**NAME** : Patwari Venkatesh Nandkumar

**ROLL:** 407B001

**ASSIGNMENT NO. 6**

**CODE:**

import java.awt.\*;

import java.awt.image.BufferedImage;

import java.io.\*;

import javax.imageio.ImageIO;

public class Assignment\_6 {

private static BufferedImage original, answer;

public static void main(String[] args) throws IOException {

// Load the original image from file

File originalFile = new File("flower.jpg");

original = ImageIO.read(originalFile);

// Generate the histogram image (red channel modified)

answer = imageHistogram(original);

// Write the modified image to a new file

writeImage("featureExtraction");

}

// Method to write the image to a file

private static void writeImage(String output) throws IOException {

File file = new File(output + ".jpg");

ImageIO.write(answer, "jpg", file);

}

// Convert color components to an RGB integer

private static int colorToRGB(int alpha, int red, int green, int blue) {

int newPixel = 0;

newPixel += alpha;

newPixel = (newPixel << 8) + red;

newPixel = (newPixel << 8) + green;

newPixel = (newPixel << 8) + blue;

return newPixel;

}

// Method to create a histogram image by modifying the red channel

public static BufferedImage imageHistogram(BufferedImage input) {

BufferedImage redGraph = new BufferedImage(input.getWidth(), input.getHeight(), input.getType());

// Loop through each pixel and adjust color channels

for (int i = 0; i < input.getWidth(); i++) {

for (int j = 0; j < input.getHeight(); j++) {

Color pixelColor = new Color(input.getRGB(i, j));

int alpha = pixelColor.getAlpha();

int red = pixelColor.getRed();

int green = pixelColor.getGreen();

int blue = pixelColor.getBlue();

// Set the new RGB value with the red channel modified

redGraph.setRGB(i, j, colorToRGB(alpha, 0, green, blue));

}

}

return redGraph;

}

}

**OUTPUT:**

****

****