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 🡨 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)

ComputeOpening (zeroFramedAry, morphAry, tempAry) // 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)

step 1: i 🡨 rowFrameSize

step 2: j 🡨 colFrameSize

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

onePixelErosion (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)

onePixelErosion(i, j, inAry, outAry)

step 0 : iOffset  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 🡨 0

step 2: cIndex 🡨 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][j] 🡨 1

else

outAry[i][j]  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)

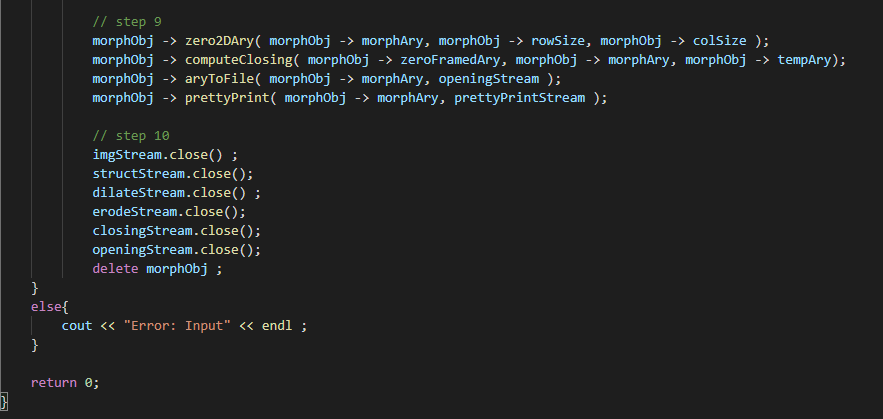
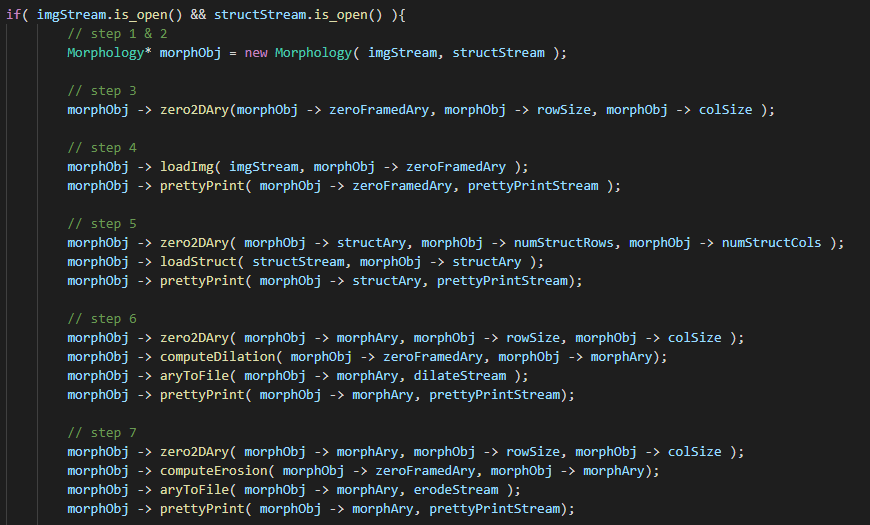
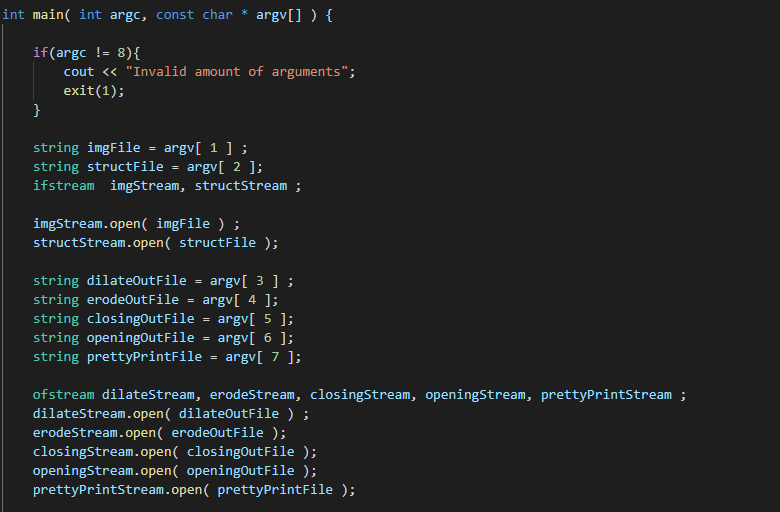
step 2: ComputeDilation (tempAry, morphAry)

onePixelDilation(inAry, outAry)

