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:	Darren Yeung	Name:	
PID:	A15943292	PID:	
Email:	dyeung@ucsd.edu	Email:	

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.
```

Canvas Test

```
    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();

        // TODO: Add two more test cases


        Image canvas2 = new Image(1,1, Color.green);
        canvas2.explore();

        Image canvas3 = new Image(5,5, Color.black);
        canvas3.explore();
    }
```

CropTest

```
    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();
        Image img2 = new Image("images/pixel-heart.png");
        Image cropped2 = img2.crop(100, 100, 500, 500);

        cropped2.explore();


        Image image3 = new Image("images/crane.jpg");
        Image imagecropped2 = image3.crop(5,5, 6,6);
        imagecropped2.explore();

        // TODO: Add two more test cases
    }


Overlaytest

  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
        fgImg1.explore();
        overlayed1.explore();
```

```
        Image backgroundimage2 = new Image("images/crane.jpg");
        Image actualimage2 = new Image("images/pixel-heart.png");

        Image overlayed2 = actualimage2.overlay(backgroundimage2,20,20);

        overlayed2.explore();


       Color[][] backgroundtest3 =
   {{Color.black,Color.black},{Color.black,Color.black},{Color.black,Color.black
   },{Color.black,Color.black}};
       Color[][] actualtest3 = {{Color.red,Color.red},{Color.red,Color.red}};

       Image backgroundtest33 = new Image(backgroundtest3);
       Image actualtest33 = new Image(actualtest3);


       Image overlayed3 = actualtest33.overlay(backgroundtest33,0,0);

       overlayed3.explore();







        // TODO: Add two more test cases
    }



TesChroma


    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);
    //if foreground is close enough to green, replace it with whatever is
in background

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

    Color[][] backgroundtest2 = {
      {Color.blue, Color.blue,Color.blue,Color.blue},
      {Color.blue, Color.blue,Color.blue,Color.blue},
      {Color.blue, Color.blue,Color.blue,Color.blue},
      {Color.blue, Color.blue,Color.blue,Color.blue}
    };

    Color[][] foregroundtest2 = {

      {Color.pink,Color.pink,Color.pink,Color.pink},
      {Color.pink,Color.pink,Color.pink,Color.pink},
      {Color.pink,Color.pink,Color.pink,Color.pink},
      {Color.pink,Color.pink,Color.pink,Color.pink}

    };

    Image backgroundimage2 = new Image(backgroundtest2);
    Image foregrounddimage2 = new Image(foregroundtest2);

    Image chorma2 = foregrounddimage2.chromakey(backgroundimage2,
Color.pink,1);
    chorma2.explore();




    Color[][] backgroundtest3 = {

      {Color.pink,Color.pink,Color.pink,Color.pink},
      {Color.pink,Color.pink,Color.pink,Color.pink},
      {Color.pink,Color.pink,Color.pink,Color.pink},
      {Color.pink,Color.pink,Color.pink,Color.pink}
```

```
        };

        Color[][] foregroundtest3 = {

          {Color.green,Color.green,Color.green,Color.green},
          {Color.green,Color.green,Color.green,Color.green},
          {Color.green,Color.green,Color.green,Color.green},
          {Color.green,Color.green,Color.green,Color.green}

        };

        Image backgroundimage3 = new Image(backgroundtest3);
        Image foregroundimage3 = new Image(foregroundtest3);

        Image chroma3 = foregroundimage3.chromakey(backgroundimage3,
   Color.green,1);

        chroma3.explore();



        // TODO: Add two more test cases
    }
```

Flip test

```
    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();
```

```
        Color[][] test2 = {

          {Color.pink, Color.pink},
          {Color.red, Color.red}


        };

        Image image2 = new Image(test2);

        Image flipped2 = image2.flipHorizontal();

        flipped2.explore();

        Image image3 = new Image("images/crane.jpg");

        Image flipped3 = image3.flipHorizontal();

        flipped3.explore();



