**Review Activities**

* **Get Letter Grade (num\_to\_letter\_grade\_starter.py)**
  + Write a function that takes in a number grade and prints the associated letter grade
    - Examples:
      * >> get\_letter\_grade(95)

A

* + - * >> get\_letter\_grade(82)

B

* **Print only the even numbers in an arbitrary list (even\_elements\_starter.py)**
  + Write a function that takes in a list of numbers and returns a list with all of the odd numbers removed
* **Print Star Triangle/Tree (star\_triangle\_starter.py)**
  + Write a function that takes in a number and uses a loop to print out a triangle of stars (\*) with as many lines as that number
* **Quadratic function (quadratic\_starter.py)**
  + The quadratic function defines the solutions to the polynomial
  + Write a function that takes in a, b, and c and returns the solutions, given by the quadratic formula:
  + **Note:** the quadratic formula returns two solutions in most cases!

* **Area of Triangle Function (triangle\_area\_starter.py)**
  + Write a function that takes in a triangle’s base *b* and height *h* and returns the area of that triangle

* **Factorials (factorials\_starter.py)**
  + Write a function that takes in a number and returns the factorial of that number
  + factorial(n) = n \* factorial(n-1) \* factorial(n-2) \* … \* 2 \* 1
  + Examples:
    - >> factorial(1)

1

* + - >> factorial(2)

2

* + - >> factorial(5)

60

* **Print Fibonacci sequence (fibonacci\_starter.py)**
  + Write a function that takes a number and returns the list of fibonacci numbers until that number
  + Examples:
    - >> fib(1)

[0]

* + - >> fib(2)

[0, 1]

* + - >> fib(5)

[0, 1, 1, 2, 3]

* + - >> fib(10)

[0, 1, 1, 2, 3, 5, 8, 13, 21, 34]

* **List Shuffle (list\_shuffle\_starter.py)**
* Write a function that takes in two lists and “shuffles” them like you would a deck of cards
* The function should return the shuffled list. It should have one item from one list, then one item from the other list, then one item from the first list, and so on.
* If one list is longer than the other, just append the rest of the longer list onto the shuffled list once you hit the end of the shorter list
* **Phonebook (phonebook\_starter.py)**
  + Program a phonebook
  + Write a function to add an entry to the phonebook
  + Write a function to print the phone number of a name

**Brain Teasers**

1. **Text compressor**
   * Write a function that takes in a string and replaces recurring words with pointers to the initial occurrence of each word
2. **File input**
   * Read and write to a file!
   * Read [this article](https://en.wikibooks.org/wiki/Non-Programmer's_Tutorial_for_Python_3/File_IO) for more information
3. **Encryption/Decryption**
   * Implement a [Caesar Shift](https://en.wikipedia.org/wiki/Caesar_cipher)
     + Write both a function to encrypt a message and decrypt a message
   * Read about other encryption methods and think about how you would implement those!
4. **Binary Search**
   * Write a function that takes in a sorted list and uses the Binary Search algorithm to find a target element
   * Binary search is a divide-and-conquer search algorithm where we start our search by looking at the middle element of our list
   * If the target element is equal to the middle element, then we return true (since we’ve found our element!)
   * Otherwise our target is less than or greater than our middle element, therefore we will want to split our list in half and only look at the upper half (if our target is greater than the middle element) or lower half (if our target is less than the middle element)
   * We want to repeat this approach until we find what we were originally looking for
   * If we cannot find our target value, return false to indicate that it’s not in our sorted list
5. **Conway’s Game of Life**
   * Implement [Conway’s Game of Life](https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life)
   * Create a two dimensional grid (a list of lists)
   * Write a function that checks how many “neighbors” an entry in the grid has
   * Go through every single element and apply the four rules defined [here](https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life#Rules)
   * Write a function that displays the grid in the console by using one letter to represent a living cell and another one to represent a dead cell