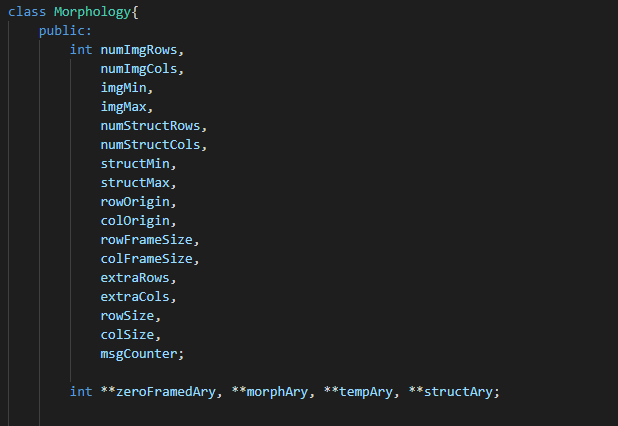
1. Offset x and y by -1
2. Replace value at offset position with structure element
3. Increment x
4. Repeat step 2 until end of row
5. Increment y
6. Repeat until end of columns

Source Code

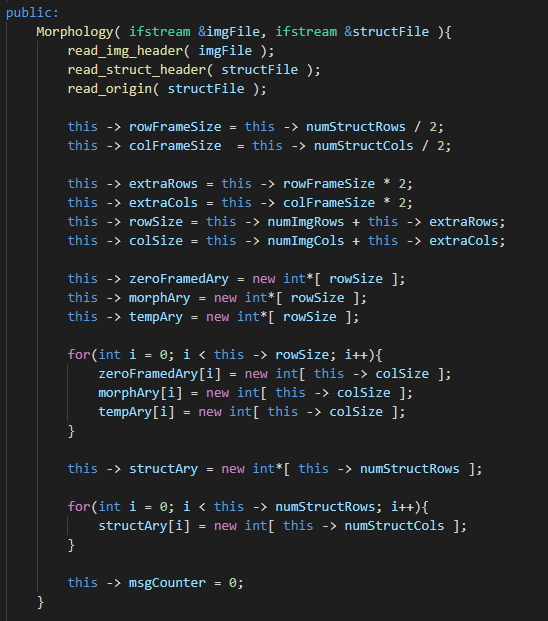
Main Class



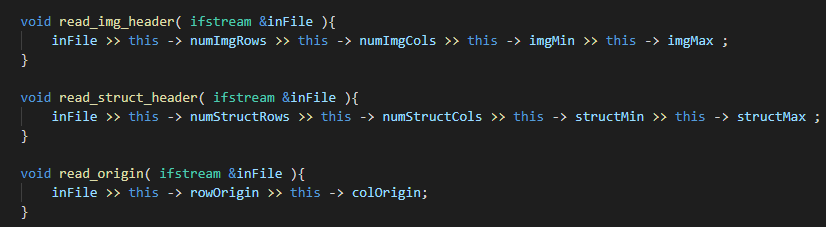
