

Project 0A: (in C++) This project is for you to practice on a simple I/O in C++ via the implementation of two (2) threshold operations: a) the binary threshold, and b) the non-binary threshold operations. If you know C++, this project can be easily done with little time. ** There is a simple requirement for you to record 2 minutes of screen recording while you are doing compilation and running of the program. -2 if you do not submit screen recording with your submission. See the screen recording requirements guide posted on Blackboard. *** Place your screen recording YouTube link in your project submission email body below the 5 questions.

- a) The binary threshold operation: given a grey-scale image and a threshold value, the binary threshold operation is to transform pixels in the input image (imgIn) from grey-scale to binary values in the output image (imgOut) where

```
if imgIn (i, j) >= threshold value
    imgOut (i, j) ← 1
else
    imgOut (i, j) ← 0
```

- b) The non-binary threshold operation: Given a grey-scale image, and a threshold value, the non-binary threshold is to transform pixels in the input image (imgIn) from grey-scale to non-binary values in the output image (imgOut) where

```
if imgIn (i, j) >= threshold value
    imgOut (i, j) ← imgIn (i, j)
else
    imgOut (i, j) ← 0
```

=====

What you have to do:

- 1) implement the program according to the specs given below.
- 2) compile your program until it passes compilation
- 3) run your program with data1

Include in your hard copy (pdf file) // See Project Submission Requirement for how to create hard copy.

- a cover page
- source code
- inFile
- outFile1
- outFile2

Language: C++

Project Name: Binary and non-binary thresholding

Project points: 4 pts (-2 if not submit screen recording) //Screen recording requirements is posted on Blackboard.

Due Date: (4/4) 2/3/2023 Saturday before midnight

(-4/4) 2/3/2023 Saturday after midnight

*** Name your soft copy and hard copy files using the naming convention in the **project submission requirement**.

*** All submission MUST include Soft copy (*.zip) and hard copy (*.pdf) in the same email attachments with correct email subject as stated in the project submission requirement; otherwise, your submission will be rejected.

*** **place your screen recording in your project submission email body below the 5 questions.**

*** **Email subject: (CV) first name last name <Project 0A: binary and non-binary thresholding (C++)>**

I. Inputs:

- a) inFile (argv [1]): a txt file representing a grey-scale image, where the first text line (4 integers) is the "header" of the input image then follows by rows and cols of integers. The header of an image consists of four integers:
- numRows – number of rows in the input image
 - numCols – number of columns in the input image
 - minVal – the minimum grey scale value in the input image
 - MaxVal – the maximum grey scale value in the input image

For example,

```
4 6 1 12    // image has 4 rows, 6 cols, min is 1, max is 12
2 3 4 11 2 9
5 6 11 2 10 7
1 1 12 1 9 9
4 5 6 9 9 9
```

- b) thrValue (argv[2]) : for this project use threshold value 43.

II. a) outFile1 (argv [3]): The result of the binary threshold operation of input image.

Note: The output binary image also needs to have the image header.

For example, given the above image and 6 as the threshold value then the binary image would be:

```
4 6 0 1          // notice the min and max values have changed!
0 0 0 1 0 1
0 1 1 0 1 1
0 0 1 0 1 1
0 0 1 1 1 1
```

- b) outFile2 (argv [4]): The result of the non-binary threshold operation of input image.

Note: The output non-binary image also needs to have the image header.

For example, given the above image and 6 as the threshold value then the non-binary image would be:

```
4 6 0 12         // notice the min value has changed!
0 0 0 11 0 9
0 6 11 0 10 7
0 0 12 0 9 9
0 0 6 9 9 9
```

III. Data structure:

- Image class

- (int) numRows
- (int) numCols
- (int) minVal
- (int) maxVal
- (int) thrValue

Method: constructor ()

Processing (...)

IV. SmithJ_Project0A_Main (...) // replace “SmithJ” with your last name and first name initial!!

step 0: inFile ← open argv [1] // i.e., ifstream inFile.open (argv [1])
thrValue ← cast argv [2] from string to integer // i.e., int (argv [2]) or (int) argv [2]
outFile1 ← open argv [3]
outFile2 ← open argv [4]
numRows ← read from inFile // i.e., inFile >> numRows
numCols ← read from inFile
minVal, maxVal ← read from inFile
maxVal ← read from inFile

step 1: outFile1 ← output numRows, numCols, 0, 1 to outFile1 // i.e., outFile1 << numRows etc.
outFile2 ← output numRows, numCols, 0, maxVal to outFile2

step 2: processing (inFile, outFile1, outFile2, thrValue)

step 3: close all files

V. processing (inFile, outFile1, outFile2, thrValue) // the 3 files need to pass by references (i.e., precede by &)

step 0: (int) pixelVal
(int) count ← 0

step 1: pixelVal ← read one integer from inFile // i.e., inFile >> pixelVal

step 2: if pixelVal >= thrVal
outFile1 ← write 1 follows by 1 blank // binary threshold.
outFile2 ← write pixelVal follows by 1 blank //non-binary threshold.
else
outFile1 ← write 0 follows by 1 blank
outFile2 ← write 0 follows **by 2 blanks**

step 3: count++

step 4: if count > numCols
outFile1 ← write end of text line
outFile2 ← write end of text line
count ← 0

step 5: repeat step 1 - step 4 until the inFile is empty