CCPS506 Lab 3 – Elixir: Lists and Expressions

Preamble

In this lab you'll write several Elixir functions that operate on lists. Every problem in this lab could be very easily solved with explicit looping or branching, but we're trying to experience functional programming through its best practices rather than simply writing imperative-style code in a functional language. Powerful built-in functions for list operations are a hallmark of functional programming languages, so that's where we'll start.

Getting Started

Create an Elixir source file called **`Lab3.ex`**. In this file you will define a module named **`Lab3`**. All functions written for this lab will be placed in this single module. Make sure each of your functions is named precisely according to the specifications below. Alongside this file, you may (should) create another Elixir script file (*.exs). In this script you can write a series of test calls to your functions to make sure they work correctly.

Lab Description

Write an Elixir function in your **`Lab3`** module to solve each of the problems below. Each function will accept a list as input and return either a Boolean value or another list, depending on the question. For this part of the lab, you may ***not*** use branching or repetition of any kind. No recursion, no if/else, no indexing. Head, tail, length, concatenation, and other list operations will serve you well here.

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

vi) **`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 end of the list.

Submission

Labs are to be completed and submitted *individually*. Submit your **Lab3.ex** file containing the Lab3 module and all completed functions on D2L, under the submission for Lab 3.