COMMAND LINE IMAGE EDITOR

PROBLEM STATEMENT: Create A Command Line Image Editor in JAVA using Buffered Images and perform various editing features and additional functionalities to enhance editing tool.

STRUCTURE CHART:

1. CONVERT TO GRAYSCALE
2. CLOCKWISE ROTATE
3. ANTI - CLOCKWISE ROTATE
4. HORIZONTAL INVERT
5. VERTICAL INVERT
6. CHANGE BRIGHTNESS
7. BLURR IMAGE

Terminates the program

1. CONVERT TO GRAYSCALE:

INPUT: BUFFERED INPUT IMAGE

We store the values of height and width of Buffered Image in variables “height” and “width” and then we create a new Output Image of type BUFFERED.TYPE_BYTE_GRAY since we want to grayscale the image. We iterate over all coordinates of image by using nested loops and sets the RGB values of INPUT IMAGE to OUTPUT IMAGE.

RETURN: BUFFERED GRAYSCALED OUTPUT IMAGE

2. ROTATE CLOCKWISE:

INPUT: BUFFERED INPUT IMAGE, INT DEGREE

To achieve this, we used 2 functions TRANSPOSE OF IMAGE and HORIZONTAL INVERT.

TRANPOSE OF IMAGE:

INPUT: BUFFERED INPUT IMAGE

This time we make the OUTPUT IMAGE with (“width” = height of INPUT IMAGE) and (“height” = width of INPUT IMAGE) because for a non-square matrix (M x N) the transpose matrix would be (N x M). So, we iterate over all the coordinates of INPUT IMAGE (j , i) and store the RGB values in (i , j) of the OUTPUT IMAGE.

RETURN: BUFFERED TRANSPOSED OUTPUT IMAGE

HORIZONTAL INVERT:

INPUT: BUFFERED INPUT IMAGE

We iterate over all the rows of the INPUT IMAGE and half the columns. We store the value of the coordinate’s RGB values (j,i) in a new Color pixel and then assign it RGB values of (width – j – 1 , i). Then we assign (width – j – 1 , i) RGB values of the Color. We run the loop till half the columns because all the coordinates in 1 specific row would be interchanged with their inversion coordinates giving us the horizontally inverted image.

RETURN: BUFFERED HORIZONATALLY INVERTED OUPUT IMAGE

Now in the ROTATE CLOCKWISE FUNCTION, we do degree by 90 to find how much image must be rotated. Then we do modulo by 4 because after 4 rotations we come to original image.

Then if degree is 1 that means 1 rotation only (90 degree), so we transpose the image and horizontal invert only once. For degree 2 we perform the degree 1 process 2 times and for degree 3, 3 times.

RETURN: BUFFERED CLOCKWISE ROTATED OUTPUT IMAGE

3. ROTATE ANTI - CLOCKWISE:

INPUT: BUFFERED INPUT IMAGE

This is same as that CLOCKWISE ROTATE function. The only difference is that now if degree is 1 then we must perform transpose followed by horizontal invert 3 times because anti – clockwise 90-degree is equivalent to 270-degree clockwise rotation.

RETURN: BUFFERED ANTI-CLOCKWISE ROTATED OUTPUT IMAGE

4. HORIZONTAL INVERT:

Already discussed in the ROTATE CLOCKWISE SECTION.

5. VERTICAL INVERT:

INPUT: BUFFERED INPUT IMAGE

This is similar to HORIZONTAL INVERT function. The only difference is that this time we iterate over all columns and half the rows because the upper pixel's RGB values must be interchanged with lower pixel's RBG values.

RETURN: BUFFERED VERTICALLY INVERTED IMAGE

6. CHANGE THE BRIGHTNESS:

INPUT: BUFFERED INPUT IMAGE, DOUBLE INCREASE

In this function we go over all the pixels one by one with help of nested loops and for every pixel we contain RBG values in Color and then we increase or decrease separately red , green , blue values by INCREASE argument. Then we store those values in the OUTPUT IMAGE's pixel of same coordinate.

We also make sure that values do not exceed 255 or go below 0 because RGB values range from [0,255].

