\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CSCI 381 (CV): Project 1 Histogram Language: C++

Name: Akshar Patel

Due date: Soft copy: 2/4/2020

Hard copy: 2/6/2020

Submitted date: Soft copy: 2/4/2020

Hard copy: 2/6/2020

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Part 1: Algorithm**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

I. Data structure:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

- image class

- numRows (int)

- numCols (int)

- minVal (int)

- maxVal (int)

- histAry(int\*) //a 1D integer array, size of maxVal + 1

// need to be dynamically allocated at run time

- thresholdValue (int)

Methods:

- computeHist(...)

- printHist(...)

- thresholdOp(...)

- prettyPrint(...)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

II. main (...)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

step 0: inFile 🡨 open input file use argv[1]

outFile1, outFile2, outFile3 🡨 open via argv[2], argv[3], argv[4]

thresholdValue 🡨 get from argv[5]

step 1: numRows, numCols, minVal, maxVal 🡨 read from inFile

step 2: output numRows, numCols, minVal, maxVal to outFile1

output numRows, numCols, minVal, maxVal to outFile2

step 3: dynamically allocate the hist array and initialize to 0

step 4: ComputeHist (inFile, histAry) //algorithm is given below.

step 5: printHist(histAry, outFile1) //on your own, see the format in the above

step 6: close inFile

step 7: reopen inFile

step 8: thresholdOp(inFile, thresholdValue, outFile2)

//algorithm is given below.

step 9: close outFile2

step 10: reopen outFile2

step 11: prettyPrint (outFile2, outFile3) //algorithm is given below.

step 12: close all files

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

III. ComputeHist (inFile, histAry)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Step 1: pixel\_val 🡨 read from inFile // you must read one integer at a time

histogram[pixel\_val]++

step 2: repeat step 1 until the file is empty

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

IV. thresholdOp(inFile, thresholdValue, outFile2)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

step 0: iRow 🡨 0

step 1: jCol 🡨 0

step 2: pixel\_val 🡨 read from inFile

if pixel\_val >= threshold value

outFile2 🡨 write 1 follows by a blank

else

outFile2 🡨 write 0 follows by a blank

step 3: jCol++

step 4: repeat step 2 to step 3 while jCol < numCols

step 5: output the end of text line

step 6: iRow++

step 7: repeat step 1 to step 6 while iRow < numRows

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

V. prettyPrint (outFile2, outFile3)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

step 0: iRow 🡨 0

step 1: jCol 🡨 0

step 2: pixel\_val 🡨 read from outFile2

if pixel\_val > 0

outFile3 🡨 pixel\_val follow by a blank

else

outFile3 🡨 output 2 blanks

step 3: jCol++

step 4: repeat step 2 to step 3 while jCol < numCols

step 5: output the end of text line

step 6: iRow++

step 7: repeat step 1 to step 6 while iRow < numRows

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Part 2: Program (C++)**

#include <iostream>

#include<fstream>

using namespace std;

class image{

public:

int numRows, numCols, minVal, maxVal;

int computeHist(ifstream &inFile,int \* histAry){

int pixel\_val;

while(!inFile.eof()){

inFile>>pixel\_val;

histAry[pixel\_val]++;

}

return 0;

};

int printHist(int \* histAry,fstream &outFile1){

outFile1>>maxVal;

for(int i = 0;i<maxVal; i++){

outFile1<<i<<" "<<"("<<histAry[i]<<"):"<<string

(min(histAry[i],60), '+')<<endl;

}

return 0;

};

Int thresholdOp(ifstream& inFile,

int thresholdValue,fstream&outFile2){

inFile>>numRows;

inFile>>numCols;

inFile>>minVal;

inFile>>maxVal;

outFile2<<numRows<<" ";

outFile2<<numCols<<" ";

outFile2<<"0 ";

outFile2<<"1 "<<endl;

int pixel\_val;

for(int i=0;i<numRows;i++){

for(int j=0;j<numCols;j++){

inFile>>pixel\_val;

if(pixel\_val>=thresholdValue){outFile2<<"1 ";}

else{outFile2<<"0 ";}

}

outFile2<<endl;