*Morphology Class*



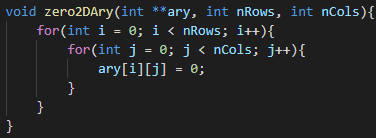
*Constructor*



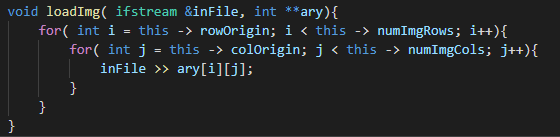
*Read Headers and origin*



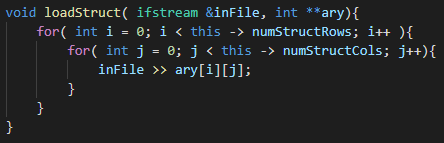
*Zero2DAry*



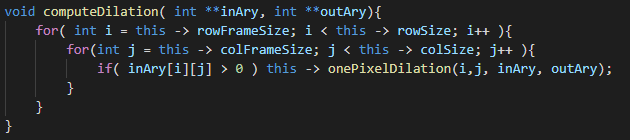
*loadImg*



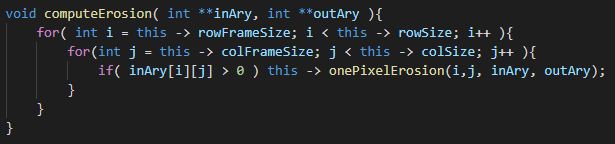
*loadStruct*



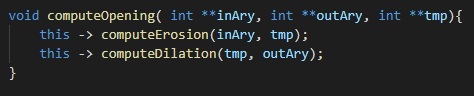
