Project 2 Pre-work

These short exercises will introduce students to concepts that are helpful in function implementations of this project. Make sure you copy the signatures **exactly**, and **do not** modify them.

1. `public static boolean consecutiveFours(int[] arr)`
   This method will take in an array of integers and return true if the array contains at least 4 consecutive numbers with the same value. **Useful for method 2 (countRuns) and 3 (encodeRle).**
   Ex: `consecutiveFours (new int[]{3,2,5,4,4,4,5,5,5})` returns `false`.
   Ex: `consecutiveFours (new int[]{3,2,5,4,4,4,4,7,12})` returns `true`.

2. `public static int[] sumByParity(int[] arr)`
   Parity is the formal name for the property of a number being even or odd. This method will take in an array of integers, store the sum of all the values located at even indices in the first index of a new array, then store the sum of all the values located at odd indices in the second index of this new array. **Useful for method 4 (getDecodedLength).**
   Ex: `sumByParity(new int[]{5,12,8,5,3,11,7,2,3,16,4})` returns `[30,46]`.

3. `public static int[] expandByIndex(int[] arr)`
   This method will take in an array of integers and expand them into a larger array. The value in the original array represents how many times that index (0-indexed) will appear in the new array. **Useful for method 5 (decodeRle).**
   Ex: `expandByIndex(new int[]{2,1,3})` returns `[0,0,1,2,2,2]`.

4. `public static int numericalCount(String string)`
   This method will take in a string that is composed of numbers and letters. It will return the count of numbers in that string, ignoring letters. **Useful for method 6 (stringToData).**
   Ex: `numericalCount("abcd3fgh1")` returns `2`.