


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.

- (selected)
```
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);
  }
}
```

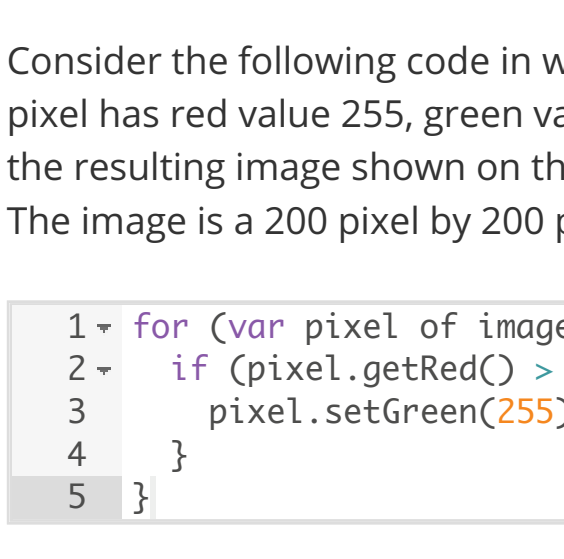
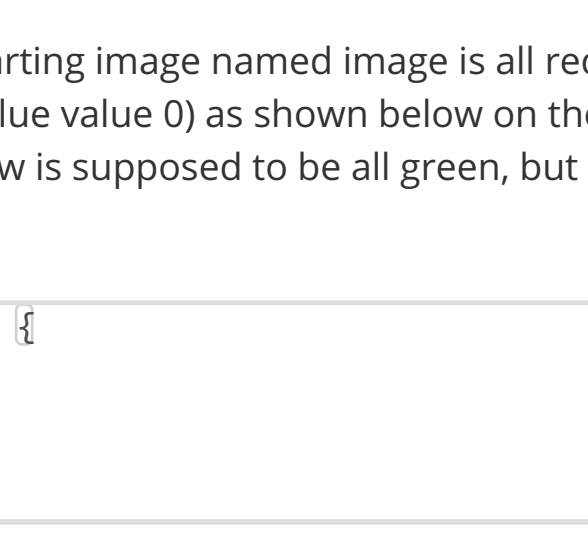
-
```
w = image.getWidth();
for (var pixel of image.values()) {
  x = pixel.getX();
  y = pixel.getY();
  if (x + y < w/2) {
    pixel.setRed(0);
    pixel.setBlue(0);
  }
}
```

-
```
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);
  }
}
```

2. 1 / 1 points

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

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

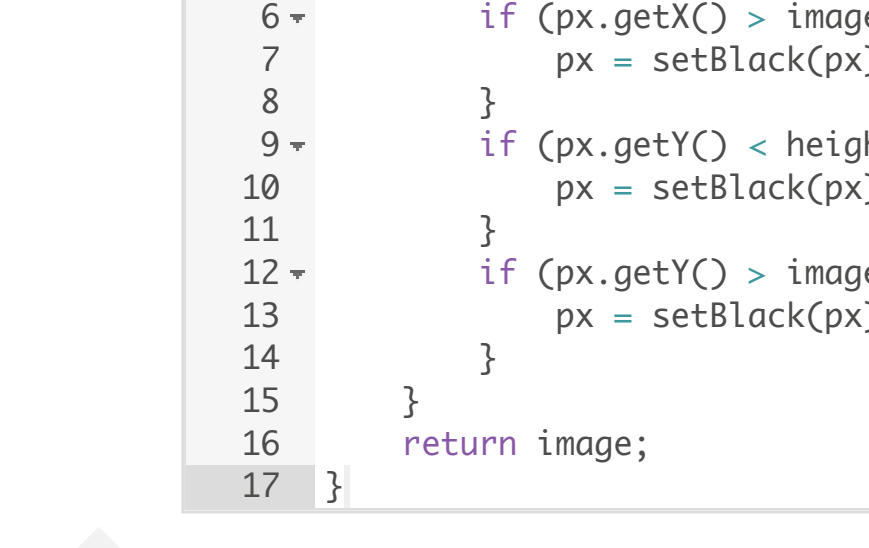
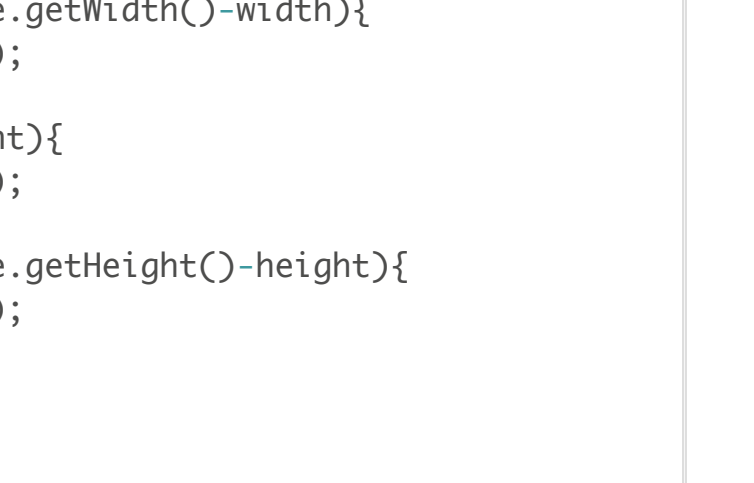
-
```
pixel.setRed(255);
```