*computeDilation*



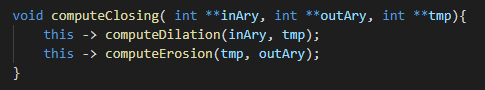
*computeErosion*



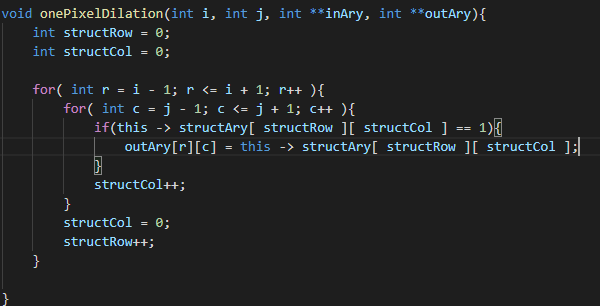
*computeOpening*



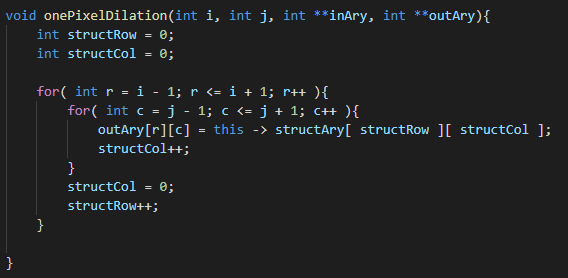
*computeClosing*



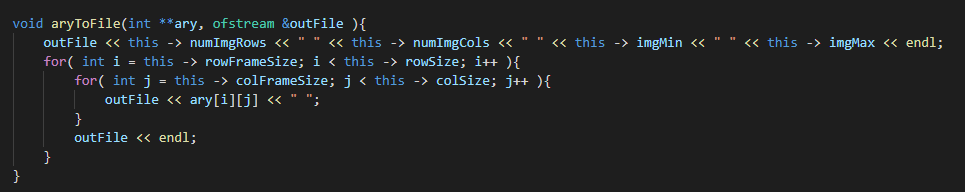
*onePixelDilation*



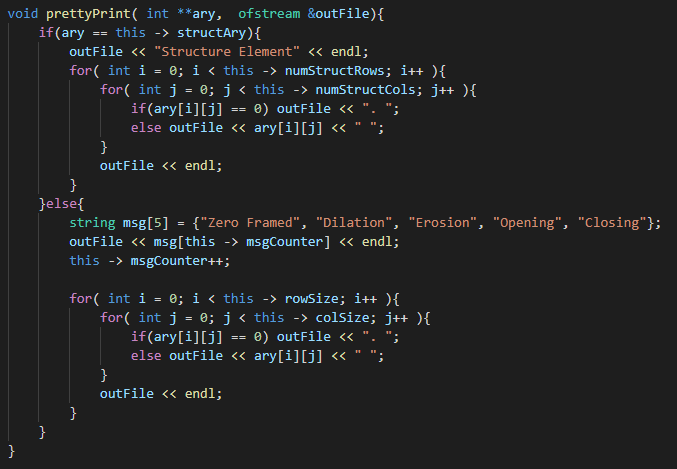
*onePixelErosion*



