

---

**Name:****Spire ID:**

---

**COMPSCI 220****Written Homework 1**

---

**Question 1:** In part 4 of the programming component, you wrote a `blur` function. If the function is provided an image of size 20 pixels x 20 pixels, what is the minimum number of times (in total) that `blur` would have to call `getPixel` to compute the values of **three** blurred pixels, at coordinates (10,3), (9, 4), and (11, 4) ?

**Question 2:** Consider the following code fragment, that is supposed to compute the pixel value in `highlightEdges`, for part 3 of the programming component.

```
let c = image.getPixel(x, y);
const m1 = (c[0] + c[1] + c[2]) / 3;
c = image.getPixel(x + 1, y);
const m2 = (c[0] + c[1] + c[2]) / 3;
image.setPixel(x, y, [m1 - m2, m1 - m2, m1 - m2]);
```

**Part 1.** Give three pairs of pixel values  $(x,y) = [?, ?, ?]$  and  $(x + 1, y) = [?, ?, ?]$  in the input image, for which this code **does** produce the correct result.

$(x,y) =$ [     ,     ,     ]	$(x,y) =$ [     ,     ,     ]	$(x,y) =$ [     ,     ,     ]
$(x + 1, y) =$ [     ,     ,     ]	$(x + 1, y) =$ [     ,     ,     ]	$(x + 1, y) =$ [     ,     ,     ]

**Part 2.** Give three pairs of pixel values  $(x,y) = [?, ?, ?]$  and  $(x + 1, y) = [?, ?, ?]$  in the input image, for which this code **does not** produces the correct result.

$(x,y) =$ [     ,     ,     ]	$(x,y) =$ [     ,     ,     ]	$(x,y) =$ [     ,     ,     ]
$(x + 1, y) =$ [     ,     ,     ]	$(x + 1, y) =$ [     ,     ,     ]	$(x + 1, y) =$ [     ,     ,     ]

Name: \_\_\_\_\_ Spire ID : \_\_\_\_\_

**Question 3:** An implementation of the `makeGrayscale` function, when given the following image as input, produces the following output image.

**Input Image:**

[0.1, 0.2, 0.3]	[0.9, 0.1, 0.2]
[0.0, 0.1, 0.5]	[0.0, 0.0, 0.0]

**Output Image:**

[0.2, 0.2, 0.2]	[0.33, 0.33, 0.33]
[0.3, 0.3, 0.3]	[0.0, 0.0, 0.0]

Which pixel(s) in the output image have the correct expected value, and which pixel(s) in the output image have an incorrect value? Provide the pixel coordinates. e.g. "(1,0)".

**Correct Pixels:**

**Incorrect Pixels:**

**Question 4:** Consider the following function, that is supposed to return true only if provided two arrays `a` and `b` that are the same.

```
function arrayEqual(a, b) {
  for (let i = 0; i < a.length && i < b.length; ++i) {
    if (a[i] !== b[i]) {
      return false;
    }
  }
  return true;
}
```

**Part 1.** Give three pairs of arrays for which the function **correctly returns false**.

Name: \_\_\_\_\_ Spire ID : \_\_\_\_\_

a = [                    ]	a = [                    ]	a = [                    ]
b = [                    ]	b = [                    ]	b = [                    ]

**Part 2.** Give three pairs of arrays for which the function **correctly returns true**.

a = [                    ]	a = [                    ]	a = [                    ]
b = [                    ]	b = [                    ]	b = [                    ]

**Part 3.** Give three pairs of arrays for which the function **incorrectly returns true**.

a = [                    ]	a = [                    ]	a = [                    ]
b = [                    ]	b = [                    ]	b = [                    ]