}

return 0;

};

int prettyPrint(fstream& outFile2,fstream& outFile3){

outFile2>>numRows;

outFile2>>numCols;

outFile2>>minVal;

outFile2>>maxVal;

int pixel\_val;

for(int i=0; i<numRows;i++){

for(int j=0; j<numCols;j++){

outFile2>>pixel\_val;

if (pixel\_val>0){outFile3<<"1 ";}

else {outFile3<<" ";}

}

outFile3<<endl;

}

return 0;

};

};

int main(int argc, char\*\* argv){

int numRows, numCols, minVal, maxVal;

int \* histAry;

image img;

ifstream inFile;

inFile.open(argv[1]);

fstream outFile1;

outFile1.open(argv [2]);

fstream outFile2;

outFile2.open(argv [3]);

fstream outFile3;

outFile3.open(argv [4]);

int thresholdValue = atoi(argv [5]);

inFile>>numRows;

inFile>>numCols;

inFile>>minVal;

inFile>>maxVal;

outFile1<<numRows<<" ";

outFile1<<numCols<<" ";

outFile1<<minVal<<" ";

outFile1<<maxVal<<" "<<endl;

histAry = new int[maxVal];

img.computeHist(inFile, histAry);

inFile.close();

inFile.open(argv[1]);

img.printHist(histAry, outFile1);

img.thresholdOp(inFile,thresholdValue, outFile2);

outFile2.close();

outFile2.open(argv[3]);

img.prettyPrint(outFile2, outFile3);

inFile.close();

outFile1.close();

outFile2.close();

outFile3.close();

return 0;