aryToFile



prettyPrint



*Test 1*

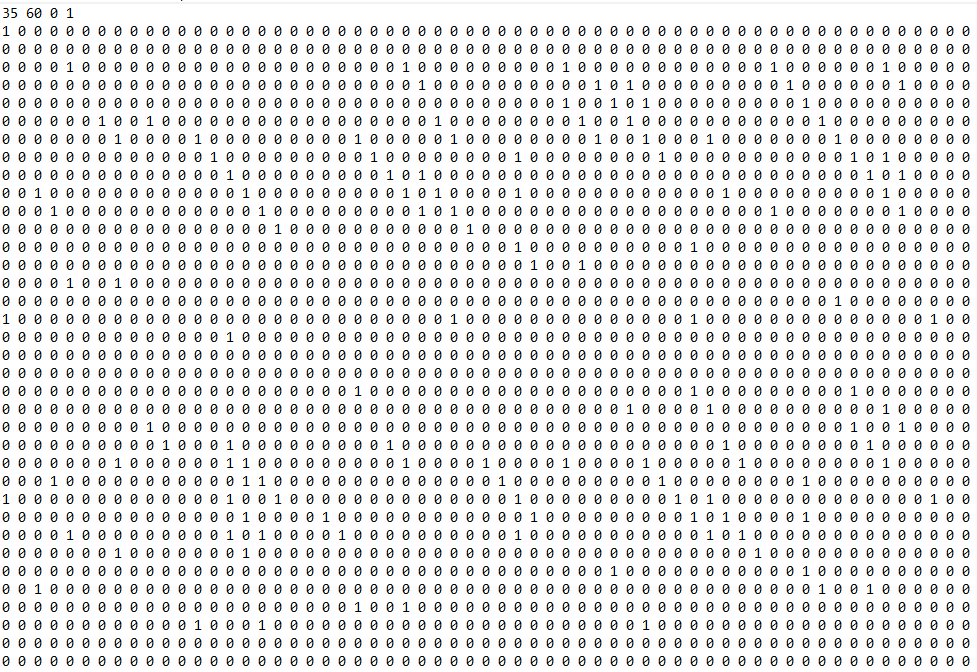
*dilateOutFile*



*erodeOutFile*



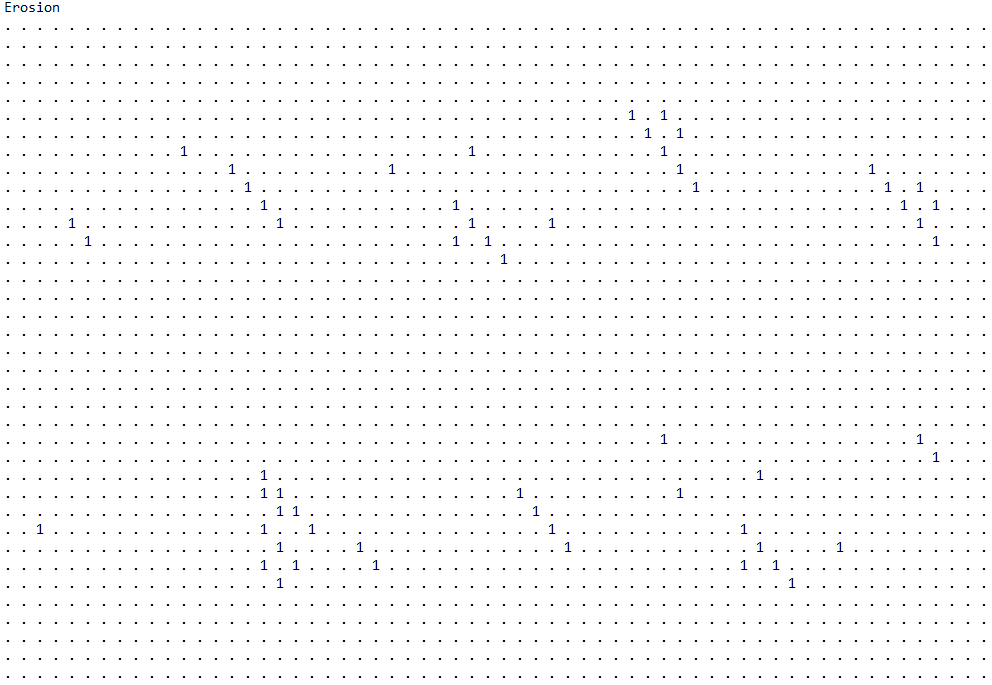
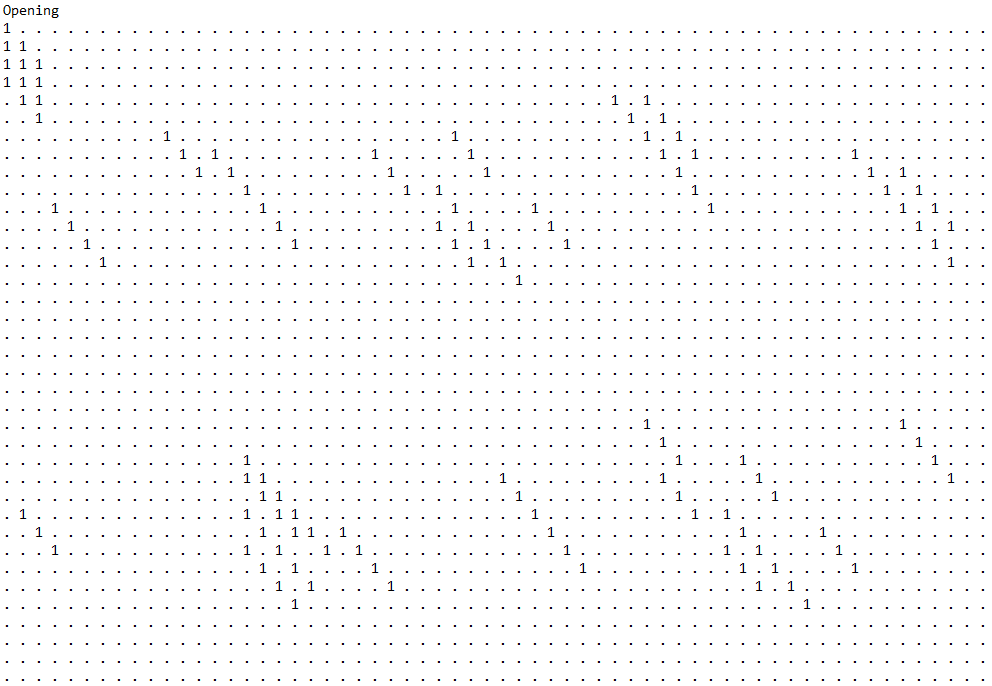
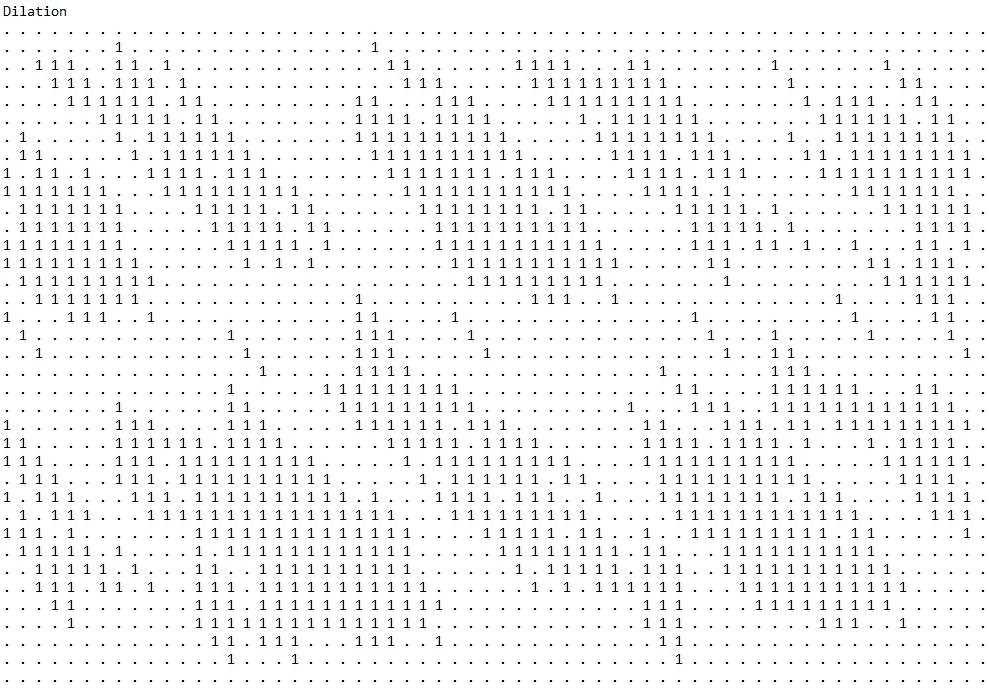
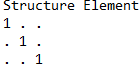
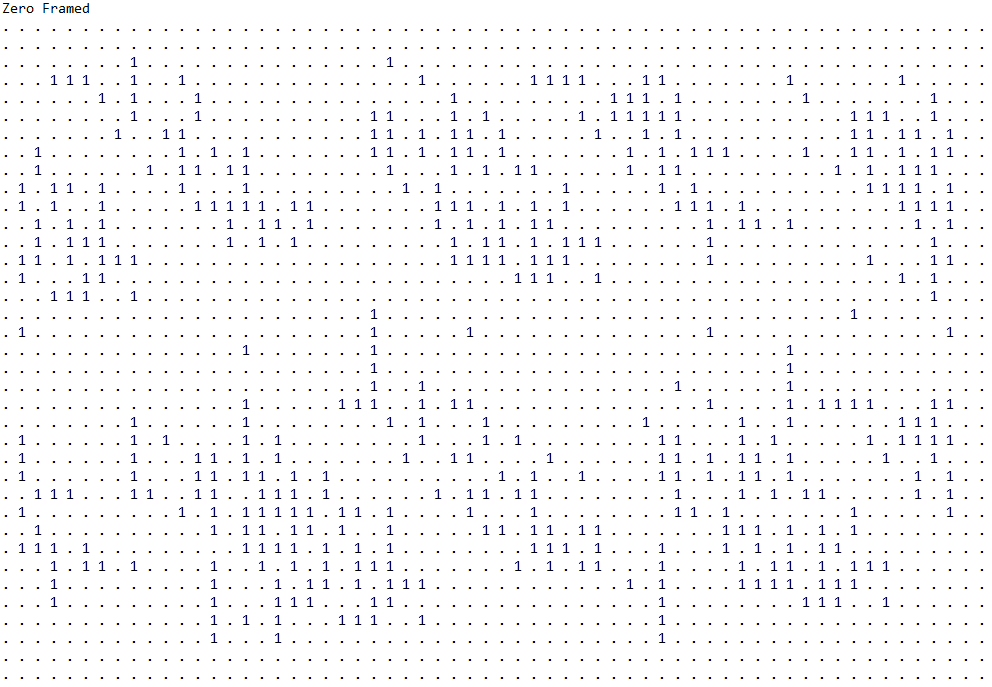
*openingOutFile*



