CCPS506 Lab 3 – Expressions in Elixir

Preamble

Powerful built-in functions for list operations are a hallmark of functional programming languages. In this lab you'll explore the Elixir interactive shell and write several Elixir functions that operate on lists. Every problem in this lab can be very easily solved with explicit looping or branching. However, we're trying to experience functional programming through its best practices rather than simply writing imperative-style code using a functional language. This means getting very good at working with lists!

Lab Description

Write an Elixir function to solve each of the problems below. Each will accept a list as input and return either a Boolean value or another list, depending on the question. All functions should be placed in a single module called **`Lab3`**. Make sure each of your functions is named precisely according to the specifications below. You may not use branching or repetition of any kind. No recursion, no if/else. Only use the list operations we've seen in class so far. They start very easy, and get progressively trickier.

i) **`firstTwo` –** Returns True if the first two elements are the same, False otherwise.

ii) **`evenSize` –** Returns True if the list has an even number of elements, False otherwise.

iii) **`frontBack` –** Removes the first element of the input list and appends it to the back. Return the new list.

iv) **`nextNineNine` –** Insert the integer 99 in the second position of the list.

v) **`sayHello` –** Return True if the list contains the string "Hello", False otherwise.

vi) **`IsCoord`** – Return True if the input list could represent a coordinate, False otherwise. A coordinate has two elements, both are numbers.

vii) **`helloIfSo` –** If the input list contains the string "Hello", remove it and place it at the end of the list. If it does not contain "Hello", add it to the beginning of the list.

viii) **`makeTriple`** – If the length of the list is divisible by three, return the original list. If not, remove elements from the front until it is, and then return the resulting list.

Submission

Your Lab3 module and all its functions must be placed in a single Elixir file called **lab3.ex**.

Labs are to be submitted ***individually***! Submit lab3.ex on D2L.