****
CSCI 381-26 Project 6: Connected-Component Language: C++

Name: Akshar Patel

Due date: Soft copy: 3/22/2020
Hard copy: 3/22/2020

Submission date: Soft copy: 3/23/2020
Hard copy: 3/23/2020

****

Part 1: Algorithm

```
*****************************
I. main(...)
*****************************
step 0: inFile ← open the input file
        prettyPrintFile, labelFile, propertyFile ← open from argc[]
        numRows, numCols, minVal, maxVal ← read from inFile
        dynamically allocate zeroFramedAry.

step 1: zero2D (zeroFramedAry) // ** Initialized zeroFramedAry to zero.
step 2: loadImage(inFile, zeroFramedAry)
          // read from input file and write to zeroFramedAry begin at(1,1)
step 3: Connectness ← from argv[2]

step 4: newLabel ← connectPass1 (Connectness, zeroFramedAry, NonZeroNeighborAry)
                // see algorithm below

step 5: prettyPrint (prettyPrintFile) // Print zeroFramedAry to prettyPrintFile
        printEQAry (newLable, prettyPrintFile)
                // print the EQAry up to newLable with proper caption
step 6: connectPass2 (Connectness, zeroFramedAry, NonZeroNeighborAry)

step 7: prettyPrint (prettyPrintFile) // Print zeroFramedAry to prettyPrintFile
        printEQAry (newLable, prettyPrintFile)
                // print the EQAry up to newLable with proper caption
step 8: manageEQAry (EQAry, newLabel)
        printEQAry (numCCLable, prettyPrintFile)
          // print the EQAry up to newLabel with proper caption
step 9: connectPass3 (...) // See algorithm below
       prettyPrint (prettyPrintFile) // Print zeroFramedAry to prettyPrintFile
        printEQAry (numCCLable, prettyPrintFile)
          // print the EQAry up to numCCLabel with proper caption
step 10: output numRows, numCols, newMin, newMax to labelFile

step 11: printImg (labelFile) // Output the result of pass3 from zeroFramedAry to
        //labelFile, begins at zeroFramedAry[1, 1] and ending at ??

step 12: printCCproperty (propertyFile) // print cc properties to propertyFile
step 13: drawBoxes(zeroFramedAry, CCproperty)

step 14: prettyPrint (prettyPrintFile) // Print zeroFramedAry to prettyPrintFile

step 15: close all files
```