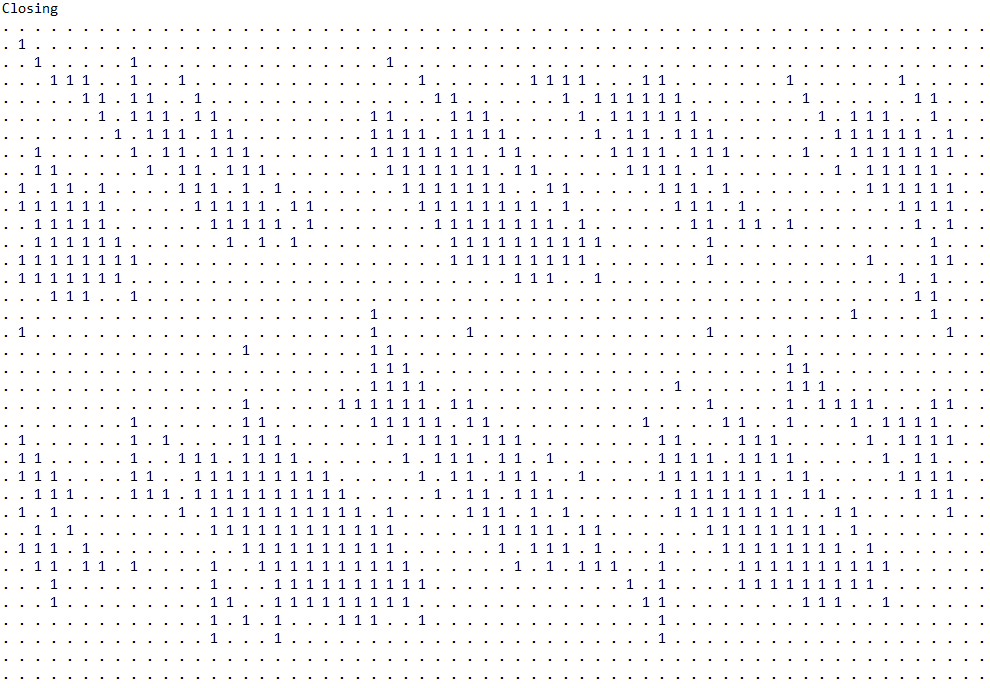
*closingOutFile*



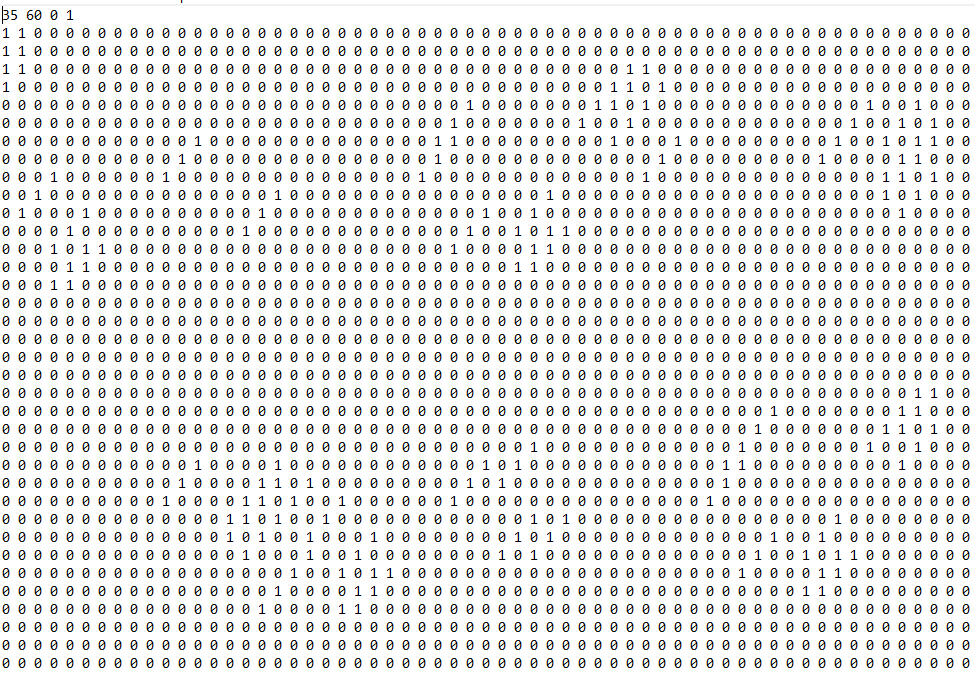
*prettyPrintFile*



*Test 2*



