## Guidelines:

- **Make sure to submit your files to Google Classroom before the deadline,** otherwise, your work won't be considered for grading.
- **Submit only the java files as separate files (i.e. not as zipped files).**
- **Create a class for every program named based on the question name then write all the necessary methods and/or the main method**

---

## String Recursion:

Write the class StringRecursion that contains the following static methods.

1. **int countUpper(String str)**: counts upper case letters in a string
2. **String stutter(String str)**: repeats every character of the string. "hello" → "hheellllloo"
3. **int toNumber(String str)**: counts the number of digits in a string
4. **int searchString(String str, char c)**: searches for character c in the string str
5. Test your methods in a main method.

## MathRecursion:

Create a class MathRecursion that contains the following static methods:

1. **public static double power(double x, int n)**: calculates the power of x^n, n can be positive or negative. Test comparing it to Math.pow(double, int)
2. **pubic static int factorial(int n)**: calculates the factorial of n. Throws IllegalArgumentException if n is negative
3. **pubic static int gcd(int m, int n)**: calculates the gcd(m,n).
4. Test your methods in a main method.

## ArrayRecursion:

Create a class ArrayRecursion that contains the following static methods:

1. **public static int search(Object[] items, Object target)**: returns the index of target in the array items. Returns -1 if target is not found in items.
2. **pubic static int sum(int[] array)**: calculates the sum of all values in array of integers.
3. **pubic static int max(int[] array)**: returns the maximum of an array of integers.
4. Test your methods in a main method.

## LinkedListRec:

In the class LinkedListRec provided with Lesson 7 write the following recursive methods:

1. **public boolean search(E items)**: returns true if item exists in the list, false otherwise.
2. **pubic void insertBefore(E target, E inData)**: inserts inData before the first occurrence of target. If the target is not found, then the inData is inserted at the end of the list.

3. **pubic void remove(int index)**: removes the item at the specified index. Throws IndexOutOfBoundsException if n is invalid, or if the list is empty.
4. Test your methods in a main method.