CSE 8A Programming Assignment 7

Name should be formatted as (last, first)
If you are working solo you may leave the right column blank.

Name:	Ma,Tiancheng	Name:	Pan, Bochen
PID:	A15419480	PID:	A15562542
Email:	tima@ucsd.edu	Email:	bpan@ucsd.edu

Part 1 Testing Code:

```
// Copy and paste the code from your test methods (including comments!) here
// Make sure to set the font to Courier New
// IMPORTANT: Make sure your code is properly formatted and commented.
// Code that does not have correct indentation and comments will lose marks.


import java.awt.Color;

public class PA7 {

    // -----------------------------------------------------------------------
    // Tests
    // -----------------------------------------------------------------------

    public static void testCanvas() {
        // Create a blue canvas with width 5 and height 10
        Image canvas1 = new Image(5, 10, Color.blue);

        // Verify the result is a 10x5 blue canvas
        canvas1.explore();

        // create a red canvas with width 10 and height 1000
        Image canvas2 = new Image (10, 1000, Color.red);
        //verify the result is a 1000x10 red canvas
        canvas2.explore();

        //create a red canvas with width 1000 and height 1
        Image canvas3 = new Image (1000,1,Color.green);
        //verify the result is a 1x1000 green canvas
        canvas3.explore();

    }
    public static void testCrop() {
        // Create a 4x4 2D array
        Color[][] test1 = { { Color.black, Color.black, Color.red, Color.red },
// row0
                { Color.black, Color.black, Color.red, Color.red }, // row1
                { Color.black, Color.black, Color.red, Color.red }, // row2
                { Color.black, Color.black, Color.black, Color.black } // row3
        };

        // Create an image from the array and crop the bottom right corner
        Image img1 = new Image(test1);
        Image cropped1 = img1.crop(2, 0, 4, 3);
        // Visualize the cropped picture
        cropped1.explore();

        // import the heart picture
        Image img2 = new Image("images/pixel-heart.png");
        //create an image from the array and crop the bottom right corner
        Image cropped2 = img2.crop(100, 100, 500, 500);
        // Visualize the cropped picture
        cropped2.explore();

        // Import the grace hopper picture
        Image img3 = new Image("images/grace-hopper.png");
        // Create the image from the array and crop the bottom right corner
        Image cropped3 = img3.crop(1,1,50,50);
        //Visualize the cropped picture
        cropped3.explore();

        // Import the cute cat and dog picture
        Image img4 = new Image("images/Cute_cat_and_dog_3.jpg");
        //Create the image from the array and crop the bottom right corner
        Image cropped4 = img4.crop(0,0,100,100);
        //Visualize the cropped picture
        cropped4.explore();


    }

    public static void testOverlay() {
        // Create a 4x4 2D array
        Color[][] bgTest1 = {
            { Color.black, Color.black, Color.black, Color.black }, // row0
            { Color.black, Color.black, Color.black, Color.black }, // row1
            { Color.black, Color.black, Color.black, Color.black }, // row2
            { Color.black, Color.black, Color.black, Color.black } // row3
        };
        // Create a 3x2 2D array
        Color[][] fgTest1 = { { Color.red, Color.red }, { Color.red, Color.red
}, { Color.red, Color.red } };

        // Create an image from the array and crop the bottom right corner
        Image bgImg1 = new Image(bgTest1);
        Image fgImg1 = new Image(fgTest1);
        Image overlayed1 = fgImg1.overlay(bgImg1, 1, 1);

        // Visualize the cropped picture
        overlayed1.explore();

        //Import the background picture
        Image bgImg2 = new Image ("images/crane.jpg");
        //Import the foreground picture
        Image fgImg2 = new Image ("images/checker5.bmp");
        //Create an overlay image
        Image overlayed2 = fgImg2.overlay(bgImg2,100,100);
        // Visualize the cropped picture
        overlayed2.explore();

        //Import the background picture
        Image bgImg3 = new Image ("images/pixel-heart.png");
        //Import the foreground picture
