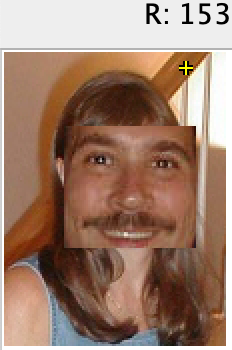
Name:   
Date:

**16.06 Picture Lab Worksheet**

**Directions**: Make note of your responses to the following questions as you work through the activities and exercise in the lesson.

**Activity 8 Exercise Results**

1. Paste the image that is the result of calling the second copy method that adds parameters to allow you to copy just part of the fromPic.



public void copy(Picture fromPic,

int fromStartRow,

int fromStartCol,

int fromEndRow,

int fromEndCol,

int toStartRow,

int toStartCol)

{

Pixel fromPixel = null;

Pixel toPixel = null;

Pixel[][] toPixels = this.getPixels2D();

Pixel[][] fromPixels = fromPic.getPixels2D();

for (int fromRow = fromStartRow, toRow = toStartRow;

fromRow < fromEndRow &&

toRow < toPixels.length;

fromRow++, toRow++)

{

for (int fromCol = fromStartCol, toCol = toStartCol;

fromCol < fromEndCol &&

toCol < toPixels[0].length;

fromCol++, toCol++)

{

fromPixel = fromPixels[fromRow][fromCol];

toPixel = toPixels[toRow][toCol];

toPixel.setColor(fromPixel.getColor());

}

}

}

1. Paste the image that is the result of calling the myCollage method.



public void myCollage()

{

Picture mark = new Picture("blue-mark.jpg");

this.copy(mark,211,300,272,366,231,363);

Picture mark1 = new Picture(mark);

mark1.keepOnlyBlue();

this.copy(mark1,211,300,272,366,111,0);

Picture mark2 = new Picture(mark);

mark2.mirrorHorizontal();

this.copy(mark2,211,300,272,366,289,100);

Picture flower1 = new Picture("flower1.jpg");

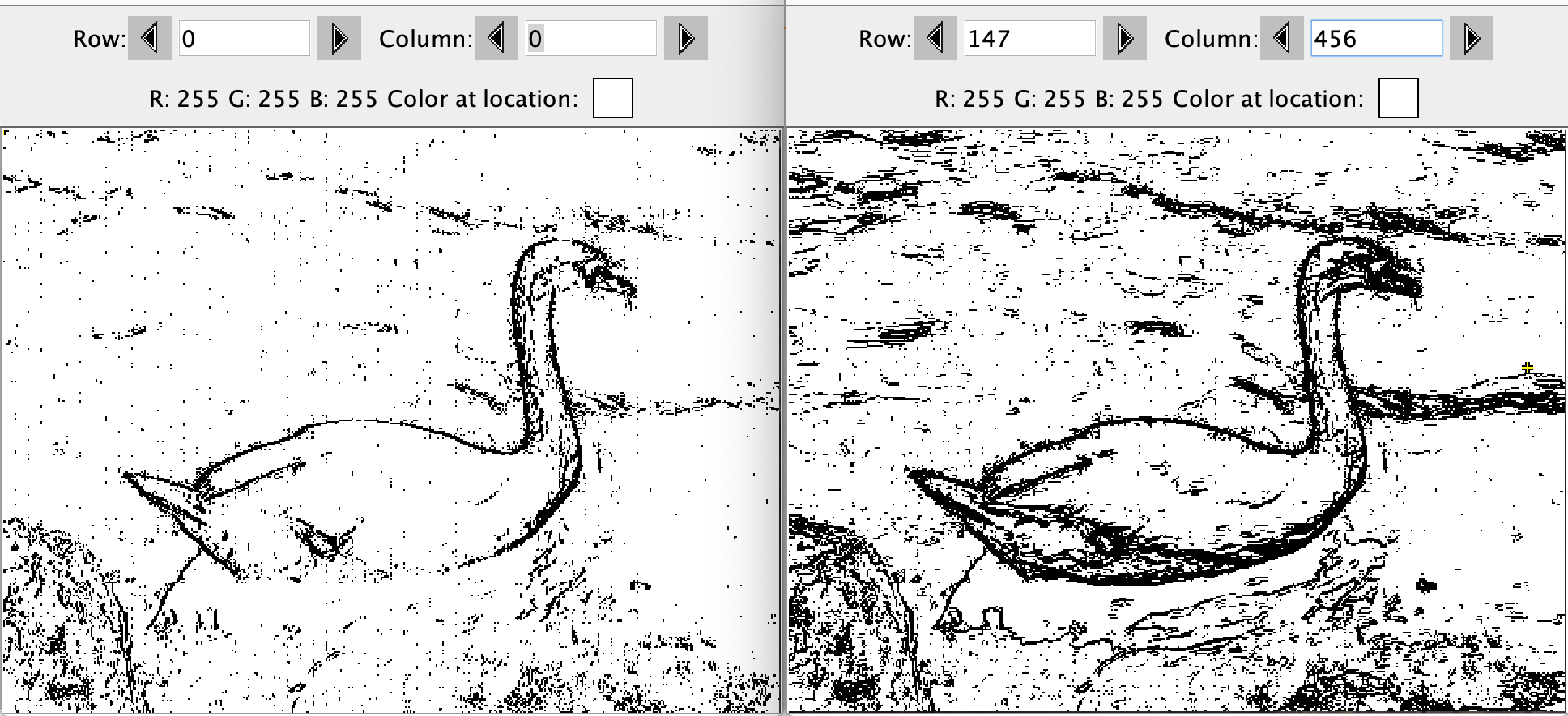
flower1.grayscale();

this.copy(flower1,0,24,65,78,160,79);

}

**Activity 9 Exercise Results**

1. Paste the image that is the result of calling the edgeDetection2 method.

public void edgeDetection2(int edgeDist)

{

Pixel leftPixel = null;

Pixel rightPixel = null;

Pixel downPixel = null;

Pixel[][] pixels = this.getPixels2D();

Color rightColor = null;

Color downColor = null;

for (int row = 0; row < pixels.length-1; row++)

{

for (int col = 0; col < pixels[0].length-1; col++)

{

leftPixel = pixels[row][col];

rightPixel = pixels[row][col+1];

downPixel = pixels[row+1][col];

rightColor = rightPixel.getColor();

downColor = downPixel.getColor(); Left is edgeDetection(), Right is edgeDetection2()

if ((leftPixel.colorDistance(rightColor) > edgeDist) || (leftPixel.colorDistance(downColor) > edgeDist)){

leftPixel.setColor(Color.BLACK);

}

else{

leftPixel.setColor(Color.WHITE);

}

}

}

}

1. Think about and propose another algorithm for edge detection.

Another algorithm would be seeing if pixels above also have a distance greater than edge distance. this would define edges even more.