RETURN: BUFFERED CHANGED BRIGHNTESS OUTPUT IMAGE

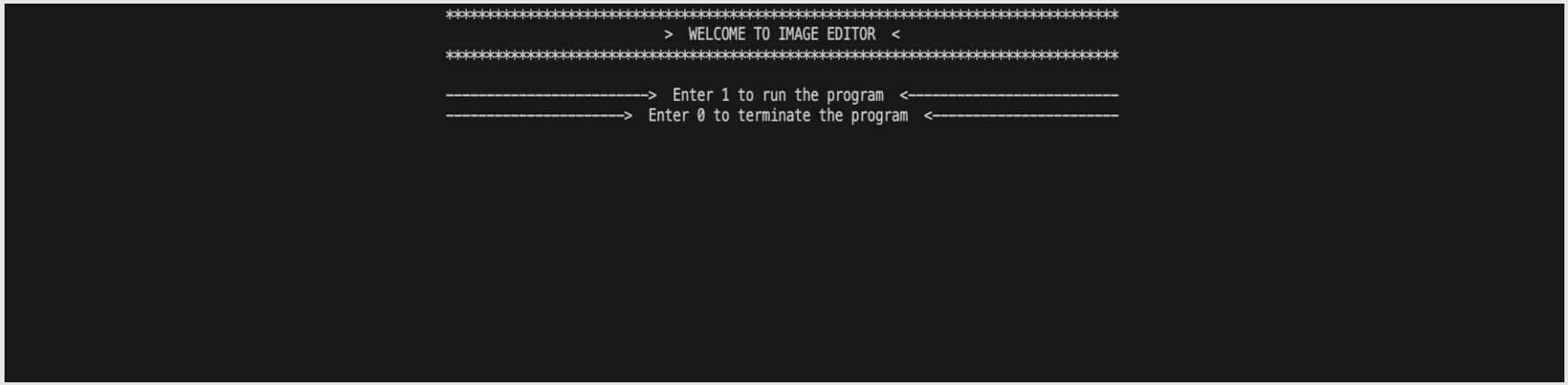
7. BLURRNESS:

INPUT: BUFFERED INPUT IMAGE, INT blurrnessLevel

To achieve blurrness the basic idea is that we iterate for every (blurrnessLevel * blurrnessLevel) matrices and take average of RGB values then set them in all the pixels of that matrix in OUTPUT IMAGE. We introduce 4 new variables heightlimit , widhtlimit , temp1 and temp2. So every time we iterate over rows = blurrnessLevel , widhtlimit will track the count of indices we are iterating over the columns so that we don't exceed the actuall width of the INPUT IMAGE. The heightlimit makes sure we don't exceed the height of INPUT IMAGE. The temp1 and temp2 are used for accessing the first indices of the current matrix while iterating so that we are always taking unique matrices. In loop we introduce 3 new variables red , green and blue to take the sum of all RGB values of all pixels of matrix. Then we average them out by total number of pixels and store them in all pixels of same matrix of OUTPUT IMAGE.

RETURN: BUFFERED IMAGE BLURRED OUTPUT IMAGE

PROGRAM INTERFACE:



SAMPLE RUN:

