← Overview of Programming Concepts

Quiz, 5 questions

✖ Try again once you are ready.

Required to pass: 80% or higher

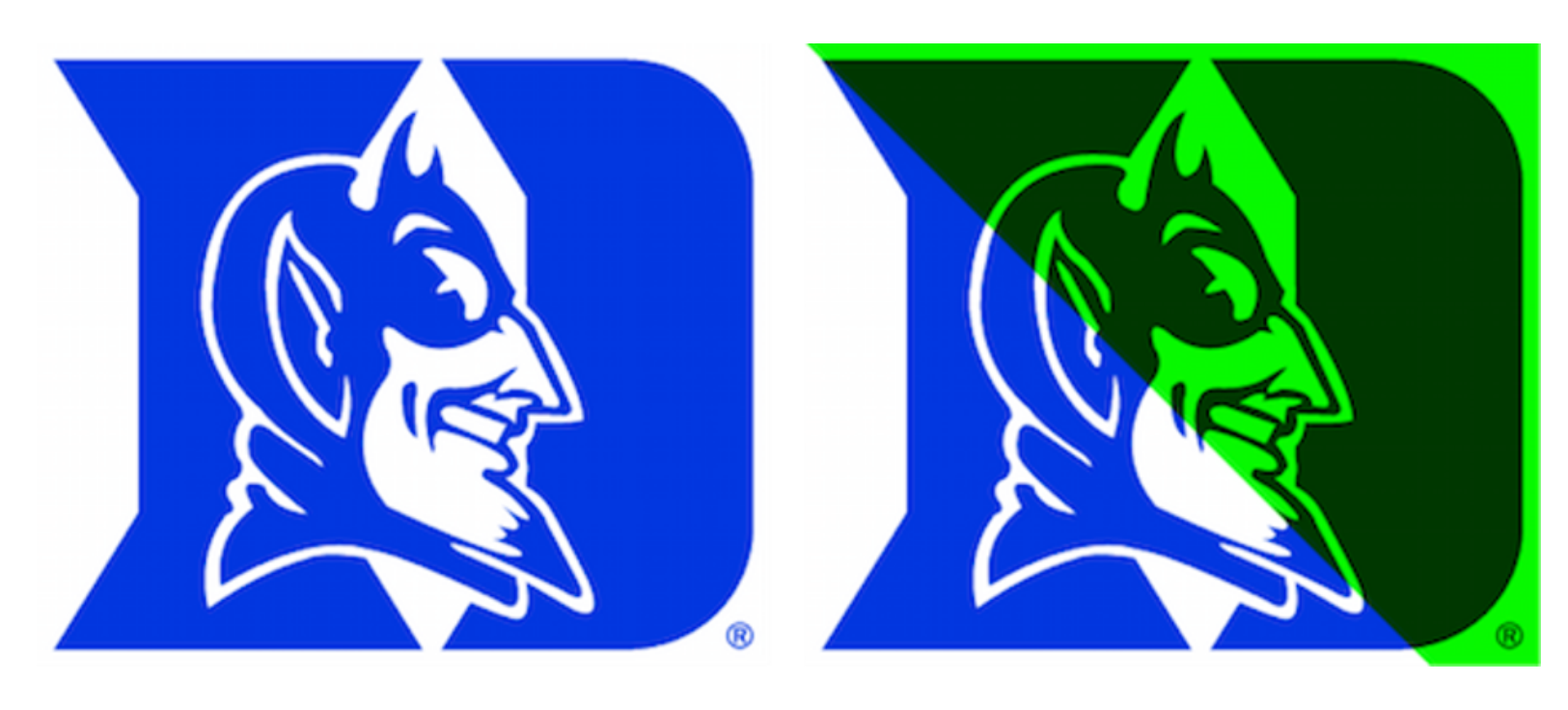
You can retake this quiz up to 3 times every 8 hours.

Back to Week 2

Retake

✔ 1/1 points

1. Consider the following image on the left, which has been modified into the image on the right with green by changing the red and blue values of some pixels to 0.



Which one of the following is most likely the code that modifies the first image to look like the second image?

Hint: be sure to review how image x and y coordinates work. You can review this on our documentation page.

