ACS-2947-050
Lab #5
Due by Friday November 2 at 11:59 pm
Submit your `.java` files to 2947L-070@acs.uwinnipeg.ca or 2947L-071@acs.uwinnipeg.ca

1. Using your Lab 4 `LinkedPositionalList` implementation, add a position and element iterator. Alternatively you may use the solution in MS Teams.
   a) Have the `PositionalList` interface extend `Iterable`. Add the empty methods `iterator()` and `positions()` that return `Iterator<E>` and `Iterable<Position<E>>` respectively.
   Option: Use the `PositionalList` provided here.

   b) Add the nested classes and methods from your notes/text.
      - `PositionIterator, PositionIterable, and positions()`
      - `ElementIterator and iterator()`

   * Note that `PositionalList, LinkedPositionalList` and `Lab5_Driver` require `java.util.Iterator`

2. `Lab5_Driver` includes a populated positional list of integers. Complete the driver with the following requirements. Display the list after each step.
   a. Declare and initialize an *elemen*t iterator. Iterate through the list and remove the positions where elements are a 1 or 2.
   b. Using the *position* enhanced for, replace every 4 with 3
   c. With a new instance of *element* iterator, remove all consecutive duplicates
   i.e. for list 1 2 2 3 3 3 4, the result would be 1 2 3 4.
   d. Using the *element* enhanced for, calculate the sum of integers.

EXTRA WORK: Do not submit

Create a class called `RecursionDemo` that contains the following generic static methods:

a. `clearStack()` that *recursively* empties a stack
   - Create and clear a stack, showing the before and after results in your output

* Note that you may use your implementation Stack from our ADT or `java.util.Stack`

b. `numDigits()` method that *uses recursion* to determine the number of digits in a positive integer (recall that the number of digits is $\lfloor \log_{10} n \rfloor + 1$).

   - Use the Scanner object to read the integer input from keyboard and display the result in your output