Student: Simon Kong

Project Due date: 9/24/2021

Algorithm Steps:

Main(...)

step 0: imgFile, structFile, dilateOutFile, erodeOutFile, openingOutFile, closingOutFile, prettyPrintFile ← open

step 1: numImgRows, numImgCols, imgMin, imgMax ← read from imgFile

numStructRows, numStructCols, structMin, structMax ← read from structFile

rowOrigin, colOrigin □ read from strucFile

step 2: zeroFramedAry, structAry, morphAry, tempAry \leftarrow dynamically allocate // see description in the above

step 3: zero2DAry(zeroFramedAry, rowSize, colSize) // see description in the above

step 4: loadImg (imgFile, zeroFramedAry) // see description in the above prettyPrint (zeroFramedAry, prettyPrintFile) // write a meaningful caption before prettyPrint

step 5: zero2DAry(structAry, numStructRows, numStructCols)

loadstruct (structFile, structAry) // see description in the above
prettyPrint (structAry, prettyPrintFile) // see description in the above

step 6: zero2DAry(morphAry, rowSize, colSize)

ComputeDilation (zeroFramedAry, morphAry) // see algorithm below

AryToFile (morphAry, dilateOutFile) // see description in the above

prettyPrint (morphAry, prettyPrintFile) // write a meaningful caption before prettyPrint

step 7: zero2DAry(morphAry, rowSize, colSize)

ComputeErosion (zeroFramedAry, morphAry) // see algorithm below AryToFile (morphAry, erodeOutFile)

prettyPrint (morphAry, prettyPrintFile) // write a meaningful caption before prettyPrint

step 8: zero2DAry(morphAry, rowSize, colSize)

 $Compute Opening\ (zero Framed Ary,\ morph Ary,\ temp Ary)\ /\!/\ see\ algorithm\ below$

```
AryToFile (morphAry, openingOutFile)

prettyPrint (morphAry, prettyPrintFile) // write a meaningful caption before prettyPrint

step 9: zero2DAry(morphAry, rowSize, colSize)

ComputeClosing (zeroFramedAry, morphAry, tempAry) // see algorithm below

AryToFile (morphAry, closingOutFile)

prettyPrint (morphAry, prettyPrintFile) // write a meaningful caption before prettyPrint

step 10: close all files
```

ComputeDilation (inAry, outAry)

```
step 1: i ← rowFrameSize

step 2: j ← colFrameSize

step 3: if inAry [i,j] > 0

onePixelDilation (i, j, inAry, outAry) // only processing one pixel inAry[i,j]

step 4: j++

step 5: repeat step 3 to step 4 while j < (colSize)

step 6: i++

step 7: repeat step 2 to step 6 while i < (rowSize)
```

ComputeErosion(inAry, outAry)

onePixelErosion(i, j, inAry, outAry)

```
step 0: iOffset \square i - rowOrigin
       jOffset □ j - colOrigin
       // translation of image's coordinate (i, j) with respected of the origin of the structuring
       element
       matchFlag ← true
step 1: rIndex \leftarrow 0
step 2: cIndex \leftarrow 0
step 3: if (structAry[rIndex][cIndex] > 0) and (inAry[iOffset + rIndex][jOffset + cIndex] ) <= 0)
matchFlag ← false
step 4: cIndex ++
step 5: repeat step 3 to step 4 while (matchFlag == true) and (cIndex < numStructCols)
step 6: rIndex ++
step 7: repeat step 2 to step 6 while (matchFlag == true) and (rIndex < numStructRows)
step 8: if matchFlag == true
               outAry[i][i] \leftarrow 1
       else
               outAry[i][j] \square 0
ComputeClosing(inAry, outAry, tempAry)
step 1: ComputeDilation (zeroFramedAry, tempAry)
step 2: ComputeErosion (tempAry, morphAry)
```

ComputeOpening(inAry, outAry, tempAry)

step 1: Compute Erosion (zeroFramedAry, tempAry)

step 2: ComputeDilation (tempAry, morphAry)

onePixelDilation(inAry, outAry)

- 1) Offset x and y by -1
- 2) Replace value at offset position with structure element
- 3) Increment x
- 4) Repeat step 2 until end of row
- 5) Increment y
- 6) Repeat until end of columns

Source Code

Main Class

```
int main( int argc, const char * argv[] ) {
    if(argc != 8){
       cout << "Invalid amount of arguments";</pre>
       exit(1);
    string imgFile = argv[ 1 ] ;
    string structFile = argv[ 2 ];
    ifstream imgStream, structStream;
    imgStream.open( imgFile );
    structStream.open( structFile );
    string dilateOutFile = argv[ 3 ];
    string erodeOutFile = argv[ 4 ];
   string closingOutFile = argv[ 5 ];
    string openingOutFile = argv[ 6 ];
    string prettyPrintFile = argv[ 7 ];
   ofstream dilateStream, erodeStream, closingStream, openingStream, prettyPrintStream;
   dilateStream.open( dilateOutFile );
   erodeStream.open( erodeOutFile );
   closingStream.open( closingOutFile );
   openingStream.open( openingOutFile );
    prettyPrintStream.open( prettyPrintFile );
```

```
if( imgStream.is_open() && structStream.is_open() ){
        Morphology* morphObj = new Morphology( imgStream, structStream );
        morph0bj -> zero2DAry(morph0bj -> zeroFramedAry, morph0bj -> rowSize, morph0bj -> colSize );
        morphObj -> loadImg( imgStream, morphObj -> zeroFramedAry );
        morphObj -> prettyPrint( morphObj -> zeroFramedAry, prettyPrintStream );
        // step 5
        morph0bj -> zero2DAry( morph0bj -> structAry, morph0bj -> numStructRows, morph0bj -> numStructCols );
        morphObj -> loadStruct( structStream, morphObj -> structAry );
        morphObj -> prettyPrint( morphObj -> structAry, prettyPrintStream);
        morphObj -> zero2DAry( morphObj -> morphAry, morphObj -> rowSize, morphObj -> colSize );
        morphObj -> computeDilation( morphObj -> zeroFramedAry, morphObj -> morphAry);
        morphObj -> aryToFile( morphObj -> morphAry, dilateStream );
        morphObj -> prettyPrint( morphObj -> morphAry, prettyPrintStream);
        morphObj -> zero2DAry( morphObj -> morphAry, morphObj -> rowSize, morphObj -> colSize );
        morphObj -> computeErosion( morphObj -> zeroFramedAry, morphObj -> morphAry);
        morphObj -> aryToFile( morphObj -> morphAry, erodeStream );
        morphObj -> prettyPrint( morphObj -> morphAry, prettyPrintStream);
```

```
// step 9
    morphObj -> zero2DAry( morphObj -> morphAry, morphObj -> rowSize, morphObj -> colSize );
    morphObj -> computeClosing( morphObj -> zeroFramedAry, morphObj -> morphAry, morphObj -> tempAry);
    morphObj -> aryToFile( morphObj -> morphAry, openingStream );
    morphObj -> prettyPrint( morphObj -> morphAry, prettyPrintStream );

// step 10
    imgStream.close();
    dilateStream.close();
    dilateStream.close();
    closingStream.close();
    closingStream.close();
    openingStream.close();
    delete morphObj;
}
else{
    cout << "Error: Input" << endl;
}
return 0;</pre>
```