- (selected)
```
pixel.setRed(0);
```

Correct
Correct!

3. 1 / 1 points

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 two of the following are correct implementations of addBorder?

- [ ]
```
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

- [x]
```
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;
}
```

Correct

- [x]
```
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;
}
```

Correct

- [ ]
```
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;
}
```

Un-selected is correct

- [ ]
```
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;
}
```

Un-selected is correct

4. 1 / 1 points

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 identify pixels within the vertical borders
- (selected) To identify pixels that are within the borders by returning true

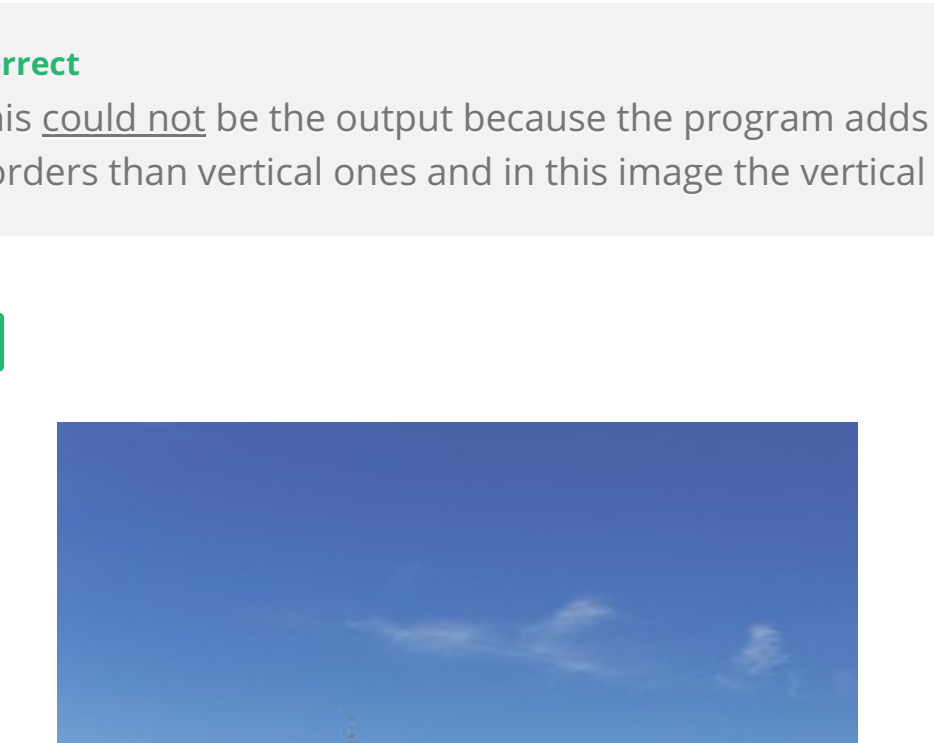
Correct
Correct!

- To color pixels that are within the borders black

5. 1 / 1 points

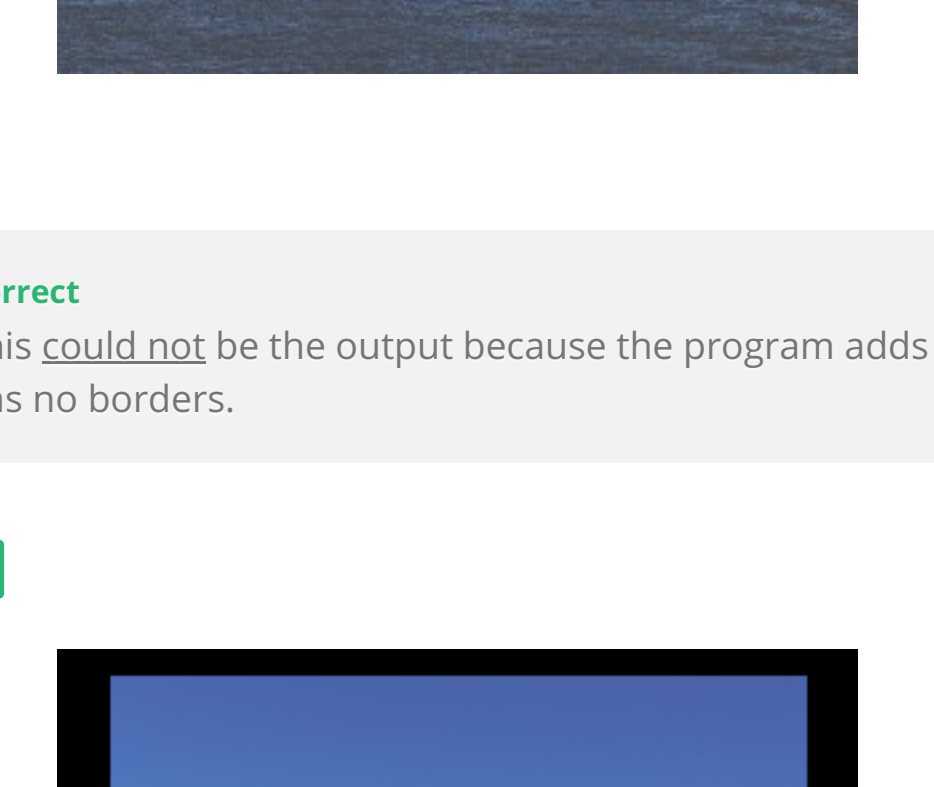
Which of the following could not be the output of running the program written in the previous question? Select all that apply.

- [x]



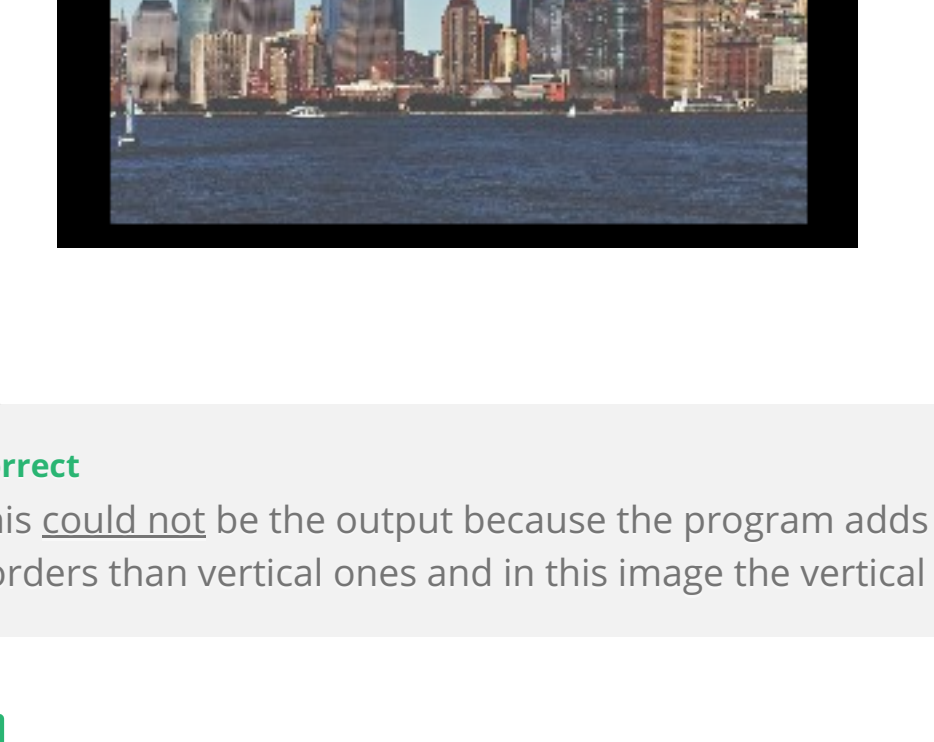
Correct