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Part 3: Input & Output**

Input File**:** inFile.txt

31 40 0 9

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

2 3 1 3 2 3 1 1 2 3 1 1 2 1 1 1 2 1 0 1 1 0 0 0 0 0 0 0 0 0 2 3 1 3 2 3 1 1 2 3

0 1 1 9 2 1 2 1 2 9 2 2 2 1 2 1 2 1 2 1 8 1 2 1 1 1 2 1 1 1 2 1 1 9 9 1 1 1 2 1

2 3 2 3 2 3 2 1 2 3 2 1 2 3 2 3 2 3 1 7 9 9 1 1 2 3 2 1 2 3 2 3 1 1 0 3 1 1 2 3

0 0 2 3 1 8 2 3 0 2 3 1 2 3 1 1 2 3 9 8 8 7 9 2 3 1 1 2 3 2 3 1 1 2 2 3 1 1 2 3

1 0 1 2 0 2 2 0 3 0 3 0 1 0 2 0 1 7 7 9 9 8 8 7 0 1 2 0 2 2 0 3 0 3 0 1 0 2 0 2

0 0 2 1 1 1 2 1 1 1 2 1 1 1 2 1 9 8 8 7 9 8 0 7 9 0 0 0 1 1 2 1 1 1 2 1 1 1 2 1

2 1 1 1 2 1 1 1 2 1 1 1 2 1 1 8 7 9 9 7 7 0 7 8 8 8 2 1 1 1 2 1 1 1 2 1 1 1 2 1

0 1 2 1 0 1 2 1 2 1 2 2 2 0 8 7 9 8 7 7 9 9 8 8 7 9 9 1 2 0 2 0 1 3 2 0 1 1 2 0

2 3 2 3 2 3 2 1 2 3 2 1 2 8 7 9 8 7 7 9 9 8 8 7 9 9 7 9 1 1 2 1 1 1 2 1 1 1 2 1

0 1 3 2 0 1 1 2 0 0 0 0 8 7 9 8 7 7 9 9 8 8 7 9 9 8 7 9 9 0 0 0 0 3 2 0 1 1 2 0

0 0 0 3 2 0 1 1 2 0 0 8 7 9 8 7 7 9 9 8 8 7 0 0 7 9 8 8 7 8 3 2 0 1 1 2 0 0 0 0

0 3 2 0 1 1 2 0 1 0 7 9 8 7 7 9 9 8 8 7 9 9 0 8 9 9 8 8 7 8 9 0 0 3 2 0 1 1 2 0

0 3 2 0 1 1 2 0 0 9 8 8 8 9 8 8 7 7 7 9 9 7 9 9 8 8 7 8 9 7 7 9 3 2 0 1 1 2 0 0

0 1 3 2 0 1 1 2 7 9 8 7 7 9 9 8 9 8 8 7 0 0 7 7 8 7 8 9 7 8 8 7 9 3 2 0 1 1 2 0

3 2 0 1 1 2 0 9 8 9 8 8 7 9 8 7 7 9 9 8 0 7 9 9 6 9 8 8 7 8 6 9 8 8 2 1 1 1 2 3

0 0 0 0 0 0 1 1 7 9 9 6 9 8 8 7 9 9 8 8 7 9 9 8 7 9 9 6 9 8 8 7 9 2 1 2 1 0 0 0

0 0 1 9 0 0 0 1 1 9 8 8 7 9 0 0 7 9 9 8 8 7 9 9 6 0 0 8 7 8 6 9 2 1 0 2 9 0 0 0

0 0 8 1 0 0 0 0 1 1 8 7 7 9 0 0 9 8 8 7 9 9 7 7 8 0 0 9 7 8 8 2 1 0 0 1 9 0 0 0

3 2 0 1 1 2 0 0 0 1 1 8 8 9 8 8 7 7 7 9 9 7 0 9 8 8 7 8 9 7 2 1 8 9 0 0 0 0 0 0

0 1 9 3 0 1 1 2 0 0 1 1 8 7 7 9 9 8 8 7 9 9 7 7 9 9 8 8 7 2 1 0 8 9 2 0 1 1 2 0

0 0 2 9 0 1 1 2 0 0 0 1 1 9 8 7 7 9 9 8 8 7 9 9 1 9 8 8 2 1 1 1 2 1 1 1 2 1 1 1

0 2 0 1 3 2 0 1 1 2 0 1 1 1 9 8 7 7 9 9 8 8 7 9 9 8 7 2 1 0 2 0 1 3 2 0 1 1 2 0

0 1 3 2 0 1 1 2 0 0 0 0 0 1 1 9 8 7 7 9 9 8 8 7 9 9 2 1 1 1 2 1 1 1 2 1 1 1 2 1

1 1 2 1 1 1 2 1 1 1 2 1 2 0 1 1 9 8 7 7 9 9 8 8 7 2 1 1 2 0 2 0 1 8 9 0 1 1 2 0

2 3 2 1 2 3 2 1 2 0 0 0 0 0 0 1 1 9 9 7 7 9 9 8 2 1 1 1 1 1 2 1 1 8 9 1 1 1 2 1

0 0 8 9 0 3 2 0 1 1 2 0 1 0 0 0 1 1 8 7 9 9 7 2 1 0 0 0 1 1 2 1 1 1 2 1 1 1 2 1

0 2 2 9 1 2 3 2 1 2 0 0 0 0 0 0 0 1 1 9 9 8 0 1 0 1 2 0 2 2 0 3 9 3 0 1 0 2 0 2

0 0 0 0 0 0 0 0 0 2 3 2 7 2 3 2 1 2 1 1 8 0 1 0 0 0 0 0 0 0 0 0 0 0 0 8 0 0 0 0

0 1 1 1 2 1 2 1 2 8 2 2 2 1 2 1 2 1 2 1 2 1 2 1 1 1 2 1 1 1 2 1 1 1 2 1 1 1 2 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

Output file 1: outFile1.txt

31 40 0 9

0 (314):++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

1 (294):++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

2 (196):++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

3 (64):++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

4 (0):

5 (0):

6 (6):++++++

7 (102):++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

8 (124):++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

9 (141):++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Output file 2: outFile2.txt

31 40 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 1 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 1 1 0 0 0 0 0

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

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

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

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

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

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

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

Output file 3: outFile3.txt

1 1 1 1 1

1 1 1

1 1 1 1 1 1

1 1 1 1 1 1 1

1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1

1 1 1 1 1

1 1 1

1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*