Morphology Class

```
class Morphology{
    public:
        int numImgRows,
            numImgCols,
            imgMin,
            imgMax,
            numStructRows,
            numStructCols,
            structMin,
            structMax,
            rowOrigin,
            colOrigin,
            rowFrameSize,
            colFrameSize,
            extraRows,
            extraCols,
            rowSize,
            colSize,
            msgCounter;
        int **zeroFramedAry, **morphAry, **tempAry, **structAry;
```

Constructor

```
public:
   Morphology( ifstream &imgFile, ifstream &structFile ){
       read img header( imgFile );
       read_struct_header( structFile );
       read_origin( structFile );
       this -> rowFrameSize = this -> numStructRows / 2;
       this -> colFrameSize = this -> numStructCols / 2;
       this -> extraRows = this -> rowFrameSize * 2;
       this -> extraCols = this -> colFrameSize * 2;
       this -> rowSize = this -> numImgRows + this -> extraRows;
       this -> colSize = this -> numImgCols + this -> extraCols;
       this -> zeroFramedAry = new int*[ rowSize ];
       this -> morphAry = new int*[ rowSize ];
       this -> tempAry = new int*[ rowSize ];
       for(int i = 0; i < this -> rowSize; i++){
           zeroFramedAry[i] = new int[ this -> colSize ];
           morphAry[i] = new int[ this -> colSize ];
           tempAry[i] = new int[ this -> colSize ];
       this -> structAry = new int*[ this -> numStructRows ];
       for(int i = 0; i < this -> numStructRows; i++){
           structAry[i] = new int[ this -> numStructCols ];
       this -> msgCounter = 0;
```

Read Headers and origin

```
void read_img_header( ifstream &inFile ){
    inFile >> this -> numImgRows >> this -> numImgCols >> this -> imgMin >> this -> imgMax ;
}

void read_struct_header( ifstream &inFile ){
    inFile >> this -> numStructRows >> this -> numStructCols >> this -> structMin >> this -> structMax ;
}

void read_origin( ifstream &inFile ){
    inFile >> this -> rowOrigin >> this -> colOrigin;
}
```

Zero2DAry

```
void zero2DAry(int **ary, int nRows, int nCols){
    for(int i = 0; i < nRows; i++){
        for(int j = 0; j < nCols; j++){
            ary[i][j] = 0;
        }
    }
}</pre>
```

loadImg

```
void loadImg( ifstream &inFile, int **ary){
   for( int i = this -> rowOrigin; i < this -> numImgRows; i++){
      for( int j = this -> colOrigin; j < this -> numImgCols; j++){
         inFile >> ary[i][j];
      }
   }
}
```

loadStruct

```
void loadStruct( ifstream &inFile, int **ary){
   for( int i = 0; i < this -> numStructRows; i++ ){
      for( int j = 0; j < this -> numStructCols; j++){
        inFile >> ary[i][j];
      }
   }
}
```

computeDilation

```
void computeDilation( int **inAry, int **outAry){
   for( int i = this -> rowFrameSize; i < this -> rowSize; i++ ){
      for(int j = this -> colFrameSize; j < this -> colSize; j++ ){
        if( inAry[i][j] > 0 ) this -> onePixelDilation(i,j, inAry, outAry);
      }
   }
}
```

computeErosion

```
void computeErosion( int **inAry, int **outAry ){
   for( int i = this -> rowFrameSize; i < this -> rowSize; i++ ){
     for(int j = this -> colFrameSize; j < this -> colSize; j++ ){
        if( inAry[i][j] > 0 ) this -> onePixelErosion(i,j, inAry, outAry);
     }
   }
}
```

computeOpening

```
void computeOpening( int **inAry, int **outAry, int **tmp){
    this -> computeErosion(inAry, tmp);
    this -> computeDilation(tmp, outAry);
}
```

computeClosing

```
void computeClosing( int **inAry, int **outAry, int **tmp){
    this -> computeDilation(inAry, tmp);
    this -> computeErosion(tmp, outAry);
}
```

onePixelDilation

onePixelErosion

```
void onePixelDilation(int i, int j, int **inAry, int **outAry){
   int structRow = 0;
   int structCol = 0;

   for( int r = i - 1; r <= i + 1; r++ ){
      for( int c = j - 1; c <= j + 1; c++ ){
        outAry[r][c] = this -> structAry[ structRow ][ structCol ];
        structCol++;
      }
      structCol = 0;
      structRow++;
   }
}
```

aryToFile

```
void aryToFile(int **ary, ofstream &outFile ){
  outFile << this -> numImgRows << " " << this -> numImgCols << " " << this -> imgMin << " " << this -> imgMax << endl;
  for( int i = this -> rowFrameSize; i < this -> rowSize; i++ ){
     for( int j = this -> colFrameSize; j < this -> colSize; j++ ){
      outFile << ary[i][j] << " ";
     }
     outFile << endl;
}
</pre>
```

prettyPrint

Test 1

dilateOutFile

35 60 0 1 $0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 1\ 1\ 0\ 0$ $0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0$

