CS/INFO 3300 // INFO 5100
Homework 2
Due at 11:59pm Tuesday, September 10

Goals: Get practice using JS functions and working with SVG elements.

Your work should be in the form of an HTML file called index.html. At the top of your <body>, please **put your name and netID in a <h3> tag**. Include one <p> element per problem. Wrap any SVG code for each problem in a <svg> element nested within the <p> element.

Create a zip archive containing your HTML file and upload to CMS before the deadline.

1. In a <script> tag for this problem, **create a recursive function**, `fact( n )`, **which takes an integer,** `n`, **as an argument and returns the factorial of** `n`. Do not use loops or the built-in factorial functions to accomplish this. Remember that recursion involves two major components: base case(s) and recursive calls of the same function.

Make sure that your `fact` function **handles negative integers by returning** `undefined`, properly **returns 1 for** `fact( 0 )`, and otherwise **uses recursion to compute correct factorials** for larger integers.

Include the following code block after your function to prove it works as expected:

```
console.log( fact(-2) );
console.log( fact(0) );
console.log( fact(1) );
console.log( fact(5) );
console.log( fact(12) );
```

(As the function is only intended to handle integers, do not worry about implementing the gamma function for decimal numbers)

2. In class, we learned that <path> elements can be very useful (see docs here). In the <p> tag for this problem, please write a sentence for each of the following commands that describes what it does when it is included in the `d` attribute of a <path> element (for example, `A` tells the path to draw an *arc* on the canvas):

a. `M`
b. `L`
c. `V`
d. `H`
e. `Z`

(next page)

3. Since we want you to get more experience working with SVG, we would like you to complete the final problem in this assignment using **handwritten SVG elements**. You are not permitted to use Javascript for this problem. You can use a color picker tool to find good colors, but once again they must appear as literal elements in the file. You **may not** use an SVG authoring tool like Adobe Illustrator. We can tell if you use one, and the time it would take to obfuscate their output is much greater than just doing the assignment.

Create a `400x400` `<svg>` canvas element in the `<p>` tag for this problem.

Use `<line>` and `<rect>` elements to create your own version of a Piet Mondrian painting in your canvas. Piet Mondrian was an early 20th century artist who, as a member of the De Stijl movement, reduced his art to three primary colors and black lines in a series of famous *neoplasticist* works. Here are some examples: one, two, three. As art often thrives in the presence of constraints, here are some rules to follow for your own work:

- You must include at least 6 lines and 3 rectangles.
- `<line>` elements must use a black stroke and `<rect>` elements must be either white, red, yellow, or blue fill (they do not need to be `#F00`, `#0F0`, and `#00F`, just recognizable shades of those colors).
- `<rect>` elements cannot have a stroke – only use a fill for them.
- If you want black borders, you must use `<line>` elements to create them.
- You may include additional features you feel would add aesthetic value, such as a frame around the canvas, but you must include required elements to receive credit.

If you use a tool to generate coordinates for shapes, please cite that tool. Faithfulness to art history will not be evaluated. We are not grading based on creativity, but obviously poor or incomplete submissions will be penalized.