Swift Extra Credit 1: Syntax

**Files to Submit: swiftSyntax.playground**

**Points: \_\_**

Swift is Apple’s main language used for programming OS X and iOS systems. It was released in 2014, so it’s relatively new.Along with Objective-C, it is the language used to develop iOS applications. This assignment explains some of Swift’s basic syntax. Further tutorials will guide you in creating your own iOS app! **Swift only runs on Macs!** (Don’t worry Android users; there is an extra credit assignment for Android development here. link here

First, download and open XCode from the App Store. Click “Get started with a Playground” when XCode opens. Name your file swiftSyntax when prompted. This is your “playground,” an environment in which you can write and run your code. Get an overview of basic Swift syntax here at the link below. (The “Classes” section introduces object-oriented programming, which you can find more information about in the Java EC assignment. We won’t work with that in this assignment). <https://guides.codepath.com/ios/Understanding-Swift>

Next, practice Swift syntax by redoing the following problems you’ve already completed in Python.

* Hw1pr3: mult(n,m)
  + Write a **recursive** function that returns the value of n\*m
  + Do not use \*
* Hw3pr2: arr\_To\_Str (L)
  + Write a non-recursive function that converts an array of strings into one string using a **for\_in loop**
* Hw6pr5: fac(n)
  + Write a **recursive** function that returns the value of n!
* Hw6pr5: power(b,p)
  + Write a function using a **for\_in loop** that returns b^p

CHALLENGE!!

Write a function called **deleteLast(e: Int, A: [Int])-> [Int]** that returns an array of integers with the last occurrence of the input integer deleted from the input array. If e is not present in A, the function should return A (use an optional to account for this case). The Swift array property count and the method remove(at: )will be helpful in writing this function. For more info on these and other array properties, go to <https://developer.apple.com/reference/swift/array> . Examples of how they are used shown below.

>var A: [Int] = [1, 2, 3]

>print("\(A.count)")

3

>var A: [Int] = [1, 2, 3]

>A.remove(at: 0)

>print("\(A)")

>print("\(A.count)")

[2, 3]

2

**Note:** Because parameters are immutable (can’t be changed) in Swift 3, the first line of your function needs to be var newA = A so that you can recurse on a modified version the input array.

**Hint:** What happens if there are a bunch of the same ints in the array and you need to remove the last one, not the others? One good strategy is to loop through the array and look for the position of the last int with value “e”. Then you can remove the int at that position. This is where a swift optional would be useful! If you make an optional variable to represent the position:

var lastPosition : Int?

lastPosition can either take on a value when your loop locates an int it should remove, or remain nil. If you loop through the entire array and don’t find any ints to remove, you can test if lastPosition is nil, and just return your original array. If lastPosition is not nil, you know it has the last position of the int you should remove. Having lastPosition be an optional is nice because you don’t have to give it an initial value.

Optionals can be tricky the first time you work with them. Read [this](https://medium.com/@ranleung/understanding-optionals-in-swift-540bfa0e44c7) short explanation of optionals for more help. The section on Optional Binding is particularly useful to help you check whether or not lastPosition is nil at the end of your function.

Use these test cases to make sure your function works:

> print("\(deleteLast(e: 2, A: [1, 2, 3, 2, 2, 2, 6]))")

[1, 2, 3, 2, 2, 6]

> print("\(deleteLast(e: 3, A: [3, 5, 2]))")

[5, 2]

> print("\(deleteLast(e: 67, A: [100, 482, 472]))")

[100, 482, 472]