erodeOutFile



openingOutFile

35 60 0 1	
$1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ $	00000000000
$\begin{smallmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 $	00000000000
$\begin{smallmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 &$	00000100000
$\begin{smallmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 $	00000010000
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	10000000000
0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0	01000000000
0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0	00100000000
0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0	00010100000
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0	00001010000
001000000000000100000000000000000000000	
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1	
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 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 \ 0 \ 0 \ 1 \ 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 \ 1 \ 0 \ 0 \ $	
0 0 0 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0	
0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0	
000000000000000000000000000000000000000	
000010000000001010000100000000000000000	
$\begin{smallmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 &$	
$\begin{smallmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 $	
$\begin{smallmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 &$	
$\begin{smallmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 $	
$\begin{smallmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0$	
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	
$ \begin{smallmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0$	0000000000

closingOutFile

be																																																					
35																																																					
1 (0	0
					0 6																																															0	0
1 :	1 (9 0	0	0	0 6	0	0	0	0	0	0	0	0	0	0	0 6	0	0	0	0	0	0	0	0 (9 6	9 6	9 6	9 0	0	0	0	0	0	0 (9 0	0	0	0	0 (9 6	0	0	0	0	0	0	0 (0 e) 0	0	0	0	0
1 :	1 (9 0	0	0	0 6	0	0	0	0	0	0	0	0	0	0	0 6	0	0	0	0	0	0	0 (0 (9 6	9 6	9 6	9 0	0	0	0	0	1 (0 :	10	0	0	0	0 (9 6	0	0	0	0	0	0	0 (0 0	0	0	0	0	0
0 :	1 (9 0	0	0	0 6	0	0	0	0	0	0	0	0	0	0	0 6	0	0	0	0	0	0	0 (0 (9 6	9 6	9 6	9 0	0	0	0	0	0	1 (ð 1	0	0	0	0 (9 6	0	0	0	0	0	0	0 (0 e	0	0	0	0	0
0 (9 6	9 0	0	0	0 6	0	1	0	0	0	0	0	0	0	0	0 6	0	0	0	0	0	0	0	1 (9 6	9 6	9 6	9 0	0	0	0	0	0 (0 1	1 0	1	0	0	0 (9 6	0	0	0	0	0	0	0 (0 e	0	0	0	0	0
0 (9 6	9 0	0	0	0 6	0	0	1	0	1	0	0	0	0	0	0 6	0	0	1	0	0	0	0 (0 :	1 6	9 6	9 6	9 0	0	0	0	0	0 (0 (a 1	0	1	0	0 (9 6	0	0	0	0	0	1	0 (0 e	0	0	0	0	0
0 (9 6	9 0	0	0	0 6	0	0	0	1	0	1	0	0	0	0	0 6	0	0	0	1	0	0	0 (0 (a 1	1 6	9 6	9 0	0	0	0	0	0 (0 (9 0	1	0	0	0 (9 6	0	0	0	0	0	0	1 (0 1	0	0	0	0	0
0 (9 6	9 0	0	0	0 6	0	0	0	0	0	0	1	0	0	0	0 6	0	0	0	0	1	0	1 (0 (9 6	9 6	9 6	9 0	0	0	0	0	0 (0 (9 0	0	1	0	0 (9 6	0	0	0	0	0	0	0 :	1 0	1	0	0	0	0
					0 6																																																0
					0 6																																																0
					0 6																																																0
					0 6																																																0
					0 6																																																0
					0 6																																																0
					0 6																																																0
0 (a (9 0	0	0	0 6	0	0	0	0	0	0	0	0	0	0	0 6	0	0	0	0	0	0	0 (0 (9 6	9 6	9 6	9 0	0	0	0	0	0 (0 (9 0	0	0	0	0 (9 6	0	0	0	0	0	0	0 (0 e	0	0	0	0	0
0 (9 6	9 0	0	0	0 6	0	0	0	0	0	0	0	0	0	0	0 6	0	0	0	0	0	0	0 (0 (9 6	9 6	9 6	9 0	0	0	0	0	0 (0 (9 0	0	0	0	0 (9 6	0	0	0	0	0	0	0 (0 e	0	0	0	0	0
					0 6																																																0
					0 6																																															0	0
0 (a (9 0	0	0	0 6	9 0	0	0	0	0	0	0	0	0	0	0 6	0	0	0	0	0	0	0 (0 (9 6	9 6	9 6	9 0	0	0	0	0	0 (0 (9 0	0	0	0	0 (9 6	0	0	0	0	0	0	0 (0 e	0	0	0	0	0
0 (9 (9 0	0	0	0 6	0	0	0	0	0	0	0	0	0	0	0 6	0	0	0	0	0	0	0 (0 (9 6	9 6	9 6	9 0	0	0	0	0	0 (0 :	1 0	0	0	0	0 (9 6	0	0	0	0	0	0	0 (0 1	0	0	0	0	0
0 (9 (9 0	0	0	0 6	0	0	0	0	0	0	0	0	0	0	0 6	0	0	0	0	0	0	0 (0 (9 6	9 6	9 6	9 0	0	0	0	0	0 (0 (a 1	0	0	0	0 (9 6	0	0	0	0	0	0	0 (0 e	1	0	0	0	0
0 (9 (9 0	0	0	0 6	0	0	0	0	0	0	1	0	0	0	0 6	0	0	0	0	0	0	0 (0 (9 6	9 6	9 6	9 0	0	0	0	0	0 (0 (9 0	1	0	0	0 :	1 0	0	0	0	0	0	0	0 (0 e	0	1	0	0	0
0 (9 (9 0	0	0	0 6	0	0	0	0	0	0	1	1	0	0	0 6	0	0	0	0	0	0	0 (0 (9 6	9 1	1 0	9 0	0	0	0	0	0 (0 (ð 1	0	0	0	0 (9 1	0	0	0	0	0	0	0 (0 e	0	0	1	0	0
0 (9 (9 0	0	0	0 6	0	0	0	0	0	0	0	1	1	0	0 6	0	0	0	0	0	0	0 (0 (9 6	9 6) 1	L 0	0	0	0	0	0 (0 (9 0	1	0	0	0 (9 6	1	0	0	0	0	0	0 (0 0	0	0	0	0	0
1 (9 (9 0	0	0	0 6	0	0	0	0	0	0	1	0	1	1	0 6	0	0	0	0	0	0	0 (0 (9 6	9 6	9 6	1	0	0	0	0	0 (0 (9 0	0	1	0	1 (9 6	0	0	0	0	0	0	0 (0 0	0	0	0	0	0
0 :	1 (9 0	0	0	0 6	0	0	0	0	0	0	0	1	0	1	1 6	1	0	0	0	0	0	0 (0 (9 6	9 6	9 6	9 0	1	0	0	0	0	0 (9 0	0	0	0	0 :	1 0	0	0	0	1	0	0	0 (0 0	0	0	0	0	0
0 (0 1	1 0	0	0	0 6	0	0	0	0	0	0	1	0	1	0	0 1	0	1	0	0	0	0	0 (0 (9 6	9 6	9 6	9 0	0	1	0	0	0 (0 (9 0	0	0	0	1 (9 1	0	0	0	0	1	0	0 (0 0	0	0	0	0	0
0 (9 (9 0	0	0	0 6	0	0	0	0	0	0	0	1	0	1	0 6	0	0	1	0	0	0	0 (0 (9 6	9 6	9 6	9 0	0	0	1	0	0	0 (9 0	0	0	0	0 :	1 0	1	0	0	0	0	1	0 (0 0	0	0	0	0	0
0 (9 (9 0	0	0	0 6	0	0	0	0	0	0	0	0	1	0	1 6	0	0	0	1	0	0	0 (0 (9 6	9 6	9 6	9 0	0	0	0	0	0	0 (9 0	0	0	0	0 (9 1	0	1	0	0	0	0	0 (0 e	0	0	0	0	0
0 (a (9 0	0	0	0 6	0	0	0	0	0	0	0	0	0	1	0 6	0	0	0	0	0	0	0 (0 (0 6	9 6	9 6	9 0	0	0	0	0	0 (0 (9 0	0	0	0	0 (9 e	0	0	1	0	0	0	0 (0 e	0	0	0	0	0
0 (9 6	9 0	0	0	0 6	0	0	0	0	0	0	0	0	0	0	0 6	0	0	0	0	0	0	0 (0 (9 6	9 6	9 6	9 0	0	0	0	0	0 (0 (9 0	0	0	0	0 (9 6	0	0	0	0	0	0	0 (0 e	0	0	0	0	0
0 (9 6	9 0	0	0	0 6	0	0	0	0	0	0	0	0	0	0	0 6	0	0	0	0	0	0	0	0 (9 6	9 6	9 6	9 0	0	0	0	0	0 (0 (9 0	0	0	0	0 (9 6	0	0	0	0	0	0	0 (0 e	0	0	0	0	0
0 (9 6	9 0	0	0	0 6	0	0	0	0	0	0	0	0	0	0	0 6	0	0	0	0	0	0	0	0 (9 6	9 6	9 6	9 0	0	0	0	0	0 (0 (9 0	0	0	0	0 (9 6	0	0	0	0	0	0	0 (0 e	0	0	0	0	0
0 (9 (9 0	0	0	0 6	0	0	0	0	0	0	0	0	0	0	0 6	0	0	0	0	0	0	0 (0 (9 6	9 6	9 6	9 0	0	0	0	0	0 (0 (9 0	0	0	0	0 (9 6	0	0	0	0	0	0	0 (0 e	0	0	0	0	0