This could not be the output because the program adds thicker horizontal borders than vertical ones and in this image the vertical borders are thicker.

- [x]



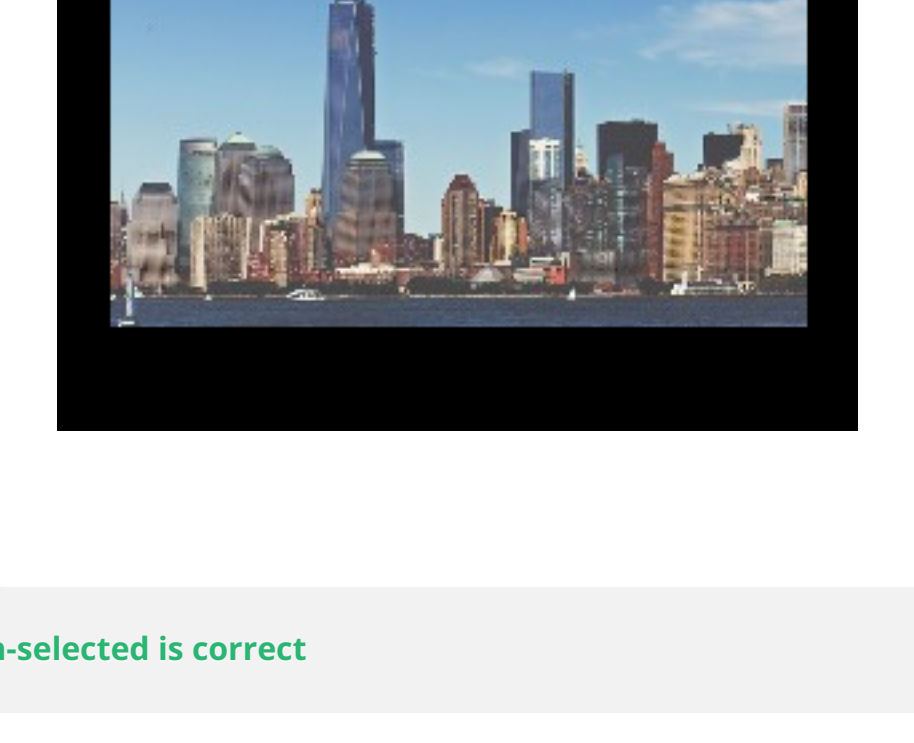
Correct
This could not be the output because the program adds borders and this image has no borders.

- [x]



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

- [ ]



Un-selected is correct

- [x]

Correct
This could not be the output because the program adds thicker horizontal borders than vertical ones and in these images the horizontal and vertical borders are of equal thickness.