        // TODO: Add two more test cases
    }
```

Part 1 Explanation:
Briefly explain why you chose each test and how you know your code is working correctly (or not).

Canvas test 1
I created a 1 by 1 image that is color green. It works because when I explore it, it shows only one green pixel. I chose this test because I wanted to see if one pixel works .

Canvas test 2
I created a 5 by 5 image that was all black and I knew it works because when I explored it, it showed me and image that was 5x5 all black. I chose this test because I wanted to test the extreme rbg values and also what if all the pixels were the same color.

Croptest 1
I chose this test because I wanted to see if my method can work on images from files. I know this works because it made the heart image smaller and the size of the cropped image corresponds with the arguments I passed in to my crop method when I explored the image.

Croptest 2
I chose this test because I wanted to see if my method works if I wanted to crop the image to one pixel. I know it works because when I cropped the crane pixel at 5,5,6,6, it returned only one pixel which was what I was looking for.

Overlay test 1
I chose this test because I wanted to see if my method can work on image from files. I chose the crane as my background and the heart as my foreground. It works because when I explored the image, the foreground image showed up on top of the background image at the correct starting position and ending position and pixels.

Overlay test 2
I chose this test because I wanted to see if my method can actually work if I created the 2d array filled with colors myself. I know this method works because my background image was a 2d array 4x2 of all black pixels. My foreground image was a 2x2 of all red. When I overlayed my foreground image on top of my background image at 0,0, the top two rows became red while leaving the bottom two rows black.

Chromakey test 1
I chose this test because I wanted to see if my method works when I made every single pixel the same color. I know this method works because when I called my method, it returned an image that was all blue. This is so because i chose pink as my key and if a color is close enough to pink, it changes the original pixel to the pixel in the corresponding background image which was blue. And since every color was pink and every color in the background was blue, I got a full blue image.

Chromatest 2
I chose this test because I wanted ot see if my method works with different colors. I chose my background as full pink, my foreground as full green, and my key as green. I know this method works because if a color is close enough to green, my method should change it to pink. And since every pixel in my original is green and every pixel in the background is pink, I should get a full pink image. That was what I got when I explored my image.

Fliptest1
I chose this test because I wanted to see if my method works if I created the image manually by creating a 2d array. It works because I created a 2x2 array of the ifrst row pink and the second row red. When I called the method, the first row became red and the second row became pink which is how I know it got flipped.

Flip Test 2
I chose this test because I wanted to see if my method works if I passed in a image from files. I know this method works because I used the crane image as my parameter and when I explored the adjusted image, the crane was turned upside down.

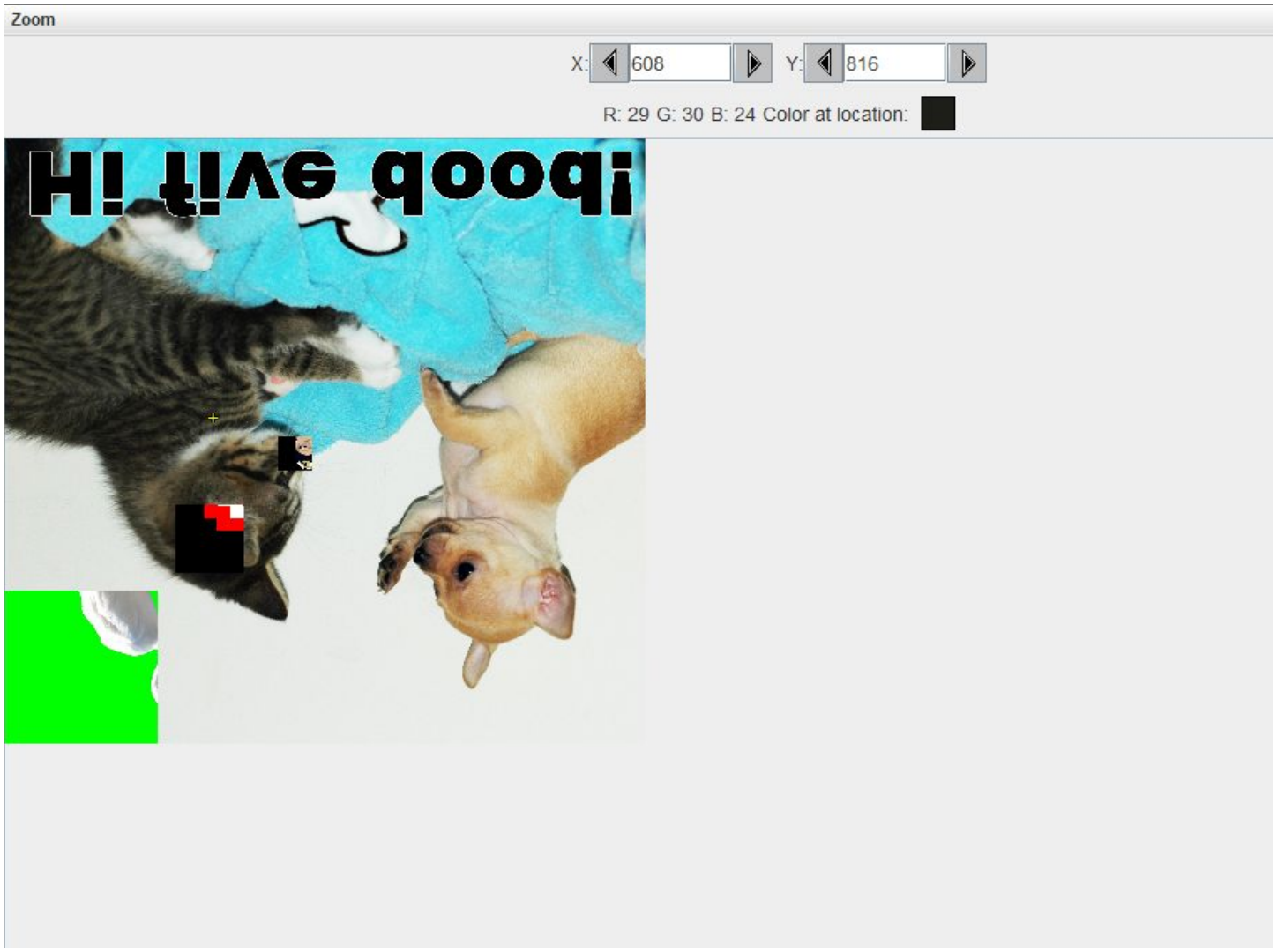
Part 2: Collage Code