prettyPrintFile

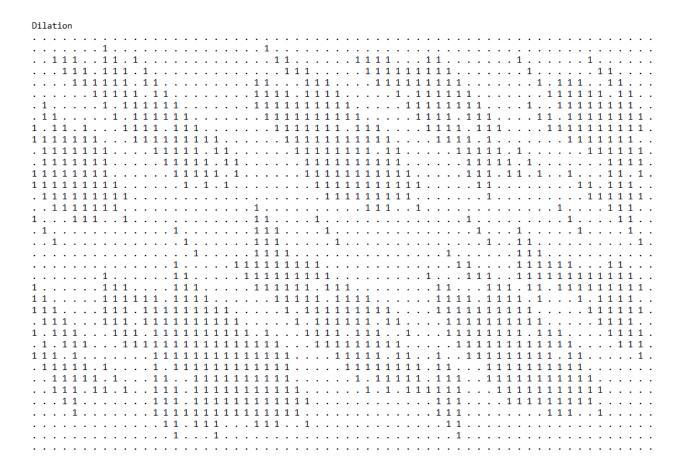
Zero Framed	
20.0 Trained	
1	
1 1 1 1 1	1
1 . 1 1	. 1 1 1
	11111
	. 1 1 1 . 1 1 . 1
1 1 . 1 . 1	1.111111.1.1.
1 1 . 1 1 . 1 1	11
. 1 . 1 1 . 1 1 1	
. 1 . 1 1 1 1 1 1 1 1 1 1 1 . 1 . 1 . 1	. 1 1 1 . 1 1 1 1 1
1 . 1 . 1 1 . 1 1 . 1 1 . 1 . 1 . 1 1	
1 . 1 1 1 1 . 1 . 1	
. 1 1 . 1 . 1 1 1 1	
. 1 1 1	
1 1 1 1	
. 1	
.11.11.1	
.111111	
. 1 1 11 . 11 . 1 . 1	
1 1 1 1 1 1 1 1 1	
. 1 1 . 1 . 1 1 1 1 1	
1 1 . 11 . 11 . 1 11 . 11 . 11	
. 1 1 1 . 1 1 1 1 1	
	11.11.1.111
	11111.111

Structure Element

1 . .

. 1 .