Part 2: Source code

```
#include <iostream>
#include<fstream>

using namespace std;

class CCLabel{
    public:
    int numRows, numCols, minVal, maxVal, newMin = 9999, newMax = 0, newLabel = 0, trueNumCC, numNb = 5,
Connectness;
    int* NonZeroNeighborAry;
    int* EQAry;
    int** zeroFramedAry;
    struct  Property{
        public:
        int label, numpixels, upperLftR, upperLftC, lowerRgtR, lowerRgtC;
    };
    Property* CCproperty;
    void set2DZero(int** Ary){
        for(int i = 0 ; i < numRows + 2 ; i++){
            for(int j = 0 ; j < numCols + 2 ; j++){
                Ary[i][j] = 0;
            }
        }
    }
    void loadImage(ifstream& file, int** Ary){
        for(int i = 1 ; i < numRows + 1 ; i++){
            for(int j = 1 ; j < numCols + 1 ; j++){
                file>>Ary[i][j];
            }
        }
    }
    int connectPass1(int connectness, int** Ary, int* nonZeroAry){
        int newLabel = 0, nonZeroCount;
        bool diffFlag;
        for(int i = 1 ; i < numRows + 1 ; i++){
            for(int j = 1 ; j < numCols + 1 ; j++){
                if(Ary[i][j] > 0){
                    int minLabel = loadNonZeroPass1(Ary, Connectness, i, j, nonZeroAry, diffFlag,
nonZeroCount);
                    if(minLabel == 0){
                                newLabel++;
                        Ary[i][j] = newLabel;
                    }
                    else if(minLabel > 0){
                        Ary[i][j] = minLabel;
                        if (diffFlag == true){
                            updateEQ(EQAry, nonZeroAry, minLabel, nonZeroCount);
                        }
                    }
                }
            }
        }
        return newLabel;
    }
    int loadNonZeroPass1(int** Ary, int connnectness, int i, int j, int* nonZeroAry,bool &diffFlag,int
&nonZeroCount){
        minus1D(nonZeroAry);
        nonZeroCount = 0;
        if(Ary[i-1][j] > 0){
            nonZeroAry[nonZeroCount] = Ary[i-1][j];
                 nonZeroCount++;
        }
        if(Ary[i][j-1] > 0){
                 nonZeroAry [nonZeroCount] = Ary[i][j-1];
                 nonZeroCount++;
        }
        if(Connectness == 8){
```

```
                if(Ary[i-1][j-1] > 0){
                        nonZeroAry[nonZeroCount] = Ary[i-1][j-1];
                        nonZeroCount++;
            }
            if(Ary[i-1][j+1] > 0){
                        nonZeroAry[nonZeroCount] = Ary[i-1][j+1];
                        nonZeroCount++;
            }
        }
        if(nonZeroCount <= 0){
            return 0;
        }
        int minLabel = nonZeroAry[0];
           diffFlag = false;
        int index = 1;
        while(index < nonZeroCount){
            if(minLabel != nonZeroAry[index]){
                diffFlag = true;
            }
            if(minLabel > nonZeroAry[index]){
                minLabel = nonZeroAry[index];
            }
            index++;
        }
        return minLabel;
    }
    void minus1D(int* Ary){
        for(int i = 0; i < numNb ; i++ ){
            Ary[i] = -1;
        }
    }
    void updateEQ(int* Ary, int* nonZeroAry, int minLabel, int nonZeroCount){
        int index = 0;
        while(index < nonZeroCount ){
            EQAry[nonZeroAry[index]] = minLabel;
            index++;
        }
    }
    void prettyPrint(ofstream& file){
        for(int i = 0 ; i < numRows + 2 ; i++){
            for(int j = 0 ; j < numCols + 2 ; j++){
                if(zeroFramedAry[i][j] > 0){
                   file<<zeroFramedAry[i][j]<<" ";
                }
                else{
                    file<<"  ";
                }
            }
            file<<endl;
        }
    }
    void printEQAry(int i, ofstream& file){
        int index = 0;
        while(index <= i){
            file<<EQAry[index]<<" ";
            index++;
        }
    }
    void connectPass2(int Connectness, int** Ary, int* nonZeroAry){
        int nonZeroCount;
        bool diffFlag;
        for(int i = numRows ; i > 0 ; i--){
            for(int j = numCols ; j > 0 ; j--){
                if(Ary[i][j] > 0){
                    int minLabel = loadNonZeroPass2(Ary, Connectness, i, j, nonZeroAry, diffFlag,
nonZeroCount);
                    if(minLabel != Ary[i][j]){
                        Ary[i][j] = minLabel;
                    }
```

```
                    if(diffFlag == true){
                        updateEQ(EQAry, nonZeroAry, minLabel, nonZeroCount);
                    }
                }
            }
        }
    }
    int loadNonZeroPass2(int** Ary, int Connectness, int i, int j, int* nonZeroAry, bool &diffFlag, int
&nonZeroCount){
        minus1D(nonZeroAry);
        nonZeroCount = 0;
        nonZeroAry[nonZeroCount] = Ary[i][j];
        nonZeroCount++;
        if(Ary[i+1][j] > 0){
                 nonZeroAry[nonZeroCount] = Ary[i+1][j];
                 nonZeroCount++;
        }
        if(zeroFramedAry[i][j+1] > 0){
                 nonZeroAry[nonZeroCount] = Ary[i][j+1];
                 nonZeroCount++;
        }
        if(Connectness == 8){
                 if(Ary[i+1][j-1] > 0){
                        nonZeroAry[nonZeroCount] = Ary[i+1][j-1];
                        nonZeroCount++;
            }
            if(zeroFramedAry[i+1][j+1] > 0){
                        nonZeroAry[nonZeroCount] = Ary[i+1][j+1];
                        nonZeroCount++;
            }
        }
        int minLabel = nonZeroAry[0];
           diffFlag = false;
        int index = 1;
        while(index < nonZeroCount){
            if(minLabel != nonZeroAry[index]){
                diffFlag = true;
            }
            if(minLabel > nonZeroAry[index]){
                     minLabel = nonZeroAry[index];
            }
            index++;
        }
        return minLabel;
    }
    int manageEQAry(int* EQAry, int newLabel){
        int index = 1;
        trueNumCC = 0;
        while (index <= newLabel){
            if(index != EQAry[index]){
                EQAry[index]=EQAry[EQAry[index]];
            }
            else{
                trueNumCC++;
                EQAry[index] = trueNumCC;
            }
            index++;
        }
        return trueNumCC;
    }
    void connectPass3(int* EQAry,int** Ary){
        for(int i = 1 ; i < numRows + 1 ; i++){
            for(int j = 1 ; j < numCols + 1 ; j++){
                if(Ary[i][j]>0){
                    Ary[i][j] = EQAry[Ary[i][j]];
                    CCproperty[Ary[i][j]].numpixels++;
                                if(CCproperty[Ary[i][j]].upperLftR == 0)
                    CCproperty[Ary[i][j]].upperLftR = i;
                                if(CCproperty[Ary[i][j]].lowerRgtR < i)
```

```
                        CCproperty[Ary[i][j]].lowerRgtR = i;
                                if(CCproperty[Ary[i][j]].upperLftC == 0)
                        CCproperty[Ary[i][j]].upperLftC = j;
                                if(CCproperty[Ary[i][j]].lowerRgtC < j)
                        CCproperty[Ary[i][j]].lowerRgtC = j;
                }
                if(newMin > Ary[i][j]){
                    newMin = Ary[i][j];
                }
                if(newMax < Ary[i][j]){
                    newMax = Ary[i][j];
                }
            }
        }
    }
    void printImg(ofstream& file){
        for(int r = 1; r < numRows + 1; r++){
            for(int c =1; c < numCols + 1; c++){
                file<<zeroFramedAry[r][c]<<" ";
            }
            file<<endl;
        }
    }
    void printCCproperty(ofstream& file){
        file<<numRows<<" "<<numCols<<" "<<newMin<<" "<<newMax<<endl;
             file<<trueNumCC<<endl;
        file<<"-- -- -- -- -- -- -- --"<<endl;
             int index = 1;
             while(index < trueNumCC + 1){
            file<<index<<endl;
                    file<<CCproperty[index].numpixels<<endl;
                    file<<CCproperty[index].upperLftR<<" "<<CCproperty[index].upperLftC<<endl;
                    file<<CCproperty[index].lowerRgtR<<" "<<CCproperty[index].lowerRgtC<<endl;
                    file<<"-- -- -- -- -- -- -- --"<<endl;
                    index++;
             }
    }
    void drawBoxes(int** Ary,Property* CCproperty){
        int index =1;
        while(index <= trueNumCC){
            int minRow = CCproperty[index].upperLftR;
                 int maxRow = CCproperty[index].lowerRgtR;
                 int minCol = CCproperty[index].upperLftC;
                 int maxCol = CCproperty[index].lowerRgtC;
                          for (int i = minRow; i < maxRow + 1; i++) {
                                Ary[i][minCol] = index;
                                Ary[i][maxCol] = index;
                          }
                          for (int j = minCol; j < maxCol + 1; j++) {
                                Ary[minRow][j] = index;
                                Ary[maxRow][j] = index;
                          }
            index++;
        }
    }
};
int main(int argc, char** argv){
    CCLabel l;
    string inputName = argv[1];
    ifstream inFile;
    inFile.open( inputName );

    string outputName1 = argv[3];
    ofstream prettyPrintFile;
    prettyPrintFile.open( outputName1 );

    string outputName2 = argv[4];
    ofstream labelFile;
    labelFile.open( outputName2 );
```

