 2 11 8 6 2 2 3 3 3 2 2 3 3 3 2 2 3 3 4

2 2 2 9 22 15 8 12 8 6 11 18 12 6 3 3 2 2 3 3 3 2 2 14 10 7 8 10 18 13 8 2 2 3 3 11 10 6 12 8 6 10 8 5 3 4

2 2 7 11 8 11 12 8 11 19 19 22 14 8 3 3 2 2 3 3 3 10 13 22 14 9 9 12 22 15 9 2 2 8 13 10 12 12 8 11 13 8 5 4 3 4

3 2 14 11 7 3 2 2 11 15 16 9 6 4 3 3 2 2 3 3 3 10 13 11 7 5 10 15 21 14 15 6 4 3 3 3 10 8 5 3 3 2 2 3 3 4

11 15 10 7 5 3 8 10 8 6 4 2 2 3 3 3 2 2 3 3 3 2 3 11 8 6 9 12 14 13 9 4 3 3 8 11 18 12 7 3 13 19 12 6 3 4

4 3 2 3 3 3 2 2 3 11 8 5 2 3 3 3 4 3 3 3 3 7 10 15 10 7 2 4 4 12 8 5 2 3 10 13 9 6 23 28 24 12 7 4 3 4

4 3 2 3 3 3 2 8 21 15 9 4 2 3 4 4 3 3 3 3 3 7 18 23 23 14 7 9 12 9 5 3 2 3 3 13 17 18 11 7 5 2 2 3 3 4

5 3 2 3 3 3 9 12 9 6 5 2 2 3 3 3 2 2 3 3 8 17 33 34 30 16 7 2 2 11 8 5 2 9 13 10 6 4 3 3 3 2 2 3 3 4

6 6 4 12 8 6 2 2 3 3 3 2 2 3 3 3 2 2 3 3 9 26 29 33 28 25 13 11 19 14 8 4 2 3 3 3 2 2 3 3 3 2 2 3 3 4

5 9 11 8 6 4 4 3 3 3 3 2 2 3 3 3 2 2 3 9 22 28 36 33 32 27 30 21 11 14 10 5 2 3 3 3 2 2 3 3 3 2 2 3 3 4

1 1 2 3 4 5 3 3 3 3 3 2 2 3 3 3 2 2 9 25 36 36 26 25 35 36 32 31 22 12 6 7 5 4 3 3 2 2 3 5 4 3 2 3 3 4

2 2 2 2 3 3 3 3 3 3 3 2 2 3 3 3 2 8 18 32 35 23 19 28 37 35 33 25 22 22 20 11 5 3 3 3 2 3 5 13 9 5 2 3 3 4

5 10 14 16 16 17 17 17 16 15 14 14 15 16 16 16 17 24 36 40 36 34 35 31 29 32 33 34 35 29 20 15 13 11 10 9 6 10 14 10 6 4 2 3 6 5

16 18 23 23 23 23 22 20 19 20 21 21 21 21 23 25 28 42 50 53 50 38 38 38 40 36 36 36 36 35 26 18 14 13 11 8 7 6 4 4 3 2 10 9 7 6

6 6 5 4 4 3 2 2 3 3 3 2 2 3 3 9 28 40 46 42 40 42 45 44 45 46 47 47 32 26 22 13 5 3 3 3 2 2 3 3 17 15 10 5 4 4

4 3 2 3 5 4 3 2 3 3 3 2 2 3 9 28 40 46 46 45 45 45 45 45 37 41 38 38 41 38 30 25 13 6 3 12 8 5 9 13 10 6 4 3 3 4

4 3 4 5 4 4 2 2 3 3 3 2 2 9 27 38 44 44 44 36 38 41 30 31 38 44 46 47 47 47 32 17 17 17 16 9 5 4 3 3 3 2 2 3 3 4

5 3 2 3 3 5 4 5 4 4 3 2 8 26 38 44 46 44 40 42 38 37 41 47 51 50 48 48 41 24 18 24 25 25 14 6 2 2 3 3 3 2 2 5 5 5

5 4 3 4 5 6 5 4 3 3 3 8 27 39 46 40 44 46 48 39 41 46 52 52 50 40 42 39 33 33 31 38 37 32 26 14 5 4 4 4 3 4 4 4 4 4

3 2 12 9 6 3 2 4 4 4 9 28 40 42 40 42 45 40 38 41 40 41 45 41 39 42 44 46 34 34 39 41 39 35 25 17 9 4 3 3 3 4 3 3 3 4

