Project 2 (in C++): Given a bimodal histogram, and the two points on the two peaks of the bimodal histogram, you are to implement the deepest concavity method for automatic threshold selection (taught in class). (An easy project!)

****

Project points: 10 pts

Language: C++

Due Date: Soft copy (*.zip) and hard copies (*.pdf):

+1 (11/10 pts): early submission, 2/18/2022, Friday before midnight
-0 (10/10 pts): on time, 2/21/2022 Monday before midnight
-1 (9/10 pts): 1 day late, 2/22/2022 Tuesday before midnight
-2 (8/10 pts): 2 days late, 2/23/2022 Wednesday before midnight
(-10/10 pts): non submission, 2/23/2022 Wednesday after midnight

*** Name your soft copy and hard copy files using the naming convention as given in the project submission requirement.
*** All on-line submission MUST include Soft copy (*.zip) and hard copy (*.pdf) in **the same email attachments** with correct email subject as stated in the email requirement; otherwise, your submission will be rejected.

==

There are two sets of data: set1 <data1_hist and data1_2pts> and set2 <data2_hist and data2_2pts

1. Implement your program as given the specs below.
2. Run your program on set1 and set2.
3. Include in your hard copy *.pdf file as follows:
   - Cover page.
   - Source code.
   - outFile for set1.
   - outFile for set2.

****

I. Inputs:

****

a) inFile1 (argv [1]): a text file representing a histogram of a gray-scale image. The input format as follows:

For example:

```
5  7  0  9     // 5 rows, 6 cols, min is 0 max 9
0  2           // hist [0] is 2
1  8           // hist [1] is 8
2  5           :
```

b) inFile2 (argv [2]): contains four integers (representing the two peak points of the input bimodal histogram).

****

II. outFile (argv [3]): // use small font so that the entire histogram can be displayed on the page.

****

This file includes the followings:

a) A 2-D display of the histogram, as done in your project 1. // With proper caption
b) The selected threshold value. // with proper caption
c) A 2-D display of the graph that includes: histogram points, line points and gap points between the line and the histogram. // With proper caption

****

III, Data structure:

****

- a Concavity class
  - (int) numRows, numCols, minVal, maxVal
  - (int) x1, y1, x2, y2 // The 2 points of the two peaks of the histogram.
  - (double) m // the slop of the line equation, y = m*x + b
  - (double) b // the y-interception of the line equation, y = m*x + b.
  - (int) histAry [] // a 1D integer array (size of maxVal + 1) to store the histogram.
    // It needs to be dynamically allocated at run time; **initialize to zero**.
  - (int) maxHeight // The max of histAry[i], i = minVal to maxVal
  - (int) bestThrVal // the auto selected threshold value by the method.

- (int) displayGraph [] [] // a 2-D int array size of maxVal+1 by maxHeight+1, **initialize to 0**,
  // It needs to be dynamically allocated at run time. Within displayGraph [] []: 1 for histogram points,
  //2 for line points and 3 for gaps points, 0 for nothing.

Methods:

- constructor (…) // It dynamically allocates all member arrays and initialization.
- (int) loadHist (…) // reads and loads the histAry from inFile and **returns** the max hist[i]. // On your own
- printHist (outFile) // Reuse code from your project 1.
- (int) deepestConcavity (…) // this is the principal method that determines the best threshold selection.
  // See algorithm below.
- plotOneRow (x, y, displayGraph) // Assign 1 to points of histAry[x], 2 to line points and 3 to gap points
  // See algorithm below.
- printGraph (…)

```
printGraph (…) // if displayGraph[i][j] == 0
                     outFile ← blank
               // if displayGraph[i][j] == 1
                     outFile ← '*'
               // if displayGraph[i][j] == 2
                     outFile ← '+'
               // if displayGraph[i][j] == 3
                     outFile ← '='
```

// Use fix font (such as 'courier') to output graph, so it will line up nicely.

****

IV. Main (...) // debug if needed.

****

```
Step 0: inFile1, inFile2, outFile← open via argv []
Step 1: numRows, numCols, minVal, maxVal ← read from inFile1.
        x1, y1, x2, y2 ← read from inFile2.
        histAry ← dynamically allocate (size of maxVal + 1) and initialized to zero.
        maxHeight ← loadHist (histAry, inFile) // loadHist ( ) returns maxHeight.
        dynamically allocate all other arrays and initialized to zero.
Step 2: printHist ()
Step 3: m ← (double) (y2-y1) / (double) (x2-x1)
        b = (double) y1 - (m * (double) x1)
Step 4: bestThrVal ← deepestConcavity (x1, x2, m, b, displayGraph)
Step 5: outFile ← output bestThrVal // with caption
Step 6: printGraph (displayGraph, outFile)
Step 7: close all files
```

****

VI. (int) deepestConcavity (x1, x2, m, b, displayGraph)

****

```
Step 0: max ← 0
        first ← x1
        second ← x2
        x ← first
        thr ← first
Step 1: y ← (int) (m * x + b)
Step 2: plotOneRow (x, y, displayGraph)
Step 3: gap ← (abs) (hist[x] - y)
Step 4: if gap > max
            max ← gap
            thr ← x
Step 5: x++
Step 6: repeat step 1 to step 5 while x <= second
Step 7: return thr
```

****

VI. plotOneRow (x, y, displayGraph)

****

```
step 1:   index ← min (histAry[x], y)
          last ← max (histAry[x], y)
Step 2: displayGraph[x][index] ← 3 // use 3 for gap point
Step 3: index ++
Step 4: repeat step 2 to step 3 while index<= last
Step 5: displayGraph[x][histAry[x]] ← 1 // use 1 for histogram point
        displayGraph[x][last] ← 2              // use 2 for line point
```