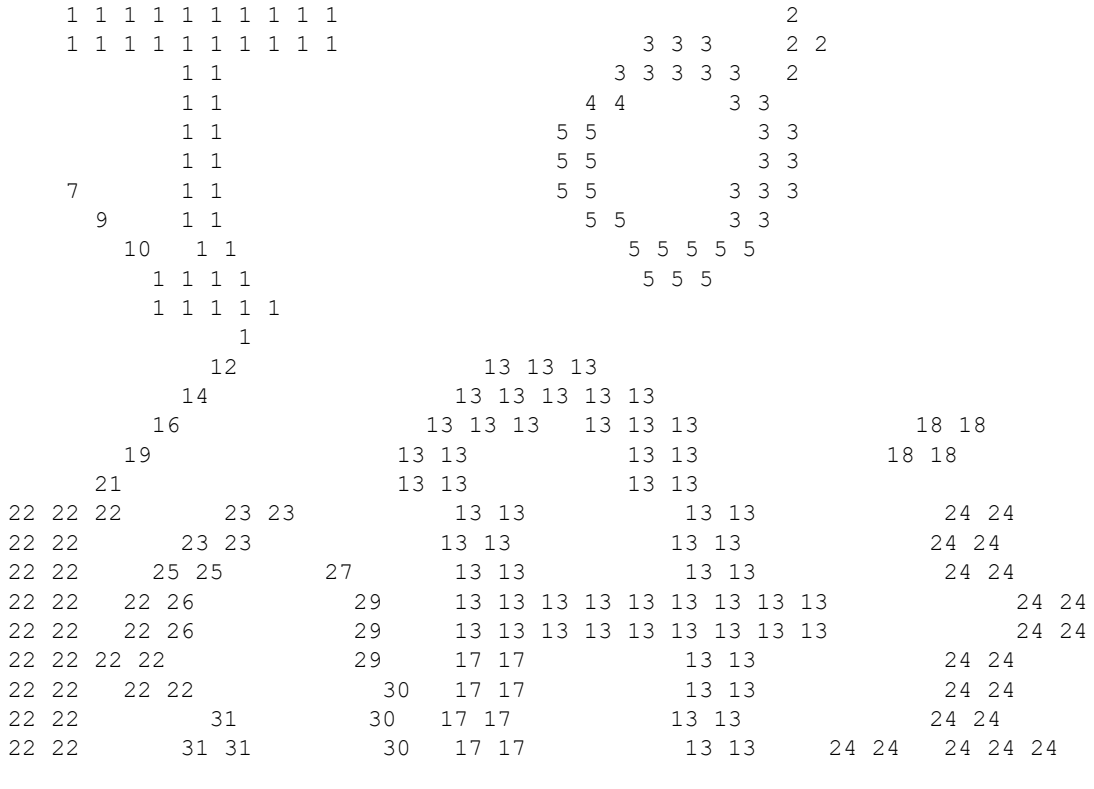
```
    string outputName3 = argv[5];
    ofstream propertyFile;
    propertyFile.open( outputName3 );

    if(inFile.is_open()){
       if(prettyPrintFile.is_open() && labelFile.is_open() && propertyFile.is_open() ){
            inFile>>l.numRows>>l.numCols>>l.minVal>>l.maxVal;
            l.zeroFramedAry = new int* [l.numRows + 2];
            for( int i = 0; i < l.numRows + 2; i++ ){
                l.zeroFramedAry[i] = new int[l.numCols + 2];
            }
            l.set2DZero(l.zeroFramedAry);
            l.loadImage(inFile, l.zeroFramedAry);
            l.Connectness = stoi(argv[2]);
            l.NonZeroNeighborAry = new int[l.numNb];
            l.minus1D(l.NonZeroNeighborAry);
            l.EQAry = new int[(l.numRows * l.numCols) / 2];
            for(int i = 0; i< ((l.numRows * l.numCols) / 2) + 1; i++){
                l.EQAry[i] = i;
            }
            l.newLabel = l.connectPass1(l.Connectness, l.zeroFramedAry,l.NonZeroNeighborAry);
            l.prettyPrint(prettyPrintFile);
            prettyPrintFile<<"EQAry up to newLable after pass 1: "<<endl;
            l.printEQAry(l.newLabel, prettyPrintFile);
            l.connectPass2(l.Connectness, l.zeroFramedAry, l.NonZeroNeighborAry);
            l.prettyPrint(prettyPrintFile);
            prettyPrintFile<<"EQAry up to newLable after pass 2: "<<endl;
               l.printEQAry(l.newLabel, prettyPrintFile);
            l.trueNumCC = l.manageEQAry(l.EQAry, l.newLabel);
            l.CCproperty = new CCLabel::Property[l.trueNumCC + 1];
               for (int i = 0; i < l.trueNumCC + 1; i++) {
                      l.CCproperty[i].label = 0;
                      l.CCproperty[i].lowerRgtR = 0;
                      l.CCproperty[i].lowerRgtC = 0;
                      l.CCproperty[i].numpixels = 0;
                      l.CCproperty[i].upperLftR = 0;
                      l.CCproperty[i].upperLftC = 0;
               }
            prettyPrintFile<<endl<<"EQAry up to newLable after manageEQAry: "<<endl;
            l.printEQAry(l.newLabel, prettyPrintFile);
            l.connectPass3(l.EQAry, l.zeroFramedAry);
            l.prettyPrint(prettyPrintFile);
            prettyPrintFile<<endl<<"EQAry up to newLable after pass3: "<<endl;
            l.printEQAry(l.newLabel, prettyPrintFile);
            labelFile<<l.numRows<<" "<<l.numCols<<" "<<l.newMin<<" "<<l.newMax<<endl;
            l.printImg(labelFile);
            l.printCCproperty(propertyFile);
            l.drawBoxes(l.zeroFramedAry, l.CCproperty);
            l.prettyPrint(prettyPrintFile);
            inFile.close();
            prettyPrintFile.close();
            labelFile.close();
            propertyFile.close();
        }else{cout<<"Error!! Could NOT create output file"<<endl ;}
    }else{cout<<"Error!! Could NOT open input file"<<endl;}

}
```

