IMY 220
Practical 3:
JavaScript

Due: Friday 16 August @ 11:30

The submission instructions are available on ClickUP. Any deviation from these instructions will cause a 10% deduction from your mark.

Instructions

- Use JavaScript to create a Euclidian Distance calculator and to control a simple animation.
- Download the files *index.php* and *script.js* from ClickUP. The HTML file contains a form for entering the coordinates of 2 points in 3-dimensional space (Section 1) and the block that must be animated (Section 2). The JS file contains two event handlers which should fire when the "Calculate Length" and "Start Animation" buttons are clicked.
- In order to clarify the required functionality for this assignment, an example file called *P3_memo.html* has been included for you to download, which demonstrates what you are required to implement using react.js. (Don't worry about the code contained inside this file, just use the file to understand the required functionality. If your code looks anything like the code inside this file, it will not be marked.)

Section 1

Correctly include *script.js* inside *index.html* (after correctly doing this, you should be able to add code inside the event handler code which will fire when the buttons are clicked). Inside *script.js* create constructor functions for two user-defined objects called Point and Line, which must have the following member variables/functions:

Point

- **coords**: this must be passed as a parameter to the constructor function and will always be an array of numerical coordinates. For example, if 1, 1, and 1 are entered into the input boxes, the array for that Point will be [1, 1, 1].
- **numDimensions**: this must be set to the number of dimensions the point is being defined as, in other words, the length of the coords array.

Line

- **point1**: this must be passed as a parameter to the constructor function and will always be a Point object
- **point2**: this must be passed as a parameter to the constructor function and will always be a Point object
- **getLength**: this must be a function that calculates the Euclidian distance between the two points in n-dimensional space. (Note that although the given form only allows the creation of two different points in 3-dimensional space, the getLength function should work for points in

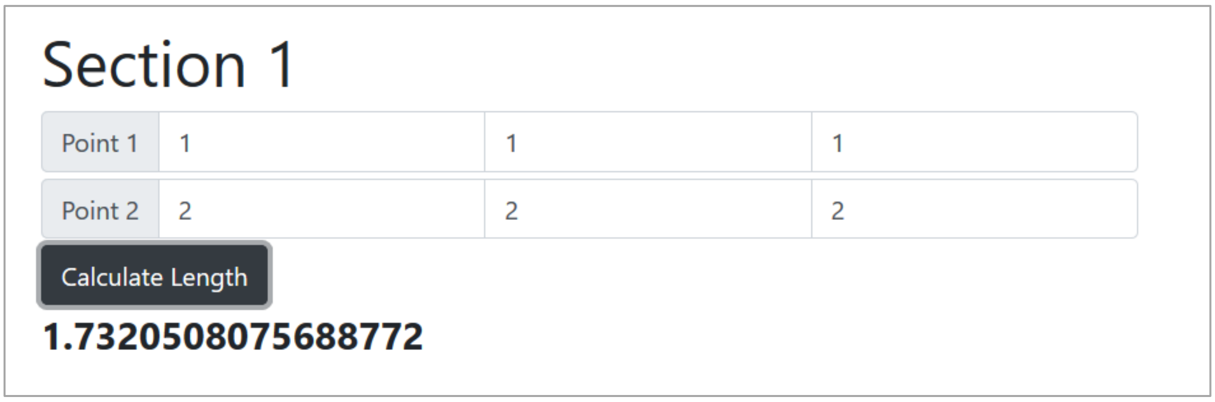
any number of dimensions.) The function should first check if the two points have been defined in the same number of dimensions, in other words, that the length of their coords arrays are the same and return false if they are not. If they are, the function should calculate and return the Euclidian distance between the two points, which is calculated as follows:

$$d(\mathbf{p},\mathbf{q}) = d(\mathbf{q},\mathbf{p}) = \sqrt{(q_1-p_1)^2 + (q_2-p_2)^2 + \cdots + (q_n-p_n)^2}$$
$$= \sqrt{\sum_{i=1}^{n}(q_i-p_i)^2}.$$

In the formula above, p and q refer to any two points and q_1 refers to the first coordinate of q, q_2 to q's second coordinate, etc. and the same for p_1, p_2, etc.

Note that you are **not** allowed to write a loop to calculate this answer. You must use one of the built-in JavaScript array functions to loop through the arrays and calculate the answer.

So, when the "Calculate Length" button is clicked, two 3-dimensional Point objects must be created (one for each group of three input boxes as indicated on the form) and a new Line object must be created with those two Point objects. The length of the Line, i.e. the Euclidian Distance between the two points, must then be calculated and the result shown in the *div* with an *id* of "*result*". For example, if the following values are entered, the following result must be shown:


Section 1

Point 1	1	1	1
Point 2	2	2	2

1.7320508075688772

Section 2

Use a timer to add simple animation to the black block when the "Start Animation" button is clicked. The block has to animate between being 0 and 200 pixels wide by growing or shrinking 1 pixel at a time every 10 milliseconds. For example, when starting, the block should grow until it is 200 pixels wide and then start shrinking until it is 0 pixels wide, start growing again until 200 pixels, etc. which should continue indefinitely. This only has to work for the first time the button is clicked, in other words, the animation does not have to restart or continue to work normally if the button is clicked more than once. Refer to *P3_memo.html* for an example of how the functionality should work.

HINT: the width of the block can be set as follows (where newWidth is the new pixel width)

```
document.getElementById("block").style.width = newWidth + "px";
```

Additional Information

- Refer to the slides as well as online resources for help.

Submit only the following file(s) according to the submission instructions.

- *index.html*
- *script.js*