Lab Summary

In this lab you will build a random number generator function and a search function. Both functions will implement JavaScript Array prototype and Math Methods.

Lab Files, Context, and Goals

Place all code in a JS file, called, *YourName_Lab3.js*

You will not build a web application for this lab, rather, your logic will run in the console in VS Code. Make sure that you have Code Runner installed as an extension in VS Code and that you have installed NodeJS. Your console application will not work if they are not present on your machine.

To build good functional style code, you are charged with these goals:

- Only use JS Array Prototype/Math methods to solve this lab
- Do not mutate immutable types: most often this is seen as ++ or += on number and string types.
- Avoid for/while loops that implement counters and instead use for … of, for … in, and/or an Array Prototype iteration method
- If you get stuck and cannot achieve all these goals, you can "break" these lab rules and find a way that works for you (see the grade rubric for more information on this choice). It is always a good choice to get things working rather than walking away from a challenge.
- You will need to perform a bit of research to implement strategies that may be new to you and will not be found in the Array Prototype site. If you used code from a site other than the Array Prototype site, please place a comment in your code with a hyperlink to that sample source.

Lab Challenges

1. Code a function that requires 2 input parameters – one that can be used to set a low value and the other for a max/high value. Both parameters will be used in the function to generate 5 random and unique numbers in an array that all fall within the low and high range (inclusive). Your function should return the 5 random numbers, sorted. Shown below are some test cases and resulting output to help you interpret these requirements:

   Test values of low=10 and high=20 was used:

   ```
   [Running] node "c:\Use
   [ 11, 12, 13, 15, 18 ]
   ```

   Test Values of low=100 and high=150 was used:

   ```
   [Running] node "c:\Users\mb
   [ 102, 104, 140, 141, 145 ]
   ```

2. Code a function that implements 4 input parameters:
   a. low: the low range for generating a random array of numbers
   b. high: the high range for generating a random array of numbers
   c. times: how many times you will generate an array of 5 numbers between the low and upper range
   d. search: a number to search for in the array of 5 numbers every time you generate a new array of numbers

   Call this function, using the 4 parameters to iterate for how every many times that the input parameter, times, specifies. Every time the code loops, call the function that you coded in Step 1 to generate an array of 5 numbers within the low and high range, and search through the array for the search number. After you have completed iteration, have this function return the number of times

that the search number was found in each random array of numbers. Use console logging to test your results. Shown below is a test case, logging, and results:

```
Using 10 for the low and 25 for the high range.
[ 10, 11, 12, 17, 22 ]
[ 10, 15, 18, 21, 25 ]
[ 12, 14, 16, 18, 25 ]
[ 11, 16, 19, 23, 25 ]
[ 10, 18, 19, 20, 23 ]
[ 10, 11, 12, 15, 24 ]
[ 11, 12, 13, 22, 24 ]
[ 12, 19, 20, 22, 23 ]
[ 13, 18, 20, 22, 25 ]
[ 10, 14, 16, 19, 21 ]
When the number 19 was searched for in random values 10 times, it was found 4 times
```

Lab Due Date

Monday, March 20th before midnight

Lab Grading

- Functional style rules were implemented (10 Marks):
  - 100% of the lab – 10/10
  - 80% - 99% of the lab – 8/10
  - 50% - 79% of the lab – 5/10
  - Less than 50% - 0/10
- Challenge 1 was achieved (5 marks)
- Challenge 2 was achieved (8 marks)
- Hyperlinks to researched code were provided as comments in the lab (2 marks)