Part 3: Output

EQAry up to newLable after pass 1:
0 1 2 3 4 5 5 7 5 9 10 1 12 13 14 13 16 17 18 19 17 21 22 23 24 25 26 27 22 29 30 31 31 24 24 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 47 51



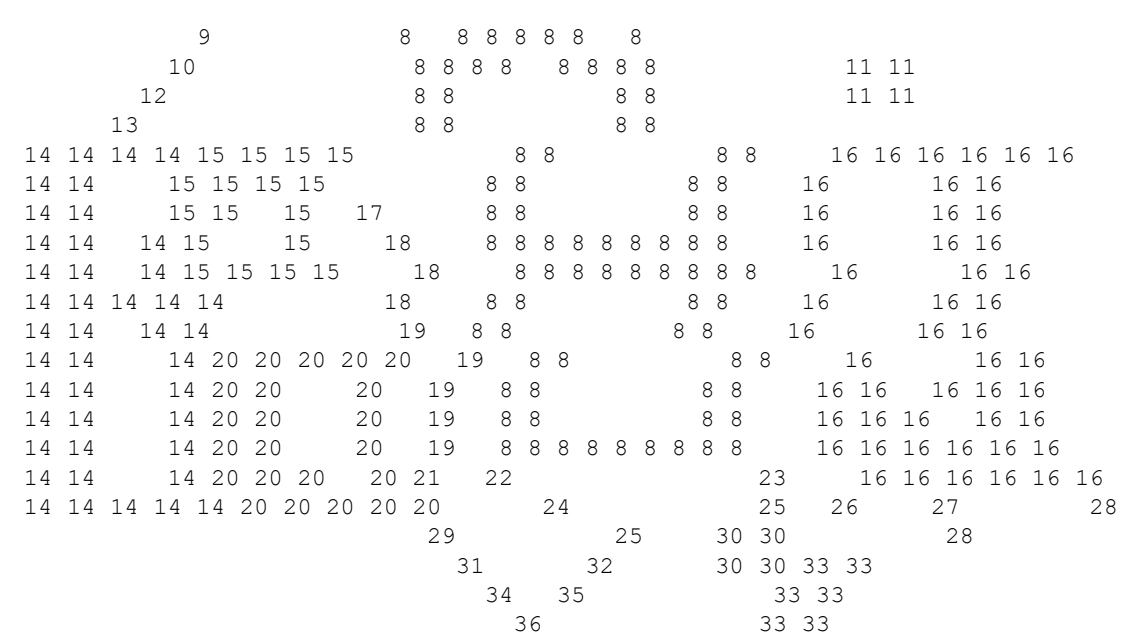
```
        22 22       31 31         30   17 17           13 13       24 24   24 24
        22 22       31 31         30                13         24 24 24 24
        22 22         31 31     35   36                37         24 24 24 24
        22 22           31 31 31       38             39   40     41         42
                                  43           39     44 44           42
                                    45       46         44   47
                                      48   49             47 47
                                        51               50
```