. . 1



Opening	
1	
11	
111	
111	
. 1 1	
1	
1	
1	
	1
	. 1
1	

Erosion	
	1 . 1
	1 . 1
1	1
1	1
	1
	1
1	
	1
	1

Closing	
. 1	
1 1	
1 1 1	1
11 . 11 1	. 1 1 1
1 . 1 1 1 . 1 1	
1 1 1 . 1	
. 1 . 1 1 . 1 1 1 1 . 1 . 1 .	
1 1 1 1 1 1 1 1 1 1 1	
1 1 1 1 1 1 1 . 1 .	
. 1 1 1 1 1 1 1 1	
. 1 1 1 1 1 1 1	
1 1 1 1	
	1 1
. 1	1
	L
	11
	111
	. 1 1 1 1 1 1
. 1 1 1 1 1 1 1 1 1 1 1	
1 1 1 1 1 1 1 1 1 1 1 1 1 1	
. 1 . 1 1 . 1 1 1 1 1 1 1	
1 . 1	
. 1 1 1 . 1 1 1 1 1	
1 1 . 1 1 . 1 1 1 1 1 1 1 1 1 1 1 1 1 . 1 . 1 1 1 1 1 1 1	
1	
1	1111

Test 2

closingOutfile



dilateOutFile

B5 60 0 1	
$\begin{smallmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 &$	9999999999
0 0 0 1 1 1 0 1 1 0 0 1 0 0 0 0 0 0 0 0	
001110110110100000000000000000000000000	
011101111101100000000000011001101111101111	
000010110111100000000000011001111111111	
001001011001100000000011101111111111110000	
011001001111111111111111111111111111111	
$1 \; 1 \; 0 \; 1 \; 1 \; 0 \; 1 \; 0 \; 1 \; 1 \; $	
$\begin{smallmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 &$	
111011100101111111111100010111111111110000	
$\begin{smallmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 &$	
$1 \; 1 \; 1 \; 1 \; 1 \; 1 \; 1 \; 1 \; 0 \; 0 \; $	
$1 \; 1 \; 1 \; 1 \; 1 \; 1 \; 1 \; 1 \; 0 \; 0 \; $	
$1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 1 \ 0 \ 0$	
$0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ $	
$0 \; 1 \; 1 \; 1 \; 0 \; 0 \; 1 \; 0 \; 0 \; 0 \; $	
$1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ $	
$\begin{smallmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 $	
$\begin{smallmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 $	00000000000
$\begin{smallmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 $	01111000110
$\begin{smallmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 &$	11110011100
$\begin{smallmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0$	11101111110
$1 \; 1 \; 0 \; 0 \; 0 \; 0 \; 1 \; 1 \; 1 \; 1 \; $	00011111100
$1 \; 1 \; 0 \; 0 \; 0 \; 0 \; 1 \; 1 \; 1 \; 0 \; 0$	00101111010
$1 \; 0 \; 1 \; 1 \; 1 \; 0 \; 1 \; 1 \; 1 \; 1 \; $	11010011110
$\begin{smallmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1$	10010111110
1 1 1 0 0 0 1 1 0 0 1 1 1 1 1 1 1 1 1 1	01110101100
011101000101111111111011100101111111111	11100001000
11111110100101111111111110001101111111001101111	11011100000
1 1 1 1 1 1 0 1 0 0 0 0 0 1 1 1 1 1 1 1	11111000000
$\begin{smallmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 &$	
011000000001110111101111000000000001011100111001	
$\begin{smallmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1$	
$\begin{smallmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 &$	
$0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ $	
	00000000

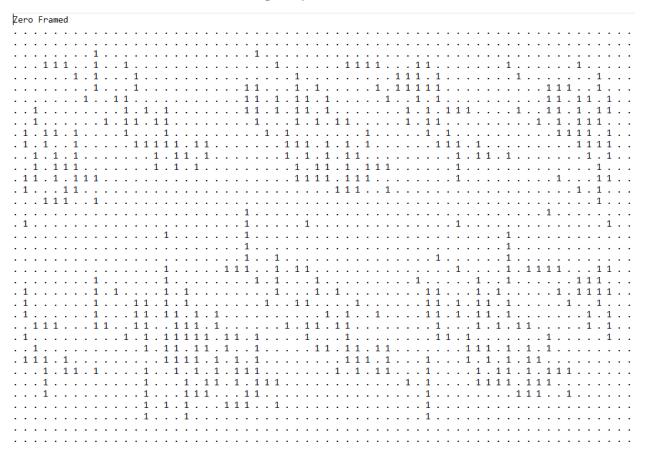
erodeOutFile



openingOutFile

B5 60 0 1	
$\begin{smallmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 $	
$\begin{smallmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 &$	
$\begin{smallmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 &$	
$\begin{smallmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0$	
$\begin{smallmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1$	
$\begin{smallmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 &$	000000110110100
$\begin{smallmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0$	000100111101100
$\begin{smallmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1$	000001011111000
$\begin{smallmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1$	000010011110100
$1 \; 0 \; 1 \; 0 \; 1 \; 1 \; 1 \; 1 \; 0 \; 0 \; $	000100000111100
$\begin{smallmatrix} 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 &$	101000000110100
$\begin{smallmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 &$	000000001001000
$1 \; 1 \; 0 \; 1 \; 1 \; 1 \; 1 \; 1 \; 0 \; 0 \; $	000000010011100
$\begin{smallmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 &$	000000000101000
$\begin{smallmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 &$	000000000001000
$\begin{smallmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 &$	000000100000000
100000000000000000000000000000000000000	0000000000000100
$\begin{smallmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 $	0010000000000000
$\begin{smallmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 $	0010000000000000
$\begin{smallmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 $	0110000000000000
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0	111011110001100
0 0 0 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0	111010000111000
100000010100111010000000010011010000000	110100010111100
1000000110011111100000001001101001000000	101000101001000
1000000100011111010100010010101001000001111	101001000010100
0111000110011111111110010010111011010000	010110000010100
1010001000101111111011010101011111000000	101100100000100
010001000100101111111101110000110110110	1110111000000000
111010001000011111111011100000000111010000	
0 0 1 1 1 1 0 1 0 0 0 0 0 1 1 1 1 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 1 0 1 1 0 0 1 1 0 0 0 0 1	
0 0 1 1 0 0 0 0 0 0 0 1 1 1 0 1 1 1 1 1	
$\begin{smallmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 &$	
$\begin{smallmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1$	
$\begin{smallmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 $	
$\begin{smallmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 $	

prettyPrintFile



Structure Element Dilation

Erosion	
	. 1
	1
	. 1 1
	. 1
1	1
1	
11	
	. 1 1
	1
11.1	
,	

11 11 <td< th=""></td<>
$\begin{array}{cccccccccccccccccccccccccccccccccccc$
$\begin{array}{cccccccccccccccccccccccccccccccccccc$
$\begin{array}{cccccccccccccccccccccccccccccccccccc$
1
$\begin{array}{c} \dots \dots$
$\begin{array}{c} \dots \dots$
$\begin{array}{c} \dots & \dots $
$\begin{array}{c} \dots & \dots $
1
1 1
1
1 . 1 1 1
11
11