```
// 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.

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();

        Image dog = new Image("images/Cute_cat_and_dog_3.jpg");


        Image crane = new Image("images/crane.jpg");
        Image croppedcrane = crane.crop(0,0,450,450);


        Image heart = new Image("images/pixel-heart.png");
        Image croppedheart = heart.crop(0,0,200,200);


        Image person = new Image("images/grace-hopper.png");
        Image croppedperson = person.crop(0,0,100,100);


        Image firstpass = croppedcrane.overlay(dog,0,0);
        Image secondpass = croppedheart.overlay(firstpass,500,500);

        Image thirdpass = croppedperson.overlay(secondpass, 800,800);

        Image finalpass = thirdpass.flipHorizontal();
```

```
        finalpass.explore();
        // TODO: Add code for Part 2 here
    }
```

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.



First, I cropped the crane, the heart, and the image with a man. Then I overlayed the three cropped images over the big image with cat and dog. After creating that image, I flipped the image and made that my final image and explored that final image.

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.

I have no known bugs or issues with my code. Everything works accordingly. The only thing that would stop it from working is if someone passed in out of bounds parameters for methods such as overlaying at a position that does not exist on a image. My code works perfectly because I got full points on the autograder and I made sure that for all methods, the methods started where it was supposed to. So for example, in the overlay method, I checked manually that the overlay did indeed start at topleftx and toplefty and continued down from there in the actual tests.