Congratulations! You passed!

Grade received 90%

Latest Submission Grade 90%

To pass 80% or higher

Go to next item

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. **0 / 1 point**

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);
  }
}
```

```
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);
  }
}
```

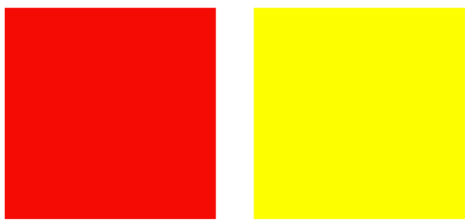
```
w = image.getWidth();
for (var pixel of image.values()) {
  x = pixel.getX();
  y = pixel.getY();
  if (x + y > w/2) {
    pixel.setRed(0);
    pixel.setBlue(0);
  }
}
```

Incorrect

This code sets the red and blue values to 0 for pixels that have an x coordinate that is smaller than their y coordinate. Think about which part of the image those pixels are in. Review how x- and y-coordinates work in images on our documentation page.

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. **1 / 1 point**

```
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.setRed(0);
```

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

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

Correct

Correct!

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. 1 / 1 point


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;
}
```

- [ ]
```
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;
}
```

- [ ]
```
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;
}
```

- [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

- [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

4. Consider the following program that uses the setBlack function you wrote in the **Advanced Modifying Images** programming exercise: 1 / 1 point

```
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 that are within the borders by returning true
- ( ) To identify pixels within the vertical borders
- ( ) To color pixels that are within the borders black
- ( ) To identify pixels within the horizontal borders

Correct
Correct!

5. Which of the following could not be the output of running the program written in the previous question? Select all that apply. **1 / 1 point**

- [ ] 
- [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.

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.

6. Consider the following function named **compose** that has two string parameters, one named **word** and one named **separator**, and returns a string of the word three times with the separator in between each pair of words. The function is shown below with missing code and two calls to the function. 1 / 1 point

```
function compose(word, separator) {
  // missing code
}

var phrase = compose("duck", "goose");
print(phrase);

var phrase2 = compose("meow","-");
print(phrase2);
```

Here is the corresponding output to those calls to the function:

```
duckgooseduckgooseduck
meow-meow-meow
```

Which one of the following is the correct missing code for this function?

- (●) return word + separator + word + separator + word;
- ( ) var temp = word + separator;
  var answer = temp + temp + temp;
  return answer;
- ( ) var answer = word + separator;
  answer = answer + answer + answer;
  return answer;
- ( ) return word + separator;
  return word + separator;
  return word;

Correct
This is the correct answer.

7. Consider the following code with if statements, in which lines are numbered for convenience so we can refer to them. 1 / 1 point

```
1  if (x > 10) {
2    print("one");
3    if (y < 10) {
4      print("two");
5    }
6  }
7  if (x < 20) {
8    print("three");
9    if (y > 5) {
10     print("four");
11   }
12 }
```

If x has the value 30 and y has the value 8, which one of the following is the output resulting from this code executing?

- ( ) one
  two
  four
- (●) one
  two
- ( ) one
- ( ) one
  two
  three
  four

Correct
This is the correct answer.

8. Consider writing the function named moreRedThanGreen that has one parameter named namefile, that is the name of an image. This function calculates how many pixels in an image have more red than green in the pixel (that is the red number is larger than the green number). 1 / 1 point

The function moreRedThanGreen has been started for you but has missing code:

```
function moreRedThanGreen(namefile) {
    var someImg = new SimpleImage(namefile);

    Missing Code

    return count;
```

```
}
```

Which one of the following is the correct missing code?

```
var count = 0;
for (var pix of someImg.values()) {
    if (pix.getRed() > pix.getGreen()) {
        count = count + 1;
    }
}
```

```
var count = 0;
for (var pix of someImg.values()) {
    if (pix.getRed() >= pix.getGreen()) {
        count = count + 1;
    }
}
```

```
for (var pix of someImg.values()) {
    var count = 0;
    if (pix.getRed() >= pix.getGreen()) {
        count = count + 1;
    }
}
```

```
for (var pix of someImg.values()) {
    var count = 0;
    if (pix.getRed() > pix.getGreen()) {
        count = count + 1;
    }
}
```

Correct
Excellent work!

9. Consider writing the function named numberRedPixels that has three parameters. One parameter is named namefile, and is the name of an image. The other two parameters are integers and named low and high. This function calculates how many pixels in an image have a red component whose value is greater than or equal to low and less than or equal to high. 1 / 1 point

The function numberRedPixels has been started for you but has missing code:

```
function numberRedPixels(namefile, low, high) {
    var someImg = new SimpleImage(namefile);
    var count = 0;
    for (var pix of someImg.values()) {


    MISSING CODE


    }
    return count;
}
```

Which one of the following is the correct missing code?

```
var red = pix.getRed();
if (red >= low) {
    if (red <= high) {
        count = count + 1;
    }
}
```

```
var red = pix.getRed();
if (red >= low) {
    count = count + 1;
}
if (red <= high) {
    count = count + 1;
}
```

```
var red = pix.getRed();
if (red > low) {
    count = count + 1;
}
if (red < high) {
    count = count + 1;
}
```

```
var red = pix.getRed();
if (red < high) {
    if (red > low) {
        count = count + 1;
    }
}
```

Correct
Great job!

10. Consider writing code to answer the following question. How many pixels in an image have a red component whose value is greater than or equal to one number and less than or equal to a second number? **1 / 1 point**

Suppose you were to write a function to calculate this value. Which one of the following is the best choice for the parameters for this function?

- (●) Three parameters: 1) filename - the name of the file for the image 2) low - representing the lowest red value and 3) high - representing the highest red value in the range.
- ( ) Four parameters: 1) filename - the name of the file for the image 2) low - representing the lowest red value, 3) high - representing the highest red value in the range, and 4) pixel - representing a pixel.
- ( ) Five parameters: 1) filename - the name of the file for the image 2) low - representing the lowest red value, 3) high - representing the highest red value in the range, 4) height - representing the height of the image, and 5) width - representing the width of the image.
- ( ) Three parameters: 1) filename - the name of the file for the image 2) height - representing the height of the image, and 3) width - representing the width of the image.

Correct
This is the correct answer.