

## Práctica

Lenguaje Python, menú con funciones de listas recursivas

I Semestre de 2017.

---

### Práctica: Funciones con listas recursivas

#### 1. Objetivos del Laboratorio

- Fomentar la robustez de las funciones implementadas con Python.
- Comprender las separaciones de las responsabilidades en las funciones.
- Crear algoritmos recursivos usando listas
- Trabajo en parejas mediante la técnica de trabajo en equipo.
- Aplicar los conceptos de “Clean Code” y “Olores del software”.

#### 2. Introducción

Trabaje de manera individual. Implemente las funciones recursivas solicitadas, controle la validación de los datos y escenarios básicos según corresponda. Respete que cada función realice sólo una única responsabilidad. Recuerde documentar según lo enseñado en clase cada función y cada archivo de Python.

#### 3. Por Hacer

Incorpore a las funciones recursivas usando listas recursivas únicamente:

##### a. Determinar el elemento mayor de una lista.

Ejemplos del comportamiento esperado del algoritmo:

- Si la lista es [0, 3, 8, 1, 5, 2] debe retornar 8
- Si la lista es [-1, -9, -7, -11, -3, -5] debe retornar -1
- Si la lista es [5, 7, 1, 8, 11, 3] debe retornar 11
- Si la lista es [] debe retornar “La lista no contiene elementos”

Escriba una función en Python llamada `determinarElementoMayor` (lista) que tenga el comportamiento mencionado previamente.

## Práctica

### Lenguaje Python, menú con funciones de listas recursivas

I Semestre de 2017.

---

**b. Codificar una función que reciba como entrada una lista que contenga únicamente letras y retorne una nueva lista que contenga solo las vocales de la lista recibida.**

Ejemplos del comportamiento esperado del algoritmo:

- A. Si la lista es ["x", "c", "a", "b", "i", "z", "e"] debe retornar ["a", "i", "e"]
- B. Si la lista es ["b", "f", "g", "d", "x", "j", "p"] debe retornar [ ]
- C. Si la lista es ["u", "a", "i", "o", "e"] debe retornar ["u", "a", "i", "o", "e"]
- D. Si la lista es ["p", "a", "q", "o", "d"] debe retornar ["a", "o"]

**Nota:** En caso que la lista contenga elementos distintos a letras, los debe ignorar.

Escriba una función en Python llamada crearListaVocales (lista) que tenga el comportamiento mencionado previamente.

**c. Codificar una función que reciba dos elementos y una lista. Cada vez que aparezca el elemento1 en la lista debe agregar inmediatamente un elemento2.**

Ejemplos del comportamiento esperado del algoritmo:

- A. Si el elemento1 es "x", el elemento2 es "hoy" y la lista es ["p", "q", "x", "n", "x"] debe retornar ["p", "q", "x", "hoy", "n", "x", "hoy"]
- B. Si el elemento1 es "intro", el elemento2 es "taller" y la lista es ["mate", "conta", "intro", "inglés", "deportiva"] debe retornar ["mate", "conta", "intro", "taller", "inglés", "deportiva"]
- C. Si el elemento1 es "empollar", el elemento2 es "huevo" y la lista es ["reto", "nebulosa", "divide", "vencerás", "examen"] debe retornar ["reto", "nebulosa", "divide", "vencerás", "examen"]
- D. Si el elemento1 es "lunes", el elemento2 es "martes" y la lista es [ ] debe retornar [ ]

Escriba una función en Python llamada insertarElemento (elemento1, elemento2, lista) que tenga el comportamiento mencionado previamente

## Práctica

Lenguaje Python, menú con funciones de listas recursivas

I Semestre de 2017.

---

**d. Codificar una función para extraer todos los primeros elementos de las sublistas contenidas dentro de la lista dada.**

Ejemplos del comportamiento esperado del algoritmo:

- A. Si la lista es `[[8,9],[11,2],[4,3],[6,7],[1,14]]` debe retornar `[8,11,4,6,1]`
- B. Si la lista es `[[1,2],[3,4],[5,6],[7,8],[9,10]]` debe retornar `[1,3,5,7,9]`
- C. Si la lista es `[[7,8],[7,8],[7,8],[7,8],[7,8]]` debe retornar `[7,7,7,7,7]`
- D. Si la lista es `[[8,9,10],[8,9,10], [8,9,10], [8,9,10], [8,9,10]]` debe retornar `[8,8,8,8,8]`

Escriba una función en Python llamada `obtenerPrimeros (lista)` que tenga el comportamiento mencionado previamente.

**e. La profundidad de una lista es un número entero determinado por el máximo nivel que alcanza una de sus sublistas. Codificar una función recursiva que determine la profundidad de una lista.**

Ejemplos:

`[1, 2, 3, 4, 5, 6, 7]` la profundidad es 1

`[1, 2, [3], 4, [5], 6, 7]` la profundidad es 2

`[1, [2, [3],4], 5, 6, [7] ]` la profundidad es 3

Puede codificar las funciones adicionales que necesite para resolver los retos planteados.