*closingOutfile*



*dilateOutFile*



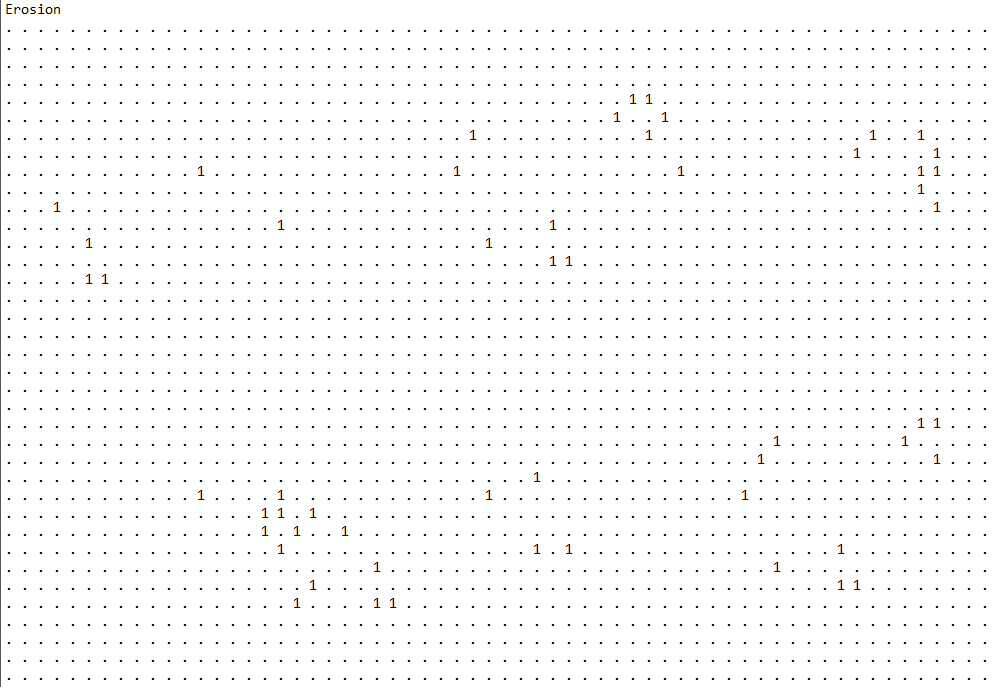
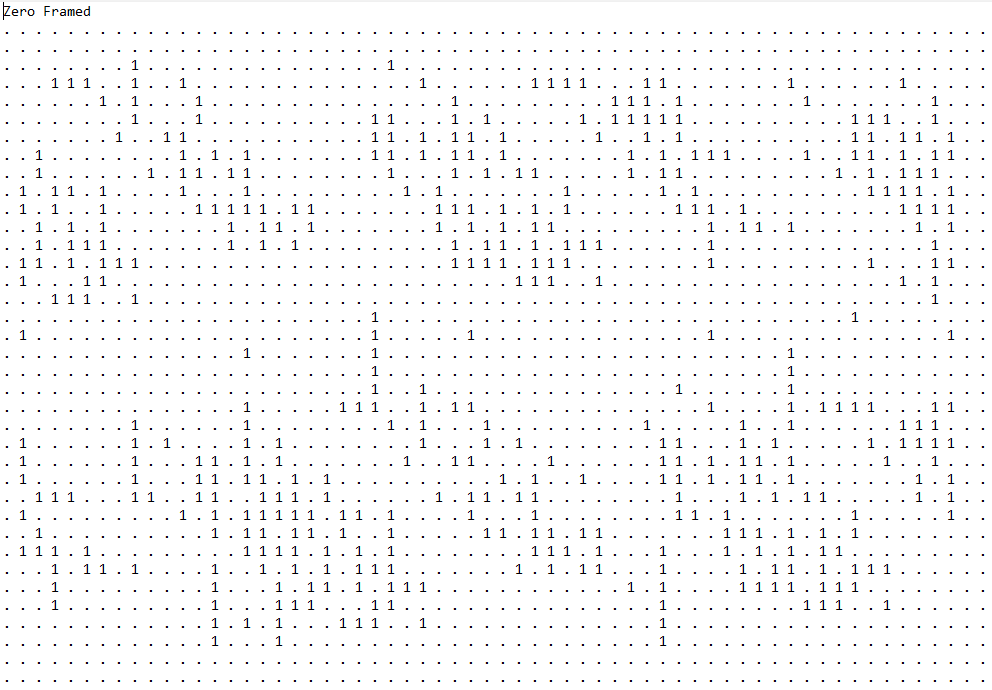
*erodeOutFile*



