

Name: Jingshi Liu

Section: Image Processing

Project: **Project 0B - Binary and Non-binary Thresholding Operations**

Due Date: Sept 4nd

## Algorithm Steps

step 0:

```
inFile <- open args[0]
```

```
outFile1 <- open args[1] // i.e., FileWriter outFile1 = new FileWriter(args[1]);
```

```
outFile2 <- open args[2]
```

```
numRows, numCols, minVal, maxVal <- read from inFile
```

```
thrValue <- ask user from console // Scanner in = new Scanner(system.in);
```

step 1:

```
outFile1 <- output numRows, numCols, 0, 1
```

```
outFile2 <- output numRows, numCols, 0, maxVal
```

step 2:

```
processing (inFile, outFile1, outFile2, thrValue)
```

step 3: close all files

Video: <https://youtu.be/jZ6FB6rErhE>

## **Source Code:**

```
#include <iostream>
#include <fstream>

using namespace std;

void processing(istream& inFile, ofstream& outFile1, ofstream& outFile2, int
thrVal, int numCols){
    int pixelVal;

    int i = 0;
    while(inFile >> pixelVal){
        if(pixelVal >= thrVal){
            outFile1 << 1 << " ";
            outFile2 << pixelVal << " ";
        }else{
            outFile1 << 0 << " ";
            outFile2 << 0 << " ";
        }
        if(++i == numCols){
            outFile1 << '\n';
            outFile2 << '\n';
            i = 0;
        }
    }
}

int main(int argc, const char * argv[]){
    ifstream inFile;
    ofstream outFile1, outFile2;

    inFile.open(argv[1]);
    outFile1.open(argv[2]);
    outFile2.open(argv[3]);

    int numRows, numCols, minVal, maxVal;
    inFile >> numRows >> numCols >> minVal >> maxVal;

    int thrVal;
    cout<<"Enter a Threshold Value: ";
    cin>>thrVal;
    cout<<endl;

    outFile1<< numRows<< " "<< numCols<< " "<< minVal<< " "<< 1<< '\n';
    outFile2<< numRows<< " "<< numCols<< " "<< minVal<< " "<< maxVal<< '\n';

    processing(inFile, outFile1, outFile2, thrVal, numCols);

    inFile.close();
    outFile1.close();
    outFile2.close();
}
```

## Program Output

## Output 1

31 40 0 1

[illegible]

[illegible]

## Output 2

31 40 0 9

[illegible]

00000000699990099009999999998700000000  
0000000569999999009999999998700000000  
000000557999999999999999099998700000050  
00000005699999999999990999998700000000  
00000005599799999999990099999000090000  
00000000588888888988887888888700000900  
0000005659979778788797977709777700000000  
00000000056565656565656565656560000000  
0000000000000000000000000000000000080000  
000000000000000000000000000000000000000  
000000000000000000000000000000000000000  
000000000000000000000000000000000000000  
0000000000000005000000000500000000050000  
000000005000000000000000000000000000000  
000050000000000000000000000000000000000  
000000000000000000000000000000000000000