(●)
```
for (var pixel of image.values()) {
  x = pixel.getX();
  y = pixel.getY();
  if (x > y) {
    pixel.setRed(0);
    pixel.setBlue(0);
  }
}
```

Correct
Correct!

( )
```
for (var pixel of image.values()) {
  x = pixel.getX();
  y = pixel.getY();
  if (x < y) {
    pixel.setRed(0);
    pixel.setBlue(0);
  }
}
```

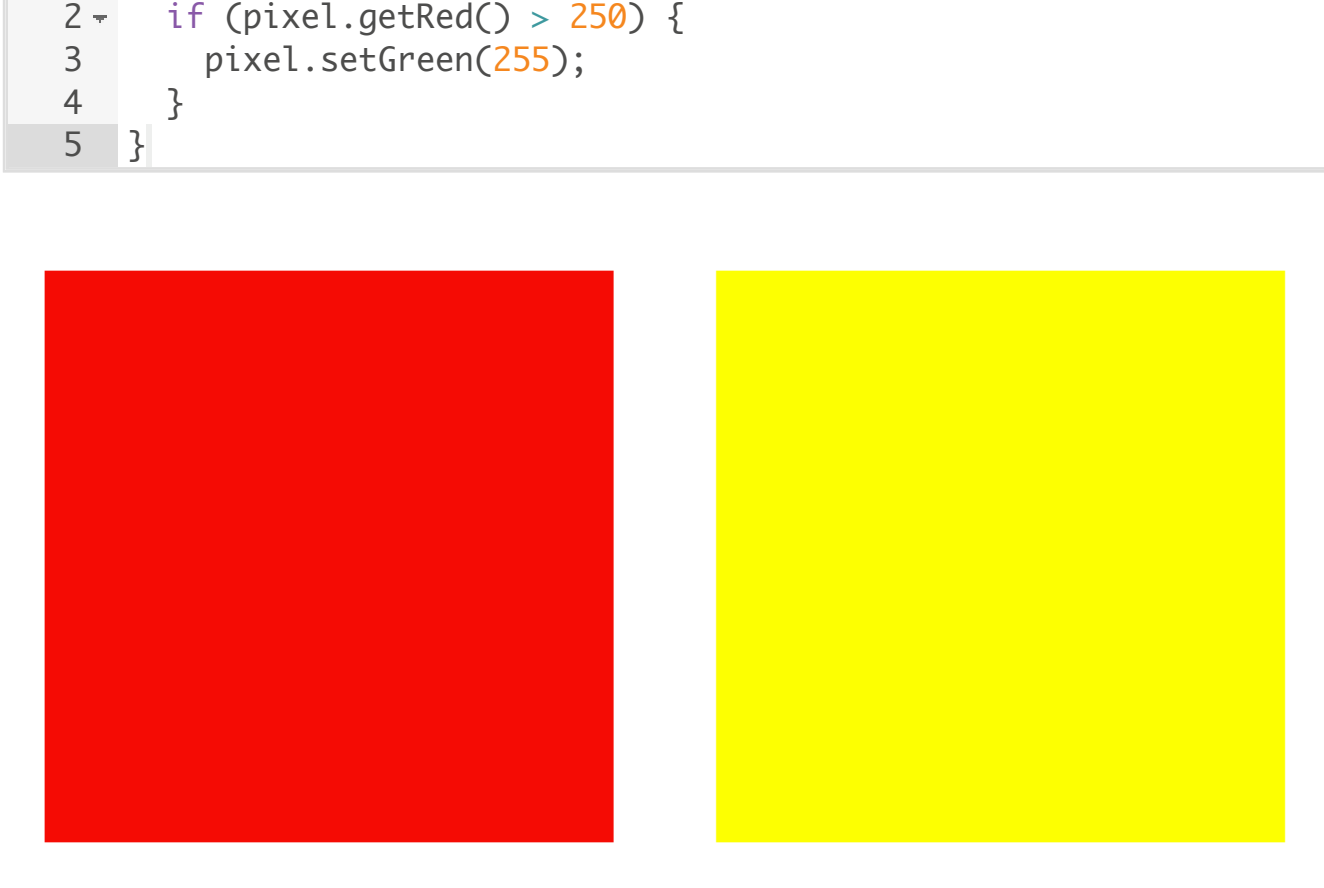
( )
```
w = image.getWidth();
for (var pixel of image.values()) {
  x = pixel.getX();
  y = pixel.getY();
  if (x + y < w/2) {
    pixel.setRed(0);
    pixel.setBlue(0);
  }
}
```