*openingOutFile*



*prettyPrintFile*



*Test 3*

*closingOutFile*



*dilateOutFile*



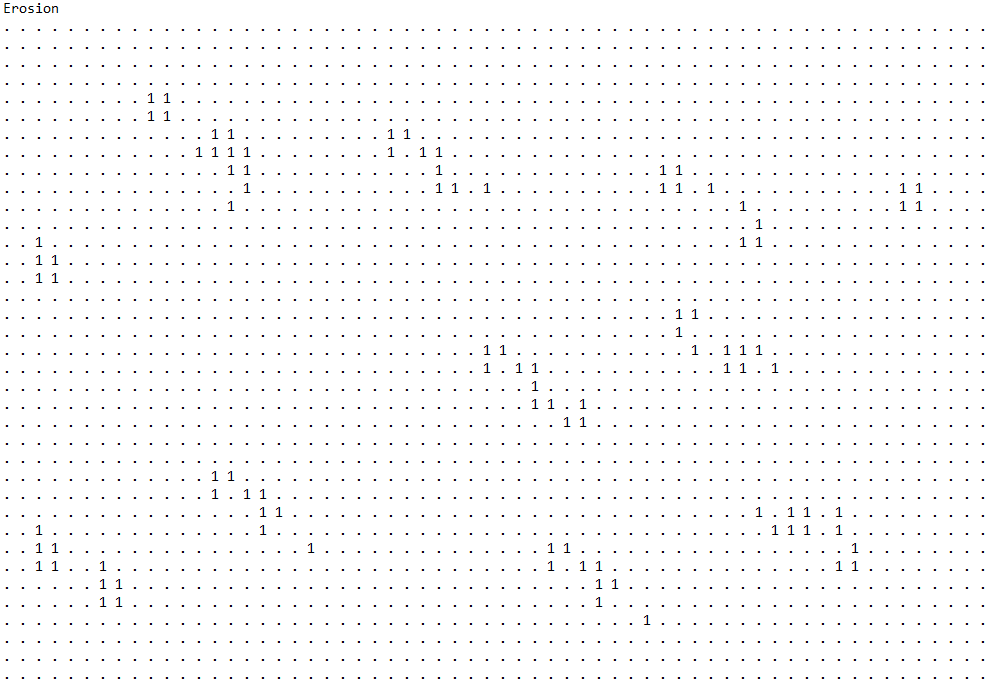
*erodeOutFile*



*openingOutFile*



*prettyPrintFile*



*Test 4*

*closingOutFile*



*dilateOutFile*



*erodeOutFile*



*openingOutFile*



*prettyPrintfile*

