**CSCI 381 – Computer Vision (C++)**

**Program: Project 1.2: Threshold Algorithm**

**Name: Isaac Gordon**

**Due Date:**

**Soft copy: 2/04/2019 Monday before midnight**

**Hard copy: 2/05/2019 Tuesday in class**

step 0: make sure all arguments are valid

step 1: inFile1 <-- argv[1]

outFile1 <-- argv[2]

outFile2 <-- argv[3]

step 2: numRows, numCols, minVal, maxVal <-- read from inFile

step 3: process the file from left to right and top to bottom;

create data array

step 4: thrValue <-- ask user from console

step 5: write numRows, numCols, 0, 1 to outFile1

step 6: process the data array from left to right and top to bottom

// For easier reading use two loops: i for rows and j for column

pixel\_val <-- read from data array

if pixel\_val >= threshold value

outFile1 <-- write 1 follows by a blank

outFile2 <-- write 1 follows by a blank

else

outFile1 <-- write 0 follows by a blank

outFile2 <-- write 2 blanks

step 7: repeat step 6 until the inFile1 is empty

step 8: close all files

**CODE:**

#include <fstream>

#include <iostream>

#include <string>

using namespace std;

bool endsWith(string str, string ex){

int pos = str.find(ex);

if(pos != str.size() - 4) return false;

return true;

}//endsWith

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

//set arg error message

string BAD\_ARGS = "Correct arguement format is \"<inputFile> <binaryOutputFile> <PrettyOutputFile>\". \nAll should end in \'.txt\'.";

//check for correct number of args

if(argc != 4){

cout << "Wrong number of arguements.\n" << BAD\_ARGS << endl;

exit(1);

}//if

//make sure they are all text files

for(int i = 1; i < argc; i++){

if(!endsWith(argv[i], ".txt")){

cout << argv[i] << " is not a .txt file. Try again." << endl;

exit(1);

}//if

}//for

//declare data structs

ifstream inFile1;

ofstream outFile1, outFile2;

inFile1.open(argv[1]);

outFile1.open(argv[2]);

outFile2.open(argv[3]);

//if input file could not be opened then exit

if(!inFile1.is\_open()){

cout << "Input File could not be opened. " << endl;

exit(1);

}//if

//if onput file 1 could not be opened then exit

if(!outFile1.is\_open()){

cout << "Output File 1 could not be opened. " << endl;

exit(1);

}//if

//if output file 2 could not be opened then exit

if(!outFile2.is\_open()){

cout << "Output File 2 could not be opened. " << endl;

exit(1);

}//if

int numRows, numCols, minVal, maxVal;

//get header values

inFile1 >> numRows;

inFile1 >> numCols;

inFile1 >> minVal;

inFile1 >> maxVal;

int header[] = {numRows, numCols, minVal, maxVal};

//make 2D array of pixel vals

int\*\* intData = new int\*[numRows];

for(int i = 0; i < numRows; i++) intData[i] = new int[numCols];

//fill array

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

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

inFile1 >> intData[r][c];

}//inner for

}//outer for

//get user defined threshold and generate files

int threshVal;

while(true){

cout << "Pick a threshold value between 0 and " << maxVal << " then press ENTER: ";

cin >> threshVal;

if(threshVal > 0 && threshVal <= maxVal) break;

cout << endl;

}//while

//add header to foreground img

outFile1 << numRows << " ";

outFile1 << numCols << " ";

outFile1 << "0" << " ";

outFile1 << "1" << " ";

outFile1 << endl;

//create both foregounnd img and pretty print foreground img

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

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

if(intData[r][c] >= threshVal){

outFile1 << 1 << " ";

outFile2 << 1 << " ";

} else {

outFile1 << 0 << " ";

outFile2 << " " << " ";

}

}//inner-for

outFile1 << endl;

outFile2 << endl;

}//outer-for

return 0;

}

**Outputs:**

**Result of Threshold 5:**

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

**Pretty Print:**

1 1 1 1 1

1 1 1

1 1 1 1 1 1

1 1 1 1 1 1 1

1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1

1 1 1 1 1

1 1 1

1