Assignment 3

Level 1:

A. Design a pseudo code algorithm to take a Sequence and remove all duplicate elements
from the Sequence. Is the algorithm the same for both a List or a Sequence? Explain.
Analyze your algorithm twice, once assuming it is a Sequence and then again assuming it
is a List. Which ADT is a better choice for this problem? Implement your choice in
JavaScript.

B. Design an algorithm, **isPermutation(A,B)** that takes two sequences A and B and
determines whether or not they are permutations of each other, i.e., same elements
but possibly occurring in a different order. **Hint**: assume that A and B do not
contain duplicates (later we will allow duplicates).

What is the worst-case time complexity of your algorithm? Justify your answer.

Implement your algorithm in JavaScript using either the Sequence or the List
class/program provided.

C. Let L be a **List** of objects colored either red or blue. Design an **in-place** pseudo
code algorithm **sortRB(L)** that places all red objects in list L before the blue
colored objects. Thus the resulting List will have all the red objects followed by
the blue objects. **Hint:** use the method swapElements to move the elements
around in the List. **To receive full credit**, you must use positions for traversal,
e.g., first, last, after, before, swapElements, etc. which is necessary to make it in-
place.

Level 2:

D. Let L be a **List** of objects colored either red, green, or blue. Design an **in-place**
pseudo code algorithm **sortRBG(L)** that places all red objects in list L before the
blue colored objects, and all the blue objects before the green objects. Thus the
resulting List will have all the red objects followed by the blue objects, followed
by the green objects. **Hint:** use the method swapElements to move the elements
around in the List. **To receive full credit**, you must use positions for traversal,
e.g., first, last, after, before, swapElements, etc. which is necessary to make it in-
place.

E. Implement C and D in JavaScript using the LinkedList data structure provided.