        Image fgImg3 = new Image ("images/purple5.bmp");
        //Create an overlay image
        Image overlayed3 = fgImg3.overlay(bgImg3,200,200);
        // Visualize the cropped picture
        overlayed3.explore();


    }

    public static void testChromakey() {
        // Create a 4x4 2D array
        Color[][] bgTest1 = {
            { Color.red, Color.red, Color.red, Color.red }, // row0
            { Color.red, Color.red, Color.red, Color.red }, // row1
            { Color.red, Color.red, Color.red, Color.red }, // row2
            { Color.red, Color.red, Color.red, Color.red } // row3
        };

        // Create a 3x2 2D array
        Color[][] fgTest1 = {
            { Color.green, Color.green, Color.black, Color.black }, // row0
            { Color.green, Color.green, Color.black, Color.black }, // row1
            { Color.green, Color.green, Color.black, Color.black }, // row2
            { Color.green, Color.green, Color.black, Color.black } // row3
        };

        // Create an image from the array and crop the bottom right corner
        Image bgImg1 = new Image(bgTest1);
        Image fgImg1 = new Image(fgTest1);
        Image chromakeyed1 = fgImg1.chromakey(bgImg1, Color.green, 1);

        // Visualize the cropped picture
        chromakeyed1.explore();

        // Create a foreground image
        Image fgImg2 = new Image ("images/crane.jpg");
        //crop the foreground image to make it sit into the size of background
        //image
        Image croppedfg2 = fgImg2.crop(300,300,500,500);
        //Create a background image
        Image bgImg2 = new Image ("images/checker200.bmp");
        //Create a chromakeyed image
        Image chromakeyed2 = croppedfg2.chromakey(bgImg2, Color.green, 30);
        //Visualize the cropped picture
        chromakeyed2.explore();

        //// Create a foreground image
        Image fgImg3 = new Image ("images/grace-hopper.png");
        //crop the foreground image to make it sit into the size of background
        //image
        Image croppedfg3 = fgImg3.crop(0,0,200,232);
        //Create a background image
        Image bgImg3 = new Image("images/pixel-heart.png");
        //Create a chromakeyed image
        Image chromakeyed3 = croppedfg3.chromakey(bgImg3, Color.white, 30);
        //Visualize the cropped picture
        chromakeyed3.explore();


    }

    public static void testFlipHorizontal() {
        // Create 4x4 2D array
        Color[][] test1 = { { Color.black, Color.black, Color.black,
Color.black }, // row0
                { Color.black, Color.black, Color.black, Color.black }, // row1
                { Color.red, Color.red, Color.red, Color.red }, // row2
                { Color.red, Color.red, Color.red, Color.red } // row3
        };

        // First visualize the original image
        Image img1 = new Image(test1);
        img1.explore();

        // Flip the image and visualize the result
        Image flippedImg1 = img1.flipHorizontal();
        flippedImg1.explore();

        //Import the heart picture
        Image img2 = new Image ("images/pixel-heart.png");
        //Visualize the picture first
        img2.explore();
        // Flip the image and visualize the result
        Image flippedImg2 = img2.flipHorizontal();
        //Visualize the flipped picture
        flippedImg2.explore();

        //Import the grace hopper picture
        Image img3 = new Image ("images/grace-hopper.png");
        //Visualize the picture first
        img3.explore();
        // Flip the image and visualize the result
        Image flippedImg3 = img3.flipHorizontal();
        //Visualize the flipped picture
        flippedImg3.explore();




    }

    // ------------------------------------------------------------------------
    // Main Method
    // ------------------------------------------------------------------------

    public static void main(String[] args) {
        // You may want to uncomment one test at a time
        // NOTE: testCanvas will error unless the canvas constructor is
implemented
        // please implement the canvas constructor before uncommenting that
line.

        testCanvas();
        testCrop();
        testOverlay();
        testChromakey();
        testFlipHorizontal();

        // TODO: Add code for Part 2 here
    }
```

```
}
```