( )
```
w = image.getWidth();
for (var pixel of image.values()) {
  x = pixel.getX();
  y = pixel.getY();
  if (x + y > w/2) {
    pixel.setRed(0);
    pixel.setBlue(0);
  }
}
```

✔ 1/1 points

2. Consider the following code in which the starting image named image is all red (each pixel has red value 255, green value 0 and blue value 0) as shown below on the left and the resulting image shown on the right below is supposed to be all green, but is all yellow. The image is a 200 pixel by 200 pixel image.

```
for (var pixel of image.values()) {
  if (pixel.getRed() > 250) {
    pixel.setGreen(255);
  }
}
```



Which one of the following correctly identifies a statement or statements that should be added to the body of the if statement so that the red square turns into a green square when the code executes?

(●)
```
pixel.setRed(0);
```

Correct
Correct!

( )
```
pixel.setRed(255);
```

( )
```
pixel.setRed(255);
pixel.setBlue(0);
```

( )
```
pixel.setRed(255);
pixel.setBlue(255);
```

0.20 / 1 points

3. Recall the function **addBorder** you wrote in a programming exercise that has a parameter **image** and another parameter **thickness**. This function returns image with an added black border around each side of the image that is thickness pixels wide.

Which <u>two</u> of the following are correct implementations of addBorder?

[■]
```
function addBorder(image, thickness){
    for (var px of image.values()){
        if (px.getX() < thickness){
            px = setBlack(px);
        }
        if (px.getX() > image.getWidth()-thickness){
            px = setBlack(px);
        }
        if (px.getY() < thickness){
            px = setBlack(px);
        }
        else{
            px = setBlack(px);
        }
    }
    return image;
}
```

This should not be selected
Go back to the **Advanced Modifying Images** programming exercise and review your code and the problem. If you have trouble completing this exercise, ask for help in the forums.

[ ]
```
function addBorder(image, thickness){
    for (var px of image.values()){
        if (px.getX() < width){
            px = setBlack(px);
        }
        if (px.getX() > image.getWidth()-width){
            px = setBlack(px);
        }
        if (px.getY() < height){
            px = setBlack(px);
        }
        if (px.getY() > image.getHeight()-height){
            px = setBlack(px);
        }
    }
    return image;
}
```

Un-selected is correct

[ ]
```
function addBorder(image, thickness){
  for (var px of image.values()){
    var x = px.getX();
    var y = px.getY();
    if (x < thickness){
      px = setBlack(px);
    }
    if (x >= image.getWidth()-thickness){
      px = setBlack(px);
    }
    if (y < thickness){
      px = setBlack(px);
    }
    if (y >= image.getHeight()-thickness){
      px = setBlack(px);
    }
  }
  return image;
}
```

This should be selected

[■]
```
addBorder(image, thickness){
    for (var px of image.values()){
        if (px.getX() < thickness){
            px = setBlack(px);
        }
        if (px.getX() > image.getWidth()-thickness){
            px = setBlack(px);
        }
        if (px.getY() < thickness){
            px = setBlack(px);
        }
        if (px.getY() > image.getHeight()-thickness){
            px = setBlack(px);
        }
    }
    return image;
}
```

This should not be selected
Go back to the **Advanced Modifying Images** programming exercise and review your code and the problem. If you have trouble completing this exercise, ask for help in the forums.