Closing

Test 3

closingOutFile



dilateOutFile

35	60	0 1																																												
0 0	0	0 0	0	0 1	. 0	0	0 6	9 6	0	0	0 0	9 0	0 (9 e	0	0	1 6	0	0	0	1	1	0 (9 0	0	0	0 (9 e	0	0	0 (9 0	0	0	0 e	0	0	0 0	0	0	0 (0 0	0	0 (9 0	0
0 0	1	1 1	0	1 1	1	1	0 6	9 6	0	0	0 0	9 0	0 (9 6	0	1	1 1	. 1	0	1	1	1	1 (a 1	1	1	1 (9 e	0	1	1 (9 0	0	0	0 e	0	1	0 0	0	0	0 (0 1	0	0 (9 0	0
0 1	1	1 1	1	1 1	1	1	1 1	1 0	0	0	0 0	9 0	0 (9 e	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 e	1	1	1 6	0	0	0 :	1 1	1	1 (9 0	0
0 0	1	1 1	1	1 1	1	1	1 1	1 1	1	0	0 1	1 1	0 (9 e	0	1	1 1	1	1	1	1	1	1 (a 1	1	1	1	1 1	1	1	1 :	1 1	0	0	0 0	0	1	1 1	0	0	0 (0 1	1	1 :	1 0	0
0 0																																														
0 1					_				_				_		_				_										_												_					
1 1					_				_						_				_			_			_				_				_	_		_										_
1 1	_		_		_	_			_	_					_				_	_	_	_			_	_			_	_			_	_		_	_		_	_			_			_
1 1																																														
1 1																																														
1 1																																														
1 1	_		_		_	_			_	_					_	_			_	_	_	_	_		_	_			_	_			_	_		_	_		_	_			_			_
1 1	_		_		_	_			_	_					_	_		_	_	_	_	_			_	_			_	_	_		_	_		_	_		_	_	_		_	_		_
1 1																																														
1 1																																														
1 1			_		_	_			_	_					_				_	_	_	_			_								_			_	_				_		_			
1 1																																														
11																																														
11																																														
11																																														
11																																														
11																																														
11																																														
11																																														
11																																														
11																																														
1 1																																														
1 1																																														
1 1																																													שנו	0
1 1																																													0 0	0
1 1					_				_						_				_						_				_		_					_			_						0 0	0
0 1	_		_		_				_	_			_		_	_			_			_			_	_	_		_	_	_		_				_		_		_		_		0 0	0
0 0																																									0 :	10	0	0 (9 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	00	0	0	0 (9 6	0	0	0 0	0	0 (9 6	0	0	0 6	0	0	0	0	0	0 (9 0	0	0	0 (0 0	0	0	0 (9 0	0	0	0 e	0	0	0 6	0	0	0 (0 0	0	0 (0 6	0

erodeOutFile



openingOutFile

β5 60 0 1
$ \verb 0 $
$\begin{smallmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 $
$1 \; 0 \; 1 \; 1 \; 1 \; 1 \; 1 \; 1 \; 1 \; 0 \; 0$
$0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\$
$0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\$
0 0 0 0 0 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1
$0\;1\;0\;0\;0\;0\;0\;0\;1\;1\;1\;1\;1\;1\;1\;1\;0\;0\;0\;0$
$1\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\$
$0\ 1\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\$
$0\ 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 1 0 0 0 0 0 0 1 1 1 1 1 1 1 1
1 1 1 0 0 0 1 0 0 0 0 0 0 1 0 1 0 1 0 1
1 1 1 1 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
$1\;1\;1\;1\;1\;1\;1\;0\;0\;0\;0\;0\;0\;0\;0\;0\;0\;0\;0\;0$
$0\ 1\ 1\ 1\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\$
$0\;1\;0\;0\;0\;0\;0\;1\;0\;0\;0\;0\;0\;0\;0\;0\;0\;0\;0\;0$
$1 \; 0 \; 0 \; 0 \; 0 \; 0 \; 1 \; 1 \; 0 \; 0 \; $
$\begin{smallmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0$
$0\;1\;0\;0\;0\;0\;0\;0\;0\;0\;1\;0\;1\;1\;1\;0\;0\;0\;0\;0$
$0\;1\;0\;0\;0\;0\;0\;0\;0\;0\;0\;1\;1\;1\;1\;1\;0\;0\;0\;0$
$0 \; 1 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; $
$0\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\$
$1 \; 1 \; 0 \; 0 \; 0 \; 0 \; 0 \; 1 \; 1 \; 1 \; $
$\begin{smallmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1$
$1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 1 \ 1$
$\begin{smallmatrix} 0&1&0&0&0&0&0&1&1&0&1&1&1&1&1&1&1&1&1&1$
$1 \; 1 \; 1 \; 0 \; 0 \; 0 \; 0 \; 0 \; 1 \; 1 \; $
$1 \; 1 \; 1 \; 1 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; $
$1 \; 1 \; 1 \; 1 \; 1 \; 1 \; 0 \; 0 \; 0 \; 0 \; $
$1 \; 1 \; 1 \; 1 \; 1 \; 1 \; 1 \; 1 \; 0 \; 0 \; $
$0\;1\;1\;1\;1\;1\;1\;1\;1\;1\;1\;1\;0\;0\;0\;1\;1\;0\;1\;1\;1\;1\;1\;1\;1\;0\;0\;0\;0\;0\;0\;1\;1\;1\;1\;1\;1\;1\;1\;1\;1\;1\;1\;0\;0\;0\;0\;1\;1\;1\;1\;1\;0\;0\;0\;0\;0\;0\;0$
$\begin{smallmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 &$
$0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 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\ 1\ 1\ 0\ 0\ 1\ 1\ 0\ 1\ 1\ 1\ 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\$
$ \verb 0 0 0 0 0 0 0 $
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