Part 1 Explanation:

Briefly explain why you chose each test and how you know your code is working correctly (or not).

1. For the canvas part, I pick the different color and different width and height to test whether the code can work correctly. As a result, the code worked correctly.
2. For the cropped part, I pick two different pictures for testing in order to test the code. Thus, I can conclude that the code can work for all pictures.
3. For the overlay part, I picked two different foregrounds and backgrounds for testing. In order to test correctly, I decided to choose the pictures with large size as backgrounds and pictures with smaller size as foregrounds.
4. For the chromakey part, I choose different backgrounds and foregrounds for testing. Since the prerequisite for chromakey is that two pictures must be the same size, I cropped the pictures first to make the background and foreground the same size.
5. For the flip horizontal part, I choose the different pictures for testing whether the flipping can be tested correctly in different pictures.

Part 2: Collage Code

```
// Copy and paste the code from your main method (including comments!) here
// Make sure to set the font to Courier New
// IMPORTANT: Make sure your code is properly formatted and commented.
// Code that does not have correct indentation and comments will lose marks.
```

Additional code of instance method in Image.java :

```
public class Image {
……
……
public Image negative(){
       Color[][] newArray = new Color[this.getHeight()][this.getWidth()];
       for (int row = 0;row < this.getHeight();row++){
         for (int col = 0;col < this.getWidth();col++){
           Color x = this.pixels[row][col];
           int RED = x.getRed();
           int GREEN = x.getGreen();
           int BLUE = x.getBlue();
           newArray[row][col] = new Color(255-RED,255-GREEN,255-BLUE);
         }
       }
       return new Image(newArray);
     }


public Image contrast(){
       Color[][] newArray = new Color[this.getHeight()][this.getWidth()];
       for (int row = 0;row < this.getHeight();row++){
         for (int col = 0;col < this.getWidth();col++){
           Color x = this.pixels[row][col];
           int RED = x.getRed();
           int GREEN = x.getGreen();
           int BLUE = x.getBlue();
           if (RED<127){
           newArray[row][col] = new Color(2*RED,GREEN/2,BLUE/2);
           }else{
             newArray[row][col] = new Color(255,GREEN/4,BLUE/4);
           }
         }
       }
       return new Image(newArray);
     }
}
```

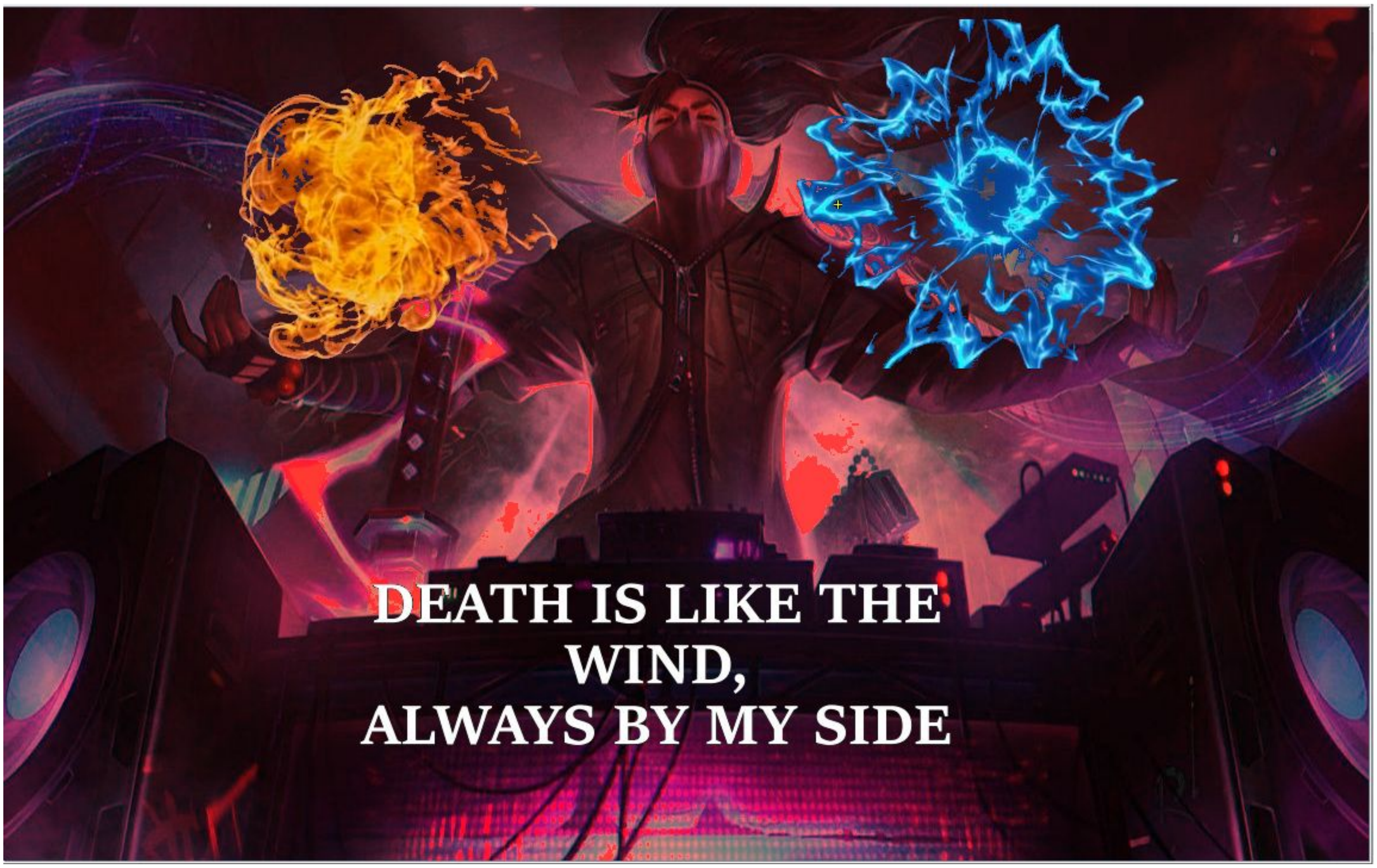
CODE IN PA7.java:

```
public class PA7 {
……
……
public static void newimg(){
    Image bgImg = new Image("images/yasuo.jpg");
    Image bgImg0 = bgImg.contrast();
    //Call contrast on bgImg and create new image bgImg0.

    Image canvas = new Image(1146, 716, Color.black);
    //create a canvas for later use where 1146*716 is the size of bgImg0 and
    //black is the background color of the foreground image.
    //(mainly for chromakey which requires same size of foreground image and
    //background image)

    Image fgImg0 = new Image("images/fire1.jpg");
    //first foreground image

    Image overlayed0 = fgImg0.overlay(canvas, 100, 0);
    //call overlay and adjust the position. Here we get a 1146*716 “fire” image
    //placed at [0][100] whose background color is black.

    Image chromakeyed0 = overlayed0.chromakey(bgImg0, Color.black, 30);
    //get chromakeyed image.

    Image fgImg1 = new Image("images/thunder4.jpg");//second foreground image.

    Image overlayed1 = fgImg1.overlay(canvas, 650, 10);
    Image chromakeyed1 = overlayed1.chromakey(chromakeyed0, Color.black, 25);
    Image caption = new Image("images/captions1.jpg");//third foreground image
    Image capConvert = caption.negative();//call negative on caption.
    Image capCrop = capConvert.crop(100,0,1100,156);//crop it for later use.
    Image overlayed2 = capCrop.overlay(canvas, 40, 480);
    Image chromakeyed2 = overlayed2.chromakey(chromakeyed1, Color.black, 1);
    //here we get final piece.

    chromakeyed2.explore();//visualize it.
}
public static void main(String[] args) {
newimg();
}
}
```