[ ]
```
function addBorder(image, thickness){
    for (var pixel of image.values()){
        if (pixel.getX() < thickness){
            pixel = setBlack(pixel);
        }
        if (pixel.getX() >= image.getWidth()-thickness){
            pixel = setBlack(pixel);
        }
        if (pixel.getY() < thickness){
            pixel = setBlack(pixel);
        }
        if (pixel.getY() >= image.getHeight()-thickness){
            pixel = setBlack(pixel);
        }
    }
    return image;
}
```

This should be selected

✔ 1/1 points

4. Consider the following program that uses the setBlack function you wrote in the **Advanced Modifying Images** programming exercise:

```
function pixelOnEdge(image,pixel,horizontalThick, verticalThick){
    var x = pixel.getX();
    var y = pixel.getY();
    if (x < verticalThick || x > image.getWidth() - verticalThick){
        return true;
    }
    if (y < horizontalThick || y > image.getHeight() - horizontalThick){
        return true;
    }
    return false;
}

function addBorders(image,horizontalThick, verticalThick){
    for (var px of image.values()){
        if (pixelOnEdge(image,px,horizontalThick,verticalThick)){
            px = setBlack(px);
        }
    }
    return image;
}

var img = new SimpleImage("skyline.png");
img = addBorders(img,40,20);
print(img);
```

What is the best description of the purpose of the pixelOnEdge function?

( ) To identify pixels within the horizontal borders

( ) To color pixels that are within the borders black

(●) To identify pixels that are within the borders by returning true

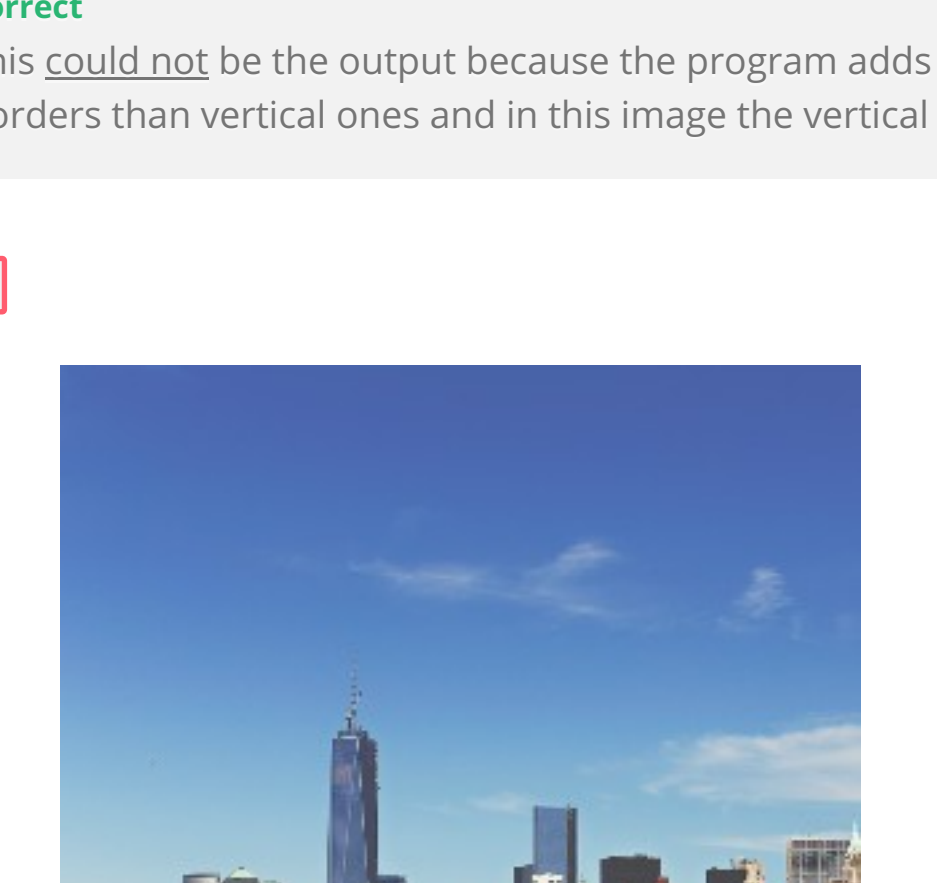
Correct
Correct!