EQAry up to newLable after pass 2:
0 1 2 3 3 4 5 7 3 9 10 1 12 13 14 13 16 13 18 19 13 21 22 23 24 23 25 27 22 29 30 31 31 24 24 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 47 51
EQAry up to newLable after manageEQAry:
0 1 2 3 3 3 3 4 3 5 6 1 7 8 9 8 10 8 11 12 8 13 14 15 16 15 15 17 14 18 19 20 20 16 16 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 33 36

```
            1 1 1 1 1 1 1 1 1 1                           2
            1 1 1 1 1 1 1 1 1 1                     3 3 3     2 2
                    1 1                           3 3 3 3 3   2
                    1 1                         3 3       3 3
                    1 1                       3 3           3 3
                    1 1                       3 3           3 3
            4       1 1                       3 3         3 3 3
              5     1 1                         3 3       3 3
                6   1 1                           3 3 3 3 3
                  1 1 1 1                         3 3 3
                  1 1 1 1 1
                        1
                    7                 8 8 8
                  9                 8 8 8 8 8
                10                8 8 8   8 8 8               11 11
              12                8 8           8 8             11 11
            13                  8 8           8 8
        14 14 14       15 15          8 8           8 8             16 16
        14 14       15 15             8 8           8 8             16 16
        14 14     15 15       17       8 8           8 8             16 16
        14 14   14 15           18     8 8 8 8 8 8 8 8 8             16 16
        14 14   14 15           18     8 8 8 8 8 8 8 8 8             16 16
        14 14 14 14             18     8 8           8 8             16 16
        14 14   14 14             19   8 8           8 8             16 16
        14 14         20         19   8 8           8 8             16 16
        14 14       20 20         19   8 8           8 8     16 16   16 16 16
        14 14       20 20         19   8 8           8 8       16 16   16 16
        14 14       20 20         19                 8         16 16 16 16
        14 14         20 20     21   22                23         16 16 16 16
        14 14           20 20 20       24             25   26     27         28
                                  29           25     30 30           28
                                    31       32         30   33
                                      34   35             33 33
                                        36               33
```

EQAry up to newLable after pass3:
0 1 2 3 3 3 3 4 3 5 6 1 7 8 9 8 10 8 11 12 8 13 14 15 16 15 15 17 14 18 19 20 20 16 16 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 33 36