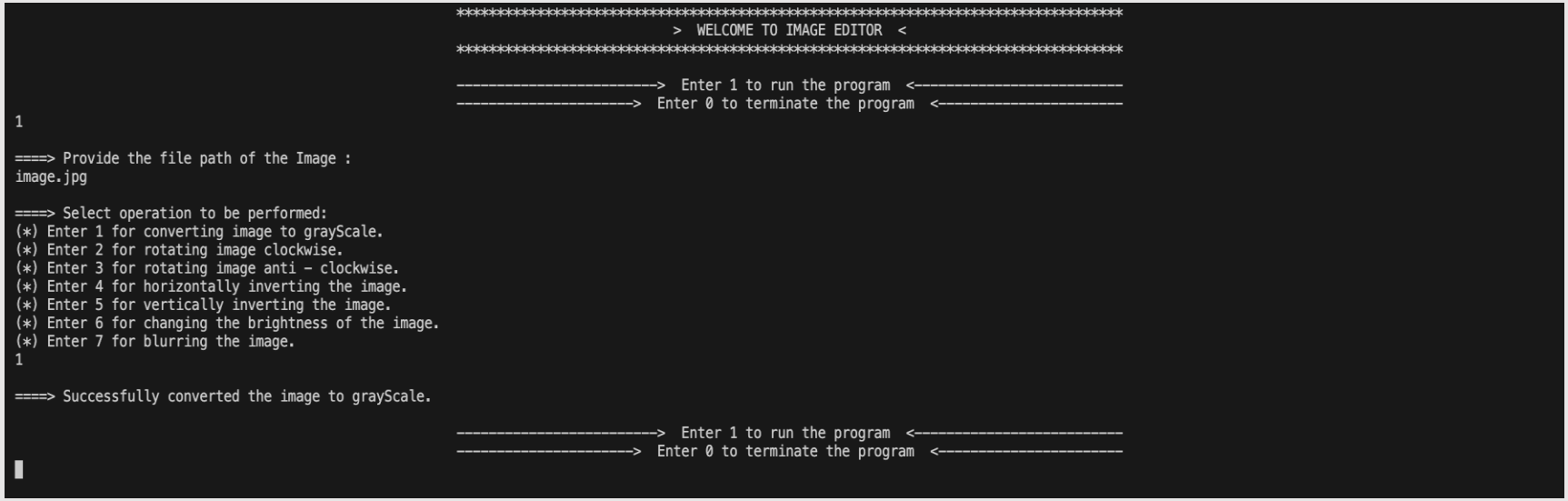
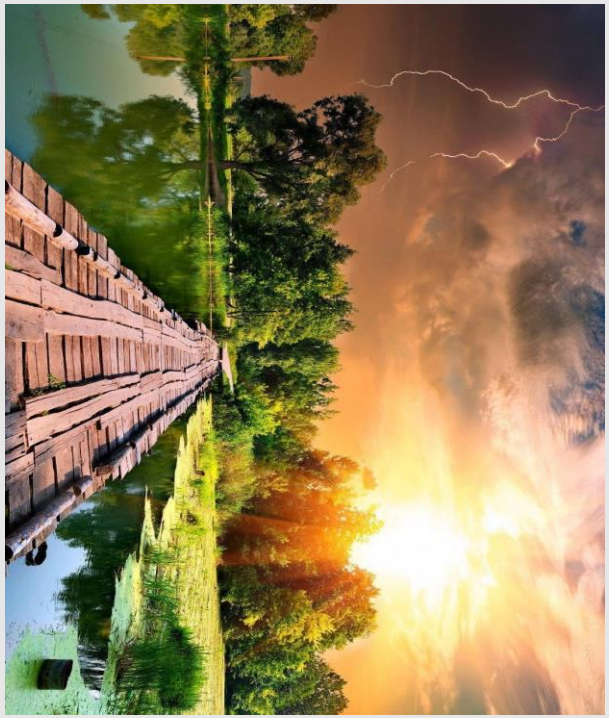


image.jpg

grayscaleImage.jpeg

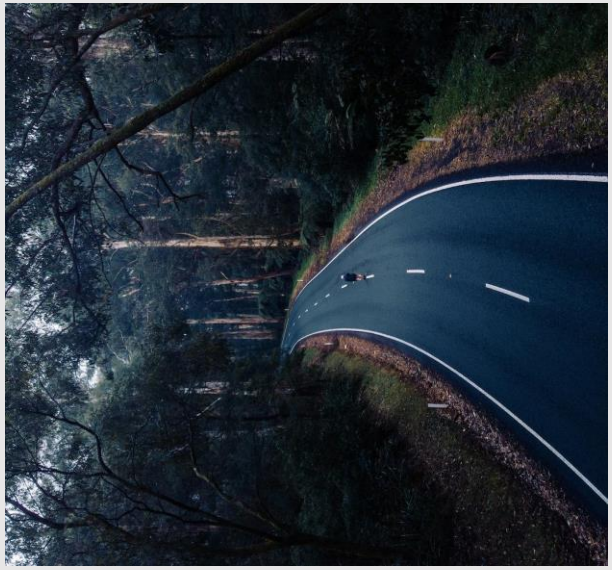
```
                                                  ------------------------->  Enter 1 to run the program  <-------------------------
                                                  ----------------------->  Enter 0 to terminate the program  <-----------------------
1

====> Provide the file path of the Image :
image.jpg

====> Select operation to be performed:
(*) Enter 1 for converting image to grayScale.
(*) Enter 2 for rotating image clockwise.
(*) Enter 3 for rotating image anti - clockwise.
(*) Enter 4 for horizontally inverting the image.
(*) Enter 5 for vertically inverting the image.
(*) Enter 6 for changing the brightness of the image.
(*) Enter 7 for blurring the image.
2

Enter degree by which image has to be rotated (POSITIVE MULTIPLE OF 90) :
810

====> Successfully rotated the image clockwise.

                                                  ------------------------->  Enter 1 to run the program  <-------------------------
                                                  ----------------------->  Enter 0 to terminate the program  <-----------------------
```