prettyPrintFile

Zero Framed	
1	
1 1 1 1 1 1	
1 . 1 1 1 1 1 1	
.11.111111.111111.111.111	
1 1 111111 . 11 1 . 11 . 11 . 1 11 . 1111	
. 1 1 . 1 . 1 1 . 1 1 . 1 . 1	
. 1 1 1 1 . 1 . 1	
. 1 1 1 1 . 1 1 1	•
. 1 1 1 1 . 1	
1 1 1 1	•
. 1 1	
	•
1 1	
1	
1	
1 1	
. 1 1 1 . 1	•
	•
. 1 1 1 1 1 1 1 . 1 .	•
1 11 11111111 . 1	•
. 1 1 1 1 1 . 1 . 1 1 1 1 1	•
. 1 1 1 1 1 1 1 1 1 1 1	•
. 1 1 1 1 . 1	•
. 1 1 1 1 1 1 1 1 1 1 . 1	•
	•
1111111.1.1.1.11111.11.	•
	•
	•
	•
Structure Element	•
. 1 .	
. 1 . 1 1 1	
.1.	
• • •	

Dilation . 1 1 1 1 1 1 1 1 1 1 1 1 1 1 . 1 . 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Erosion	
1 1	
1	
1 1	
1 1	
1	
1 1	
1 1 1	
11	
11	

Opening	
. 1	
111	
111	
11111	
1111111	
1	
. 1 1 1	
. 1 1 1 1	
. 1 1 1 1	
1 1	
	111
	1111
	11111
	11.1
11	
11	
1	
. 1 1 1	
.1111.1	
. 1 1 1 1 1 1 1	
1 1 . 1 1 1 1 1	
11111	
1 1	

	osi	_																																											
					. 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	ι.	
					. 1	1	1 1	. 1	1	1 1	l.		1 1			1	1	1 1	1	. 1	1	1				. :	1 1	1 1	1	1 1	1												. 1	١.	
					. 1	. :	1 1	1	1	1 1	1 1	1	1 1	. 1		1	1	1 1	1	. 1	1	1	1.				. 1	1 1	1	1 1	1								. 1	1 1			. 1	. 1	
	. 1						. 1	1	1	1 1	1 1	1	1.			1	1	1 1	1	. 1	1	1	1.					. 1	1	1 1	1	1				. 1			1 1	1 1	. 1		1 1	ι.	
	1 1						1 1	1	1	1 1	1 1	1					1	1 1	1	1	1	1	1 1	ι.				. 1	1	1 1	1	1	1.					1	. 1	1 1	1	1	1 1	1	
	. 1	1		1				1	1	1 1	1 1	1	1.				. :	1 1	1	. 1	1	1	1 1	1 1	1					1 1	1	1	1 1	1						. 1	. 1	1	1 1	1 1	
	. 1	1	1	1					1	1 1	1 1	1	1 1	. 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	Ĺ.	
	1 1	1	1	. :	1.					1 1	1 1	1	1 1	1	1.		1	1 1	1	1	1	1	1 1	1 1	1	1 :	1 1	1.		1 1	1	1	1 1	1	1 1	۱ 1						1	1 1	1 1	
. :	1 1	1			. 1					. 1	ι.	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	ι.							1 1	1 1	
					1 1																																								
					1 1																																								
					. 1																																								
					: :																																								
					: :																																								
					: :																																								
					: :																																								
					1.																																								
					1 1																																								
					1 1																																								
					1 1																																								
					1 1																																								
																					٠								٠													٠			

Test 4

closingOutFile



dilateOutFile

85 60 0 1	
0.111111111100000000000011111111111111	111000000
1111111111100000000000111111111111111001110000	
111111111111111111111111111111111111111	
1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
110011111111111111111111111111111111111	
111110111111111111111111111111111111111	
111110111111111111111111111111111111111	
111111001111111111111111111111111111111	
111111000111111111111111111111111111111	
111111110001111111111111111111111111111	
111111110001111111001111111111111111111	
111111110000000000000111111111111111111	
111111110000000000001111111001111111111	
111101110000000000001111111111100000001111	
0000011111000111000011101111111111000001111	
1100011111111111000011100001111111110001111	
110000011111111110001110111111111110001111	
110000001111111110011111111111111111111	
110000110011111111111111111111111111111	
110001111111111111111111111111111111111	
110001111111111110000111111111111111111	111110000
$1 \; 1 \; 0 \; 0 \; 0 \; 1 \; 1 \; 1 \; 1 \; 1 \; $	11111000
$1 \; 1 \; 0 \; 0 \; 0 \; 1 \; 1 \; 1 \; 1 \; 1 \; $	11111000
110001111111111111111111111111111111111	11111000
11110111111111111111	11111000
111111111111111111111111111111111111111	100111000
111111110011111111111111111111111111111	100111000
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	00000000
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	110000000
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	110000000
$\begin{smallmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1$	10000000
0 0 0 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1	00000000
$\begin{smallmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 &$	00000000
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	00000000
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	00000000
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	00000000

erodeOutFile



openingOutFile

B5 60 0 1	
$\begin{smallmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 &$	0000
$\begin{smallmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 &$	0000
$\begin{smallmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 &$	0000
0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 1 1 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0 1 1 1 1 1 1 1 0	0000
0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0 0	0000
0000001111111000000011111111100000011111	0000
1000000011111100000000111111110000011111	0000
1110000001111111000000001111111111001111	0000
111100000001111111000001111111111111111	0000
111100000000111111000001111111111111111	0000
1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0000
1111110000000000000001111100001111100000	0000
$1 \; 1 \; 1 \; 0 \; 0 \; 1 \; 0 \; 0 \; 0 \; 0 \; $	0000
$\begin{smallmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 $	0000
$\begin{smallmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 $	0000
$\begin{smallmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 $	0000
$\begin{smallmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 $	0000
$\begin{smallmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 &$	0000
$\begin{smallmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1$	0000
$\begin{smallmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1$	0000
$\begin{smallmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 &$	0000
$\begin{smallmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 &$	0000
$\begin{smallmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 &$	0000
$\begin{smallmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 &$	0000
$\begin{smallmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 &$	
$1\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\$	0000
11100100000111111111	0000
11111100000111111111	0000
1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1	
$\begin{smallmatrix} 0&1&1&1&1&1&1&0&0&0&1&1&1&1&1&1&1&1&1&1$	
$\begin{smallmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 &$	0000
$\begin{smallmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 &$	0000
$ \ \ 000000000000$	
$ \ \ 000000000000$	
	0000
000000000000000000000000000000000000000	0000