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



- labelFile for 4-connectness

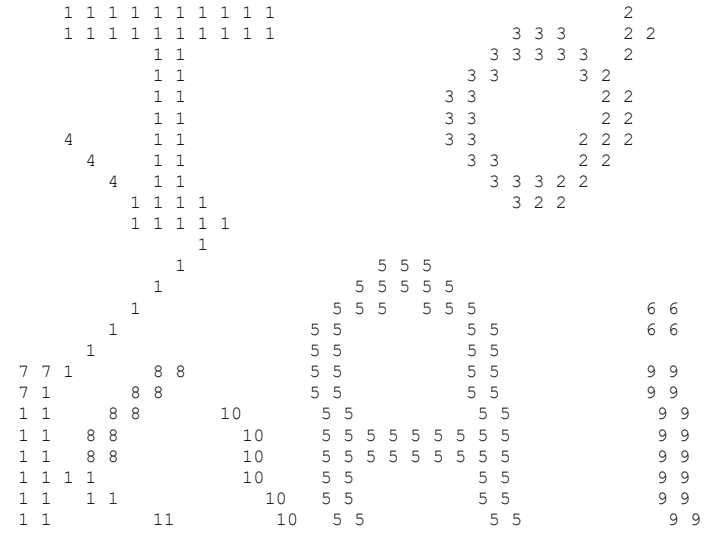
```
35 35 0 36
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0
0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 3 3 3 0 0 2 2 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 3 3 3 3 3 0 2 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 3 3 0 0 0 3 3 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 3 3 0 0 0 0 0 3 3 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 3 3 0 0 0 0 0 3 3 0 0 0 0
0 0 0 0 0 4 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 3 3 0 0 0 0 3 3 3 0 0 0 0
0 0 0 0 0 0 5 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 3 3 0 0 0 3 3 0 0 0 0 0 0
0 0 0 0 0 0 0 6 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 3 3 3 3 3 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 3 3 3 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 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 7 0 0 0 0 0 0 0 0 0 8 8 8 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 9 0 0 0 0 0 0 0 0 0 8 8 8 8 8 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 10 0 0 0 0 0 0 0 0 8 8 8 0 8 8 8 0 0 0 0 0 0 0 11 11 0 0
0 0 0 0 0 0 0 12 0 0 0 0 0 0 0 0 8 8 0 0 0 0 0 8 8 0 0 0 0 0 0 11 11 0 0
0 0 0 0 0 0 13 0 0 0 0 0 0 0 0 0 8 8 0 0 0 0 0 8 8 0 0 0 0 0 0 0 0 0 0 0
0 0 0 14 14 14 0 0 0 15 15 0 0 0 0 0 8 8 0 0 0 0 0 8 8 0 0 0 0 0 0 16 16 0 0
0 0 0 14 14 0 0 0 15 15 0 0 0 0 0 0 8 8 0 0 0 0 0 8 8 0 0 0 0 0 0 16 16 0 0
0 0 0 14 14 0 0 15 15 0 0 0 17 0 0 0 8 8 0 0 0 0 0 8 8 0 0 0 0 0 0 16 16 0 0
0 0 0 14 14 0 14 15 0 0 0 0 0 18 0 0 8 8 8 8 8 8 8 8 0 0 0 0 0 0 16 16 0 0
0 0 0 14 14 0 14 15 0 0 0 0 0 18 0 0 8 8 8 8 8 8 8 8 0 0 0 0 0 0 16 16 0 0
0 0 0 14 14 14 14 0 0 0 0 0 0 18 0 0 8 8 0 0 0 0 0 8 8 0 0 0 0 0 0 16 16 0 0
0 0 0 14 14 0 14 14 0 0 0 0 0 0 19 0 8 8 0 0 0 0 0 8 8 0 0 0 0 0 0 16 16 0 0
0 0 0 14 14 0 0 0 0 20 0 0 0 0 19 0 8 8 0 0 0 0 0 8 8 0 0 0 0 0 0 16 16 0 0
0 0 0 14 14 0 0 0 20 20 0 0 0 0 19 0 8 8 0 0 0 0 0 8 8 0 0 16 16 0 16 16 16 0 0
0 0 0 14 14 0 0 0 20 20 0 0 0 0 19 0 8 8 0 0 0 0 0 8 8 0 0 16 16 0 16 16 0 0
0 0 0 14 14 0 0 0 20 20 0 0 0 0 19 0 0 0 0 0 0 0 0 8 0 0 0 0 16 16 16 16 0 0 0
0 0 0 14 14 0 0 0 0 20 20 0 0 21 0 22 0 0 0 0 0 0 0 0 23 0 0 0 0 16 16 16 16 0 0
0 0 0 14 14 0 0 0 0 0 20 20 20 0 0 0 24 0 0 0 0 0 0 0 25 0 26 0 0 27 0 0 0 0 28 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 29 0 0 0 0 0 25 0 0 30 30 0 0 0 0 0 28 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 31 0 0 0 32 0 0 0 0 30 0 33 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 34 0 35 0 0 0 0 0 0 33 33 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 36 0 0 0 0 0 0 0 33 0 0 0 0 0 0 0
```

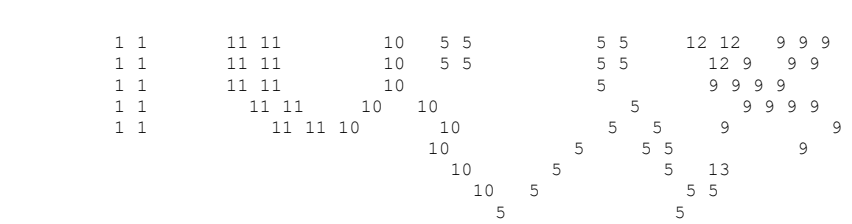
- propertyFile for 4-connectness

```
35 35 0 36
36
-- -- -- -- -- -- -- --
1
44
2 6
13 15
-- -- -- -- -- -- -- --
2
4
2 31
4 32
-- -- -- -- -- -- -- --
3
37
-- -- -- -- -- -- -- --
14
33
19 4
31 8
-- -- -- -- -- -- -- --
15
8
19 8
23 11
-- -- -- -- -- -- -- --
16
33
19 28
30 33
26
1
31 26
31 26
-- -- -- -- -- -- -- --
27
1
31 29
31 29
-- -- -- -- -- -- -- --
28
2
31 34
32 34
-- -- -- -- -- -- -- --
```