8 12 9 6 5 4 4 3 3 10 28 41 50 50 49 48 46 46 44 42 43 45 45 46 46 47 42 40 33 34 26 33 36 36 29 22 16 12 7 5 5 4 3 3 6 5

6 14 14 14 11 12 14 14 21 29 37 42 43 43 41 41 42 45 49 52 52 49 45 43 45 44 38 32 27 26 31 36 35 34 30 25 19 13 10 6 4 2 3 5 5 6

19 17 16 16 17 19 19 19 28 35 39 39 40 40 32 37 44 51 54 55 53 50 44 40 42 42 42 41 34 28 29 32 33 32 29 26 26 21 15 8 4 2 2 3 3
4

3 2 2 12 17 20 25 36 41 37 42 47 43 39 41 44 45 47 50 51 50 46 46 47 49 49 47 32 25 33 38 44 45 44 35 36 35 30 23 12 5 3 4 5 5 4

3 9 17 22 26 28 27 22 29 40 47 48 48 48 38 40 43 49 48 50 40 42 44 46 37 34 29 32 28 25 35 43 41 39 42 37 39 36 30 16 9 5 4 4 4 4

4 3 2 3 3 3 8 17 24 37 44 39 36 36 41 46 49 45 41 39 42 45 43 40 42 44 40 33 33 30 29 38 45 43 32 37 42 40 23 11 4 2 2 3 3 4

4 3 2 3 3 3 5 10 18 19 29 39 47 47 49 44 42 43 36 38 37 44 47 49 49 48 50 43 34 25 32 31 34 36 42 45 40 23 11 4 3 4 3 3 3 4

5 3 2 3 5 5 4 3 9 18 27 40 47 50 49 48 42 29 24 27 37 44 38 43 47 50 39 36 35 26 20 29 31 39 44 41 22 10 4 4 5 4 3 3 3 4

5 4 3 3 3 3 2 2 3 9 19 24 37 44 48 42 27 30 37 43 40 38 41 38 35 36 41 38 31 31 30 24 25 35 36 23 10 4 3 3 3 2 4 4 4 4

4 6 13 20 17 14 12 13 14 14 20 30 37 40 36 33 37 43 48 52 53 46 41 40 41 41 41 41 40 34 24 24 29 29 21 14 8 5 4 3 4 4 3 3 6 5

11 19 22 20 19 18 18 18 18 19 21 21 21 31 38 43 47 51 47 49 48 46 43 41 40 39 38 37 28 30 32 27 20 24 18 12 5 4 3 5 6 4 4 5 5 6

2 4 3 3 3 3 2 2 3 3 3 8 17 24 37 44 43 41 43 45 46 48 48 48 48 48 42 39 42 39 32 25 18 10 6 4 2 3 6 6 5 3 2 3 3 4

5 3 3 3 3 5 4 3 3 3 3 2 8 18 24 37 44 46 45 45 46 46 46 46 47 46 47 46 47 38 30 25 14 6 3 3 2 2 3 3 13 9 6 3 3 4

3 2 2 4 13 20 12 14 8 6 3 2 2 9 18 24 32 39 42 41 38 38 39 41 41 41 42 39 38 37 26 14 7 4 3 3 2 2 9 13 10 6 4 3 3 4

3 2 7 17 18 18 15 9 5 4 3 2 2 3 7 15 21 32 36 40 30 24 27 33 29 34 28 23 16 19 12 6 2 3 3 3 2 2 3 3 3 2 2 3 3 4

4 3 2 3 3 3 2 2 3 3 3 5 4 3 3 9 18 25 32 30 21 28 26 29 29 27 24 22 26 16 9 4 2 3 3 3 2 2 3 3 3 2 2 3 3 4

4 3 2 3 3 12 8 5 3 5 5 4 3 3 3 3 8 17 18 28 35 39 38 34 29 32 34 35 20 9 4 2 2 3 3 3 2 2 3 3 3 2 2 3 3 4

5 3 2 9 12 9 5 6 4 4 3 2 2 3 3 3 2 8 16 22 34 36 31 32 37 41 38 22 10 4 3 2 2 3 3 3 2 2 3 3 3 2 2 3 3 4

5 4 3 3 3 4 4 3 3 3 3 2 2 3 3 3 2 2 9 18 25 37 45 41 43 39 22 10 4 3 3 2 2 3 3 3 2 2 3 3 3 2 2 3 3 4

3 2 2 3 11 8 5 2 3 3 3 2 4 4 4 3 2 2 3 9 19 20 30 40 39 22 10 4 3 3 3 2 2 3 3 13 9 6 3 3 13 9 6 3 3 4