( ) To identify pixels within the vertical borders

0.40 / 1 points

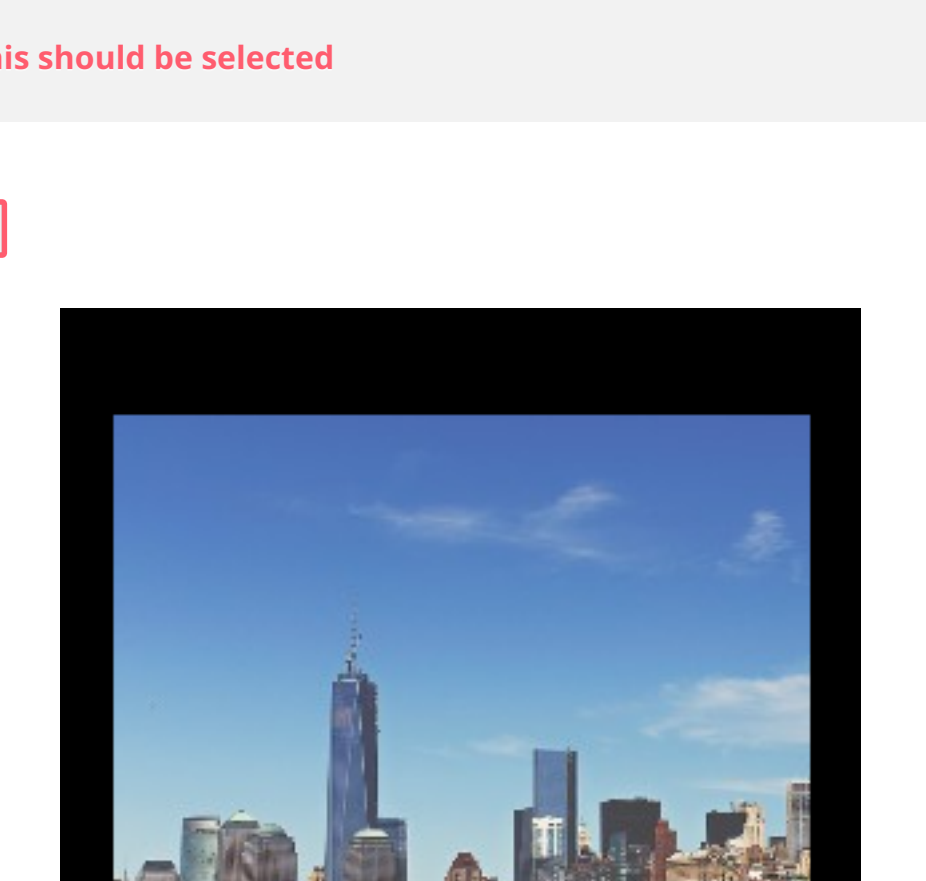
5. Which of the following could <u>not</u> be the output of running the program written in the previous question? Select all that apply.

[■]



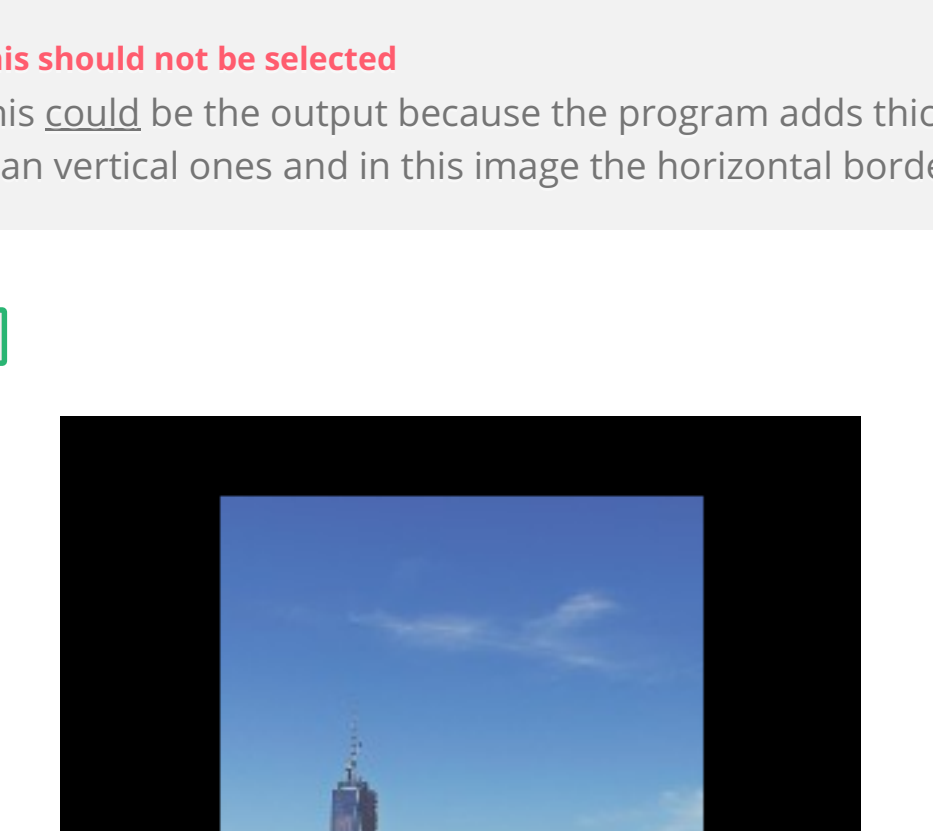
Correct
This <u>could not</u> be the output because the program adds thicker horizontal borders than vertical ones and in this image the vertical borders are thicker.