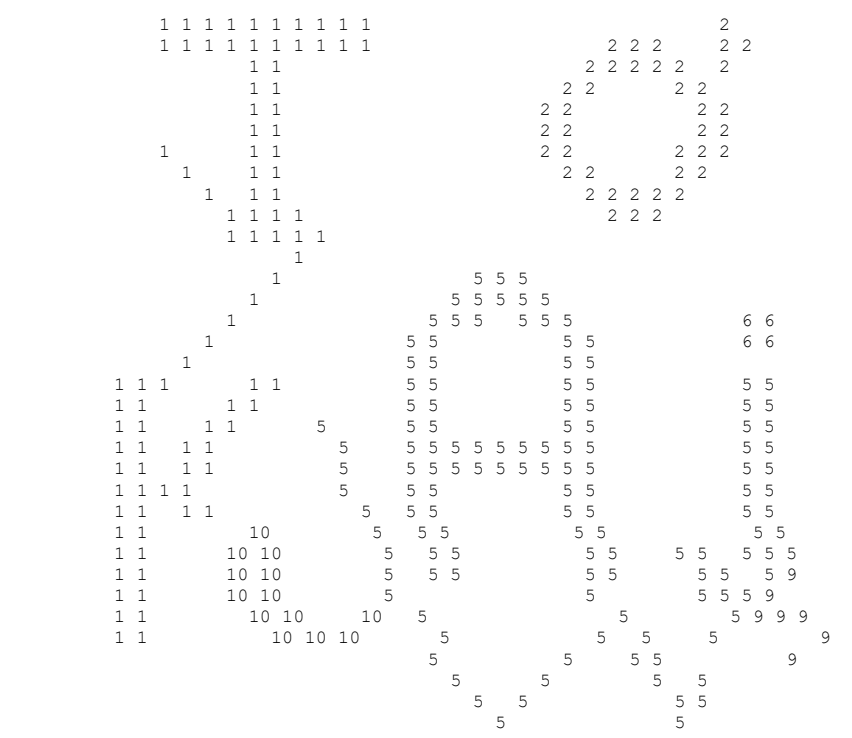
```
3 23
11 31
-- -- -- -- -- -- -- --
4
1
8 6
8 6
-- -- -- -- -- -- -- --
5
1
9 7
9 7
-- -- -- -- -- -- -- --
6
1
10 8
10 8
-- -- -- -- -- -- -- --
7
1
14 11
14 11
-- -- -- -- -- -- -- --
8
73
14 17
29 25
-- -- -- -- -- -- -- --
9
1
15 10
15 10
-- -- -- -- -- -- -- --
10
1
16 9
16 9
-- -- -- -- -- -- -- --
11
4
16 32
17 33
-- -- -- -- -- -- -- --
12
1
17 8
17 8
-- -- -- -- -- -- -- --
13
1
18 7
18 7
-- -- -- -- -- -- -- --
17
1
21 13
21 13
-- -- -- -- -- -- -- --
18
3
22 14
24 14
-- -- -- -- -- -- -- --
19
5
25 15
29 15
-- -- -- -- -- -- -- --
20
12
26 9
31 13
-- -- -- -- -- -- -- --
21
1
30 14
30 14
-- -- -- -- -- -- -- --
22
1
30 16
30 16
-- -- -- -- -- -- -- --
23
1
30 25
30 25
-- -- -- -- -- -- -- --
24
1
31 17
31 17
-- -- -- -- -- -- -- --
25
2
31 24
32 24
-- -- -- -- -- -- -- --
29
1
32 18
32 18
-- -- -- -- -- -- -- --
30
3
32 27
33 28
-- -- -- -- -- -- -- --
31
1
33 19
33 19
-- -- -- -- -- -- -- --
32
1
33 23
33 23
-- -- -- -- -- -- -- --
33
4
33 29
35 30
-- -- -- -- -- -- -- --
34
1
34 20
34 20
-- -- -- -- -- -- -- --
35
1
34 22
34 22
-- -- -- -- -- -- -- --
36
1
35 21
35 21
-- -- -- -- -- -- -- --
```