rotatedClockwiseImage.jpeg

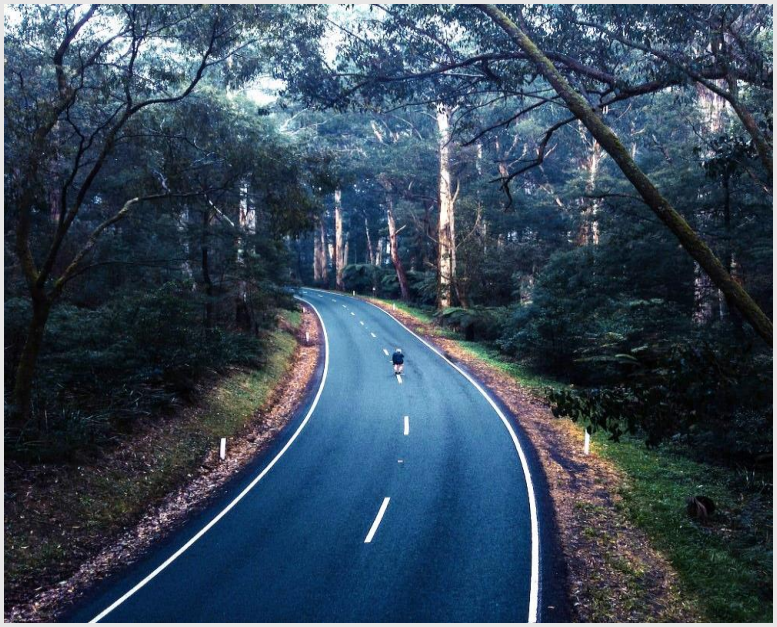
```
                                                  ------------------------>  Enter 1 to run the program  <------------------------
                                                  ------------------------>  Enter 0 to terminate the program  <------------------------
1

====> Provide the file path of the Image :
road.jpeg

====> Select operation to be performed:
(*) Enter 1 for converting image to grayScale.
(*) Enter 2 for rotating image clockwise.
(*) Enter 3 for rotating image anti - clockwise.
(*) Enter 4 for horizontally inverting the image.
(*) Enter 5 for vertically inverting the image.
(*) Enter 6 for changing the brightness of the image.
(*) Enter 7 for blurring the image.
3

Enter degree by which image has to be rotated (POSITIVE MULTIPLE OF 90) :
90

====> Successfully rotated the image anti - clockwise.

                                                  ------------------------>  Enter 1 to run the program  <------------------------
                                                  ------------------------>  Enter 0 to terminate the program  <------------------------
```

road.jpeg



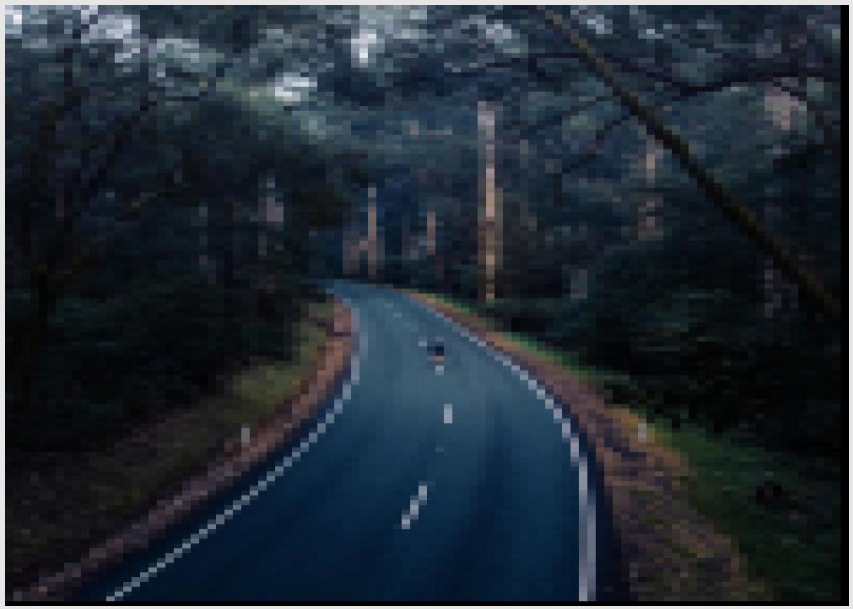
rotatedAntiClockwiseImage.jpeg

```
                                                  ------------------------->  Enter 1 to run the program  <-------------------------
                                                  ----------------------->  Enter 0 to terminate the program  <-----------------------
1

====> Provide the file path of the Image :
road.jpeg

====> Select operation to be performed:
(*) Enter 1 for converting image to grayScale.
(*) Enter 2 for rotating image clockwise.
(*) Enter 3 for rotating image anti - clockwise.
(*) Enter 4 for horizontally inverting the image.
(*) Enter 5 for vertically inverting the image.
(*) Enter 6 for changing the brightness of the image.
(*) Enter 7 for blurring the image.
6

Enter % change in brightness [-100 to 100]:
81

====> Successfully changed the brightness of the image.

                                                  ------------------------->  Enter 1 to run the program  <-------------------------
                                                  ----------------------->  Enter 0 to terminate the program  <-----------------------
```