[ ]



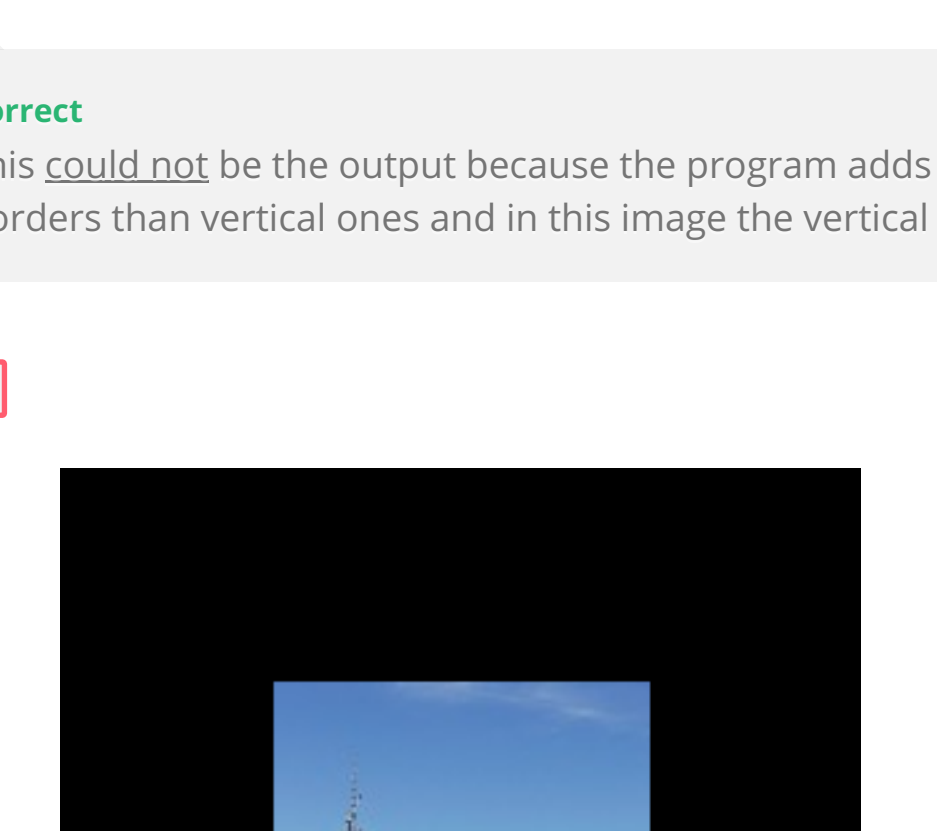
This should be selected

[■]



This should not be selected
This <u>could</u> be the output because the program adds thicker horizontal borders than vertical ones and in this image the horizontal borders are thicker.

[■]



Correct
This <u>could not</u> be the output because the program adds thicker horizontal borders than vertical ones and in this image the vertical borders are thicker.

[ ]

This should be selected