1 1 7 11 18 12 7 2 3 3 15 21 16 8 4 3 2 2 3 3 9 18 32 36 23 11 4 2 3 3 3 2 2 9 13 10 6 4 9 13 10 6 4 3 3 4

2 2 8 12 9 14 20 12 13 20 21 15 9 5 3 3 2 2 3 3 8 17 17 21 14 8 2 2 3 3 3 2 2 3 15 11 6 2 3 3 3 2 2 3 3 4

2 2 2 8 18 19 12 7 4 3 3 2 2 3 5 4 3 2 3 3 3 8 11 18 12 8 2 2 3 3 3 2 10 26 19 11 4 2 11 8 6 2 2 9 8 6

3 2 2 3 3 3 4 3 3 11 8 5 3 5 4 4 2 2 3 3 3 8 11 18 12 8 2 2 3 3 3 10 14 11 7 5 8 10 8 6 4 6 8 7 6 5

Histogram Gauss File

46 46 1 63

0 0

1 10

2 270

3 479

4 158

5 79

6 61

7 25

8 59

9 55

10 40

11 36

12 35

13 30

14 31

15 18

16 18

17 22

18 29

19 20

20 15

21 18

22 20

23 15

24 16

25 19

26 16

27 11

28 17

29 19

30 16

31 11

32 23

33 15

34 16

35 17

36 28

37 23

38 30

39 25

40 31

41 35

42 29

43 19

44 26

45 24

46 28

47 22

48 18

49 12

50 13

51 4

52 5

53 3

54 1

55 1

56 0

57 0

58 0

59 0

60 0

61 0

62 0

63 0

Project 1 Gauss Histogram and Threshold Value

Ć0 (0):

1 (10):++++++++++

2 (270):++

3 (479):+++

4 (158):++

5 (79):+++

6 (61):+++

7 (25):+++++++++++++++++++++++++

8 (59):+++

9 (55):+++

10 (40):++

11 (36):++++++++++++++++++++++++++++++++++++

12 (35):+++++++++++++++++++++++++++++++++++

13 (30):++++++++++++++++++++++++++++++

14 (31):+++++++++++++++++++++++++++++++

15 (18):++++++++++++++++++

16 (18):++++++++++++++++++

17 (22):++++++++++++++++++++++

18 (29):+++++++++++++++++++++++++++++

19 (20):++++++++++++++++++++

20 (15):+++++++++++++++

21 (18):++++++++++++++++++

22 (20):++++++++++++++++++++

23 (15):+++++++++++++++

24 (16):++++++++++++++++

25 (19):+++++++++++++++++++

26 (16):++++++++++++++++

27 (11):+++++++++++

28 (17):+++++++++++++++++

29 (19):+++++++++++++++++++

30 (16):++++++++++++++++

31 (11):+++++++++++

32 (23):+++++++++++++++++++++++

33 (15):+++++++++++++++

34 (16):++++++++++++++++

35 (17):+++++++++++++++++

36 (28):++++++++++++++++++++++++++++

37 (23):+++++++++++++++++++++++

38 (30):++++++++++++++++++++++++++++++

39 (25):+++++++++++++++++++++++++

40 (31):+++++++++++++++++++++++++++++++

41 (35):+++++++++++++++++++++++++++++++++++

42 (29):+++++++++++++++++++++++++++++

43 (19):+++++++++++++++++++

44 (26):++++++++++++++++++++++++++

45 (24):++++++++++++++++++++++++

46 (28):++++++++++++++++++++++++++++

47 (22):++++++++++++++++++++++

48 (18):++++++++++++++++++

49 (12):++++++++++++

50 (13):+++++++++++++

51 (4):++++

52 (5):+++++

53 (3):+++

54 (1):+

55 (1):+

56 (0):

57 (0):

58 (0):

59 (0):

60 (0):

61 (0):

62 (0):

63 (0):

The Bi-Gaussian Value is 31

Project 0 Binary Gauss Image

46 46 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	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	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	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	1	1	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	0	0	0	0	0	0	0	0	0	0	0	0	1	1	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	1	1	1	1	1	1	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	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	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	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	1	1	1	1	1	1	1	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	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	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
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	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	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	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	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	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	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	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	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	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	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	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	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	0	0	0	1	1	1	1	1	1	1	1	1	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	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	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	0	0	0	0	0	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	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	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	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	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	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	1	1	1	1	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 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 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 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 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 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0