Part 2: Collage Image

Show us your final collage. Make sure that this image is produced by your main method as shown above. Then describe how you used each of the methods you wrote in Part 1 for your collage.

Final collage:

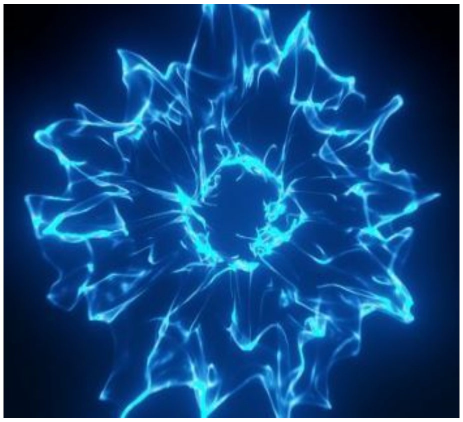


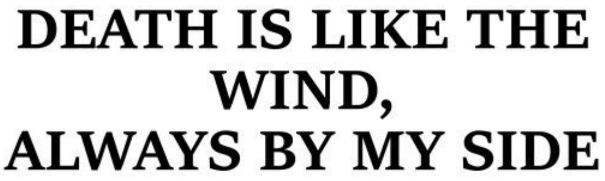
We used 4 images from files, one constructor, 3 methods and 2 additional method from PA6 recreated.

1. *The background image: we used one additional method* ***<u>contrast()</u>****, to change color of it to make it seems red overall.*
2. *Fireball and thunderball: we find two original images of them both with black background, then we create canvas of color black and size 1146*716 (which is the size of background image in 1) by calling* ***<u>Image(int widthIn, int heightIn, Color color)</u>****. Then we call* ***<u>overlay(Image bg, int topLeftX, int topLeftY)</u>*** *to overlay canvas and our two original images, where we can adjust the position of fireball*

*and thunderball by changing the value of topLeftX and topLeftY. Finally, we call **chromakey(Image bg, Color key, double threshold)** to put fireball and thunderball onto the background image, and we can change the effect of balls by changing the value of threshold.*

3. *For the caption: we call another additional method **negative()** to change what it looks like and make its background color black, then we call **crop(int topLeftX, int topLeftY, int bottomRightX, int bottomRightY)** to change the size of original image because the original size is too big. Then we use overlay on cropped image and image we already created from 2 to create the final image.*
4. *HERE ARE ORIGINAL PICTURES:*





DEATH IS LIKE THE
WIND,
ALWAYS BY MY SIDE

Known Bugs or Issues:

If you have known bugs or issues with your code, let us know here. If you think it works correctly, justify why.

1. All pictures must not be out of the ranges.
2. Since the prerequisite for chromakey is that two pictures must be the same size, we have to crop the pictures first to make the background and foreground the same size.
3. Don’t forget to fill out the reflection form.
4. We want to use method from PA6 but we need to rewrite them into instance method in PA7.