- prettyPrintFile for 8-connectness

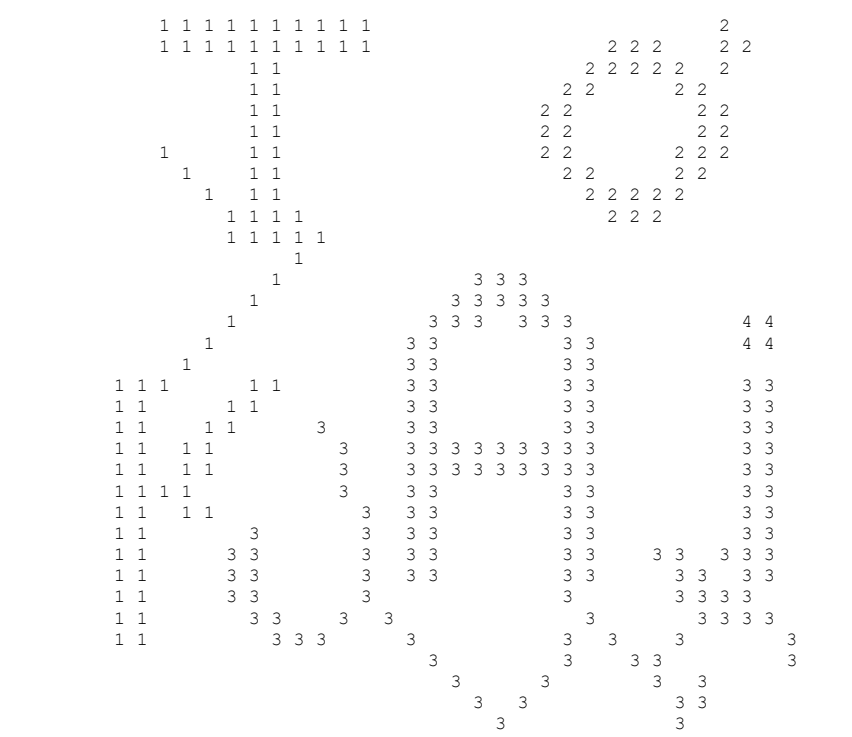




EQAry up to newLable after pass 1:
0 1 2 2 1 5 6 1 1 5 5 10 9 5



EQAry up to newLable after pass 2:
0 1 2 2 1 5 6 1 1 5 5 10 5 5
EQAry up to newLable after manageEQAry:
0 1 2 2 1 3 4 1 1 3 3 3 3 3



EQAry up to newLable after pass3:
0 1 2 2 1 3 4 1 1 3 3 3 3 3

- labelFile for 8-connectness	- propertyFile for 8-connectness
35 35 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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 2 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 2 2 2 0 0 2 2 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 2 2 2 2 2 0 2 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 2 2 0 0 0 0 2 2 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 2 2 0 0 0 0 2 2 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 2 2 0 0 0 0 0 2 2 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 0 0 0 0 2 2 2 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 0 0 0 2 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 2 2 2 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 2 2 2 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 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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 3 3 3 0 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 3 3 3 3 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 3 3 3 0 3 3 3 0 0 0 0 0 0 4 4 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 3 3 0 0 0 0 0 3 3 0 0 0 0 0 0 0 4 4 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 3 3 0 0 0 0 0 3 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 1 1 0 0 0 0 0 0 3 3 0 0 0 0 0 3 3 0 0 0 0 0 0 3 3 0 0 0 0 0 0 1 1 0 0 0 1 1 0 0 0 0 0 0 3 3 0 0 0 0 0 3 3 0 0 0 0 0 0 3 3 0 0 0 0 0 0 1 1 0 0 1 1 0 0 0 0 0 3 0 0 0 3 3 3 3 3 3 3 3 3 0 0 0 0 0 0 3 3 0 0 0 0 0 1 1 0 1 1 0 0 0 0 0 3 0 0 3 3 3 3 3 3 3 3 3 0 0 0 0 0 0 3 3 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 3 0 0 3 3 0 0 0 0 0 3 3 0 0 0 0 0 0 3 3 0 0 0 0 0 1 1 0 1 1 0 0 0 0 0 0 3 0 3 3 0 0 0 0 0 3 3 0 0 0 0 0 0 3 3 0 0 0 0 0 1 1 0 0 0 3 0 0 0 0 0 3 0 3 3 0 0 0 0 0 3 3 0 0 0 0 0 0 3 3 0 0 0 0 0 1 1 0 0 0 3 3 0 0 0 0 0 3 0 3 3 0 0 0 0 0 3 3 0 0 3 3 0 3 3 0 0 0 0 0 1 1 0 0 0 3 3 0 0 0 0 0 3 0 3 3 0 0 0 0 0 3 3 0 0 0 3 3 0 3 3 0 0 0 0 0 1 1 0 0 0 3 3 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 3 0 0 0 0 3 3 3 3 0 0 0 0 0 0 1 1 0 0 0 0 3 3 0 0 0 3 0 3 0 0 0 0 0 0 0 0 3 0 0 0 0 3 3 3 3 0 0 0 0 0 0 1 1 0 0 0 0 0 3 3 0 0 0 0 3 0 0 0 0 0 0 0 0 3 0 3 0 0 3 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 3 0 0 3 3 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 0 0 3 0 0 0 3 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 3 0 0 0 0 0 0 3 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0	35 35 0 4 4 -- -- -- -- -- -- -- -- 1 93 2 4 31 15 -- -- -- -- -- -- -- -- 2 41 2 23 11 32 -- -- -- -- -- -- -- -- 3 150 14 9 35 34 -- -- -- -- -- -- -- -- 4 4 16 32 17 33 -- -- -- -- -- -- -- --