changedBrightnessImage.jpeg

```
                                                  ------------------------->  Enter 1 to run the program  <-------------------------
                                                  ------------------------->  Enter 0 to terminate the program  <-------------------------
1

====> Provide the file path of the Image :
road.jpeg

====> Select operation to be performed:
(*) Enter 1 for converting image to grayScale.
(*) Enter 2 for rotating image clockwise.
(*) Enter 3 for rotating image anti - clockwise.
(*) Enter 4 for horizontally inverting the image.
(*) Enter 5 for vertically inverting the image.
(*) Enter 6 for changing the brightness of the image.
(*) Enter 7 for blurring the image.
7

Enter blurrness level, any integer from 3 to 40 (3 being least blurry) :
10

====> Successfully blurred the image.

                                                  ------------------------->  Enter 1 to run the program  <-------------------------
                                                  ------------------------->  Enter 0 to terminate the program  <-------------------------
```



blurredImage.jpeg

```
                                        ----------------------->  Enter 1 to run the program  <-----------------------
                                        ----------------------->  Enter 0 to terminate the program  <-----------------------
0


====> Program terminated successfully...
harshkumar@Harshs-MacBook-Air ImageEditor %
```

Terminated successfully.

EDGE CASES:

```
                                        ********************************************************************************
                                                                >  WELCOME TO IMAGE EDITOR  <
                                        ********************************************************************************

                                        ------------------------>  Enter 1 to run the program  <------------------------
                                        ----------------------->  Enter 0 to terminate the program  <-----------------------
9

Exception in thread "main" java.lang.RuntimeException: Please provide a valid input!!!
        at ImageEditor.main(ImageEditor.java:524)
harshkumar@Harshs-MacBook-Air ImageEditor % cd "/Users/harshkumar/Documents/Java/ImageEditor/" && javac ImageEditor.java && java ImageEditor
                                        ********************************************************************************
                                                                >  WELCOME TO IMAGE EDITOR  <
                                        ********************************************************************************

                                        ------------------------>  Enter 1 to run the program  <------------------------
                                        ----------------------->  Enter 0 to terminate the program  <-----------------------
1

====> Provide the file path of the Image :
image
Exception in thread "main" java.lang.RuntimeException: !!!Please provide a valid image format!!!
        at ImageEditor.main(ImageEditor.java:274)
harshkumar@Harshs-MacBook-Air ImageEditor % cd "/Users/harshkumar/Documents/Java/ImageEditor/" && javac ImageEditor.java && java ImageEditor
                                        ********************************************************************************
                                                                >  WELCOME TO IMAGE EDITOR  <
                                        ********************************************************************************

                                        ------------------------>  Enter 1 to run the program  <------------------------
                                        ----------------------->  Enter 0 to terminate the program  <-----------------------
1

====> Provide the file path of the Image :
image.jpg

====> Select operation to be performed:
(*) Enter 1 for converting image to grayScale.
(*) Enter 2 for rotating image clockwise.
(*) Enter 3 for rotating image anti - clockwise.
(*) Enter 4 for horizontally inverting the image.
(*) Enter 5 for vertically inverting the image.
(*) Enter 6 for changing the brightness of the image.
(*) Enter 7 for blurring the image.
9

Exception in thread "main" java.lang.RuntimeException: !!!Please provide a valid operation!!!
        at ImageEditor.main(ImageEditor.java:507)
```

END OF REPORT

THANK YOU

HARSH KUMAR
23BCS10066

BATCH 3