

Listas en Python

En el siguiente laboratorio se ahondará a través de ejercicios en los aspectos vistos en el material titulado “Listas y ciclos for-in”.

Lea atentamente las instrucciones proporcionadas a continuación antes de continuar.

Instrucciones

1. El trabajo es de carácter **individual**.
2. Dispone de **60 minutos** para responder.
3. Cualquier duda consultarla con el ayudante o el profesor levantando la mano.
4. Los programas deben ir **comentados**, guardando la **estructura de programas** vista en la clase 6 de Cátedra y tanto las **variables**, **constantes**, como **funciones** deben llevar nombres representativos

Listas y ciclos for-in

1. Se tenía un programa que entregaba una lista con los cubos de los números naturales entre un número dado y 1 en orden decreciente a partir de un único valor n que ingresara un usuario.

El programador que había escrito este programa olvidó guardar el programa y perdió todo el código que había escrito, sin embargo, en una versión anterior, pensando su solución antes de implementarla, había dejado los comentarios de las operaciones que iba a realizar. A partir de este código, reconstruye el código que el programador perdió, considerando que:

- Se requiere que se reconstituyan cada una de las funciones de acuerdo a las originales (es decir, cada función debe realizar la tarea asignada y sólo la tarea asignada en los comentarios).
- Se requiere además la reconstitución del bloque principal del programa.
- Se debe guardar la estructura del programa que el programador tenía originalmente (es decir, los bloques de entrada, salida y definición de funciones deben estar de acuerdo a los comentarios).

- El procesamiento que el programa realiza es el siguiente:
 1. El programa recibe una entrada `n` y valida que sea un número natural mayor que 0, si no lo es, vuelve a preguntar.
Por ejemplo: 5
 2. Con la entrada validada se crea una lista con los elementos entre 1 y `n` utilizando una función `crearLista(*)`.
Por ejemplo: [1, 2, 3, 4, 5]
 3. Luego la función `eLevarAlCubo(*)` recibe el listado para elevar cada uno de los elemento de la lista al cubo.
Por ejemplo: [1, 8, 27, 64, 125]
 4. Una última función invierte el orden de los elementos de la lista `invertirLista(*)`.
Por ejemplo: [125, 64, 27, 8, 1]
 5. Finalmente se entrega el resultado impreso por pantalla al usuario con `imprimirLista(*)`.
Por ejemplo: 125, 64, 27, 8, 1
- Se prohíbe usar la función `reverse()` de Python para invertir la lista.
- Si no puede conseguir una de las funciones omítala y avance hacia las que siguen en el código.

```
# -*- coding: cp1252 -*-
# AUTOR: Coordinación de Fundamentos de Computación y Programación
# FECHA: Segundo Semestre de 2013

# Programa que recibe un número natural n, como entrada de un usuario
# a partir de éste se crea una lista con todos los números de 1 a n,
# eleva cada uno de los elementos al cubo, los invierte y los muestra
# por pantalla al usuario

#
# FUNCIONES
#
# IMPORTACIÓN DE FUNCIONES

# Desde el módulo math importo la función para elevar al cubo
<COMPLETAR>

# DEFINICIÓN DE FUNCIONES

# Función que crea una lista con los elementos de uno a un número natural
# dado
# Entrada: <COMPLETE USTED AQUÍ>;
# Salida: <COMPLETE USTED AQUÍ>;
<COMPLETAR>

# Función que eleva al cubo cada elemento de una lista
# Entrada: <COMPLETE USTED AQUÍ>;
# Salida: <COMPLETE USTED AQUÍ>;
<COMPLETAR>

# Función que invierte los elementos de una lista
# Entrada: <COMPLETE USTED AQUÍ>;
# Salida: <COMPLETE USTED AQUÍ>;
<COMPLETAR>

# Función que imprime los elementos de una lista en una sola línea
# Entrada: <COMPLETE USTED AQUÍ>;
# Salida: <COMPLETE USTED AQUÍ>;
<COMPLETAR>
```

```
# BLOQUE PRINCIPAL
#
#   # ENTRADA DE DATOS

# Se solicita el ingreso de n y se valida que sea un número
# natural, en caso de que no sea se pregunta nuevamente
<COMPLETAR>

# PROCESAMIENTO

# Se crea la lista, posteriormente se eleva cada uno de los elementos al
# cubo para finalmente invertir el orden de los elementos de la lista
<COMPLETAR>

# SALIDA

# Se imprime la salida al usuario
<COMPLETAR>

# FIN
```