**Lab 12**

**Date : 04-23-2021**

Driver: Ayush Shresth

Scribe: Nachiketh Mamidi

**Step A:**

1. Everything turns green.

For each row

For each column

If (row divided by 3 and remainder is three)

They row is blue

If (row divided by 3 and remainder is one)

The row is red

Else

The row is white

public void makePattern()

{

for (int i = 0; i < ROWS; i++)

{

for (int j = 0; j < COLUMNS; j++)

{

if ( i % 3 == 0)

colors[i][j] = Color.BLUE;

else if ( i % 3 == 1)

colors[i][j] = Color.RED;

else

colors[i][j] = Color.WHITE;

}

}

}

1. Step\_n\_done
2. Step\_n\_done
3. Step\_n\_done

public void makePattern()

{

for (int i = 0; i < ROWS; i++)

{

for (int j = 0; j < COLUMNS; j++)

{

if ( j % 6 == 0 || j % 6 == 1)

colors[i][j] = Color.BLUE;

else if ( j % 6 == 2 || j % 6 == 3)

colors[i][j] = Color.RED;

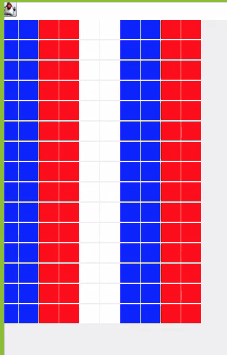
else

colors[i][j] = Color.WHITE;

}

}

}



**Part B:**

1. Step\_n\_done;
2. Step\_n\_done;
3. The image turns into a black and white version of itself.
4. Step\_n\_done;

Number of rows = 169;

Number of columns = 299;

public static void main(String[] args)

{

Picture blimp = new Picture("blimp.jpg");

int[][] grayscale = blimp.getGrayLevels();

for (int row = 0; row < grayscale.length; row ++)

{

for (int col = 0; col < grayscale[0].length; col ++)

{

int c = grayscale[row][col];

grayscale[row][col] = 255 - c;

}

}

Picture grayBlimp = new Picture(grayscale);

grayBlimp.draw();

}

1. The colors got inverted

Part C:

1. Pixel[0][0] goes to pixel[0][100]
2. Top right corner goes to top left.
3. Pixel[i][j] goes to pixel[i][100 - j]
4. Not using temp variable to swap the pixels
5. Step\_n\_done;

public static void main(String[] args)

{

Picture blimp = new Picture("blimp.jpg");

int[][] grayscale = blimp.getGrayLevels();

for (int row = 0; row < grayscale.length; row ++)

{

for (int col = 0; col < grayscale[0].length / 2; col ++)

{

int j2 = grayscale[0].length - col - 1;

int temp = grayscale[row][col];

grayscale[row][col] = grayscale[row][j2];

grayscale[row][j2] = temp;

}

}

Picture grayBlimp = new Picture(grayscale);

grayBlimp.draw();

}

1. It is inverted on x axis. Like a mirror is placed in the middle vertically.

Part D:

1. Step\_n\_done;
2. Pixel[0][0] goes to [0][200]
3. The swapping part is wrong in the pseudocode
4. Step\_n\_done;
5. The image formed is rotated by 90 degree
6. Step\_n\_done;

Picture blimp = new Picture("blimp.jpg");

int[][] grayscale = blimp.getGrayLevels();

int[][] result = new int[grayscale[0].length][grayscale.length];



public class RotateViewer

{

public static void main(String[] args)

{

Picture blimp = new Picture("blimp.jpg");

int[][] grayscale = blimp.getGrayLevels();

int[][] result = new int[grayscale[0].length][grayscale.length];

for(int i = 0; i < result.length; i ++)

{

for(int j = 0; j < result.length; j ++)

{

result[i][j] = grayscale[grayscale.length - j -1][i];

}

}

Picture finPic = new Picture(result);

finPic.draw();

}

}

Part E:

1. Step\_n\_done;
2. Step\_n\_done;
3. An image which is the overlap of first 2 image appears
4. An error occurs, and the image is not printed as the dimensions aren't the same
5. Step\_n\_done;

public class MergeViewer

{

public static void main(String[] args)

{

Picture disney = new Picture("castle.jpg");

Picture dino = new Picture("dinasour.jpg");

int[][] dinoGrey = dino.getGrayLevels();

int [][]disneyGrey = disney.getGrayLevels();

int [][] result = new int[Math.max(dinoGrey.length, castleGrey.length)][ Math.max(dinoGrey[0].length, castleGrey[0].length]

for(int i = 0; i < result.length; i ++)

{

for(int j = 0; j < result.length; j ++)

{

result[i][j] = (dinoGrey[i][j] + disneyGrey[i][j]) / 2;

}

Picture finPic = new Picture(result);

finPic.draw() ;

}

}

}

1. Step\_n\_done;
2. Step\_n\_done;

public class MergeViewer

{

public static void main(String[] args)

{

Picture disney = new Picture("castle.jpg");

Picture dino = new Picture("dinasour.jpg");

int[][] dinoGrey = dino.getGrayLevels();

int [][]disneyGrey = disney.getGrayLevels();

int [][] result = new int[dinoGrey.length][dinoGrey[0].length];

for(int i = 0; i < result.length; i ++)

{

for(int j = 0; j < result.length; j ++)

{

result[i][j] = (dinoGrey[i][j] + disneyGrey[i][j]) / 2;

}

Picture finPic = new Picture(result);

finPic.draw() ;

}

}

}

1. When we run the code now the images which are merged appear without error.