prettyPrintfile

Zero Framed	
1 1 1 1 1 1	1
1 . 1 1 1 1 1 1	1
	1
1 . 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
. 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	
111	. 1 1 1
. 1 1 1 1 . 1 1 1	1
. 1 1 1 1 . 1	
1 1 1 1	
.1111.11111	
1 1 11	
	1 1
1 1 1	
. 1 1 1 . 1 1 . 1 1 1 1 1 1 1 1 1 1 .	
	1
. 1 1 1 1 1 1 1 1 . 1	. 1 . 1
1 1 1 1 1 1 1 1 1 1 1	. 1 . 1
. 1 1 1 1 1 . 1 . 1 1 1 1 1	1
. 1 1 1 1 1 1 1 1 1 1 1	1
. 1 1 1 1 . 1 1 1 1 1 1 . 1 . 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 .	
. 1 1 1 1 1 1 1 1 1 1 . 1 1 1 1 . 1 1 1 1 1 1 1 1 1 1 1 . 1 . 1 1 1 1 1 .	
1 1 . 1 1 1 1 1 . 1 . 1 1 1 . 1 . 1 . 1 1 . 1 1 1 1 1 1 1 . 1 1 1 1 1 1	
1 1 1 1 1 1 1 1 1 . 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	

Structure Element 1111 1 1 1 1 . 11. Dilation11111.1111111111111111111111111111111...

Erosion																																				
		 						-	-	-		-					-	-	-	-	-	-		-	-			-			-					
																																				•
	-	 _	-	-	-	-		_	-	-	_	-		-			-		-			-	-	-	-	-		-	-	-	_	-	-			
	-	 	-			-	-			-		-		-		-					-	-			-	-		-		-		-		-		
	-	 				-	-			-		-		-		-		-			-	-			-		_	-		-		-		-		
					_																															
																						_														
																																_				
1 .																																				
		 			-					-			-				-		-	-					-			-			-		-			
		 			-				-	-	-		_		-		-		-		-		-	-				-	-		-		-			
		 			-				-	-	-		_		-		-		-		-		-	-				-	-	-	_		-	-		
	-	 			-	-			-	-	-		_				-		-		-		-	-				-	-	-	_		-	-		
	-	 	-		-	-			-	-	-		_				-		-				-	-				-	-	-	_		-	-		
	-	 	-	-	-	-		_	-	-	_	-		-			-		-			-	-	-	-	-	_	-	-	-	_	-	-	-		
	-	 	-	-	-	-			-	-	_	-		-			-		-			-	-	_	-	-		-	-	-	_	-		-		
1 .		 						-													-			•												
		 						-													-			•									-			
																																				•
																																				•
																																				•
																																				•
		 						-		-			-		-		-		-									-			-					٠
		 						-					•		•		•																			٠
								-					•				•																		٠.	٠
																																				٠

Op(
					. 1																											
					. 1																											
								 				 															. :	1 1	. 1			
	. 1	1						 				 																				
. :	1 1	1						 				 																				
								 				 										. 1	1									
								 				 									. :	1 1	1									
								 				 									. :	1 1	1									
								 				 												. :	1 1							
	. 1	1						 				 												1 :	1 1							
. :	1 1	1						 				 																				
. :	1 1	1		1	1.			 				 																				
			1	1	1.							 																				
			1	1	1.			 				 																				

		in																																															
																																																	•
						1																																											
						1	1 1	1 1	1	1	1 1			1	1.				1 1	۱ 1	1	1	1 1	۱ 1					1	1 :	1 1	. 1	1 .												. 1				
							1 1	1 1	1	1	1 1	1	1						1 1	۱ 1	1	1	1 1	۱ 1						1 :	1 1	1	1 .								. 1	1 1			1 1				
							. 1	1 1	1	1	1 1	1							1 1	۱ 1	1	1	1 1	۱ 1	1					. :	1 1	1	1 1	١.							1 1	1 1	1	1	1 1				
		1						. 1	1	1	1 1	1							. 1	۱ 1	1	1	1 1	۱ 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	۱ 1	1	1	1 1			1 :	1 1	1	1 1	1	1	1 1							1	1	1 1				
		1	1	1 1	ι.						1 1	1	1	1	1 1	ι.				. 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	١.			
																																																: :	
																																																: :	
																																																: :	
																																									. 1							٠.	
																																																٠.	•
																																																: :	•
																																	1 .																
								1 1																									1 .										_	_			-		_
								1 1																																									
		1					1 .			1	1 1	. 1	1	1	1 1	1	1	1	1 .				. 1	۱ 1	1	1	1 1	. 1	1						1	1 1	1	1 1	. 1	1	1 .								
		1	1	1.		1					1 1	. 1	1	1	1 1	1 1	1	1	1 .						1	1	1 1	. 1	1							1 1	1	1 1	1	1	1 1	1.							
		1	1	1 1	1	1					1 1	1	1	1	1 1	1 1	1	1	1 1	ι.						1	1 1	1	1	1						1 1	1	1 1	1	1	1 1	1.							
		1	1	1 1	1	1	1 .			1	1 1	1	1	1	1 1	1 1	1	1	1 1	۱ 1							1 1	. 1	1	1 :	1 1	. 1	1 .		1	1 1		. 1	1	1	1 1	1.							
			1	1 1	1	1	1.			1	1 1	1	1	1	1 1	1 1	1	1	1 1	۱ 1							. 1	1	1	1 :	1 1	1	1 1	1	1	1.			1	1	1 1	1 1							
				. 1	1	1					1 1	1	1	1	1 1	1 1	1	1	1 1	۱ 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.													
																																																: :	
																																																: :	
-	-	-	-						-				-		-		-	-			-				-	•			-	•			-		-		-			-			_	-			-	: :	-
																																																: :	