|  |  |  |  |
| --- | --- | --- | --- |
| Generalitat de Catalunya  Departament d’Educació  INS Provençana | Mòdul 03:  DAWBIO1 | Curs  2022-2023 |  |
| Departament de/d’  Informática  Grup  1 | Ex UF2, NF1 | Nota: | |
| Control núm.  3 | Data  16/12/2022 | |

**Professor/a: Éric Expósito i Javier Carrasco**

**Nom i cognoms:**

**INSTRUCCIONES**

* No se permitirán prints innecesarios ni dentro de condicionales ni iteraciones, salvo que sean necesarios
* No se permite el uso de métodos o funciones que no se han expuesto en clase
* Se entregarán tanto archivos como ejercicios tenga la prueba
* No se evaluará código comentado
* La entrega fuera de plazo equivaldrá a un 0
* No se valorarán métodos/funciones de + 10 líneas.
* El programa principal podrá tener también un máximo de 10 líneas. Con excepción de encadenamientos de elif y las líneas de control de errores, que no contarán para este limite.
* Controlar los errores que se puedan producir FUERA de los métodos/funciones. El programa NO puede fallar.
* Todos los métodos deberán estar comentados como se expuso en clase:

*def ejemplo(str):*

*'''*

*Esta función no hace nada*

*:param str: es un string que conforma una palabra para nada*

*:return: el 1, porque funciona*

*'''*

*print("esto no hace nada")*

*return 1*

**ENUNCIADO**

"""

-----------------------------------------------------------

EJERCICIO 1 ----------------------------------------------

(2 puntos)

- Pedir al usuario su nombre y su DNI sin letra.

- Mostrar todos los datos en forma de tabla.

- Que el programa nos pida cuantos DNI queremos añadir

-

Usuarios

---------

Nombre --> XXXXXXXX-X

Las letras son: TRWAGMYFPDXBNJZSQVHLCKE

Y las letras se relacionan del 0 al 22:

0 = T

1 = R

2 = W

etc...

"""

"""

-----------------------------------------------------------

EJERCICIO 2 ----------------------------------------------

-----------------------------------------------------------

(4 puntos)

Haremos un menú con las siguientes opciones: Salir, Crear BBDD, Crear 1 elemento BBDD

Consultar toda la BBDD, Consultar 1 elemento BBDD, Eliminar elemento BBDD

base\_datos = [

["ID",

"Nombre",

"Teléfono",

"Fecha Nacimiento"]

]

- Crear BBDD: Generara un conjunto de datos para nuestra base de datos con 4 personas diferentes,

con 4 ID, 4 nombres...etc.

- Añadir 1 elemento: Pedir al usuario los datos y añadir un nuevo elemento a nuestra base de datos

- Consultar base de datos: Mostrar la base de datos en formato tabla:

BBDD

----

ID - Nombre - ...

ID - Nombre - ...

- Consultar un elemento: Pedir al usuario un ID, y mostrar todos sus datos:

ID: ----

Nombre: -----

...

- Eliminar elemento: pedir un ID, mostrar los datos y solicitar confirmación del borrado.

ID: ----

Nombre: -----

...

¿Desea eliminar el elemento?

"""

"""

-----------------------------------------------------------

EJERCICIO 3 ----------------------------------------------

-----------------------------------------------------------

(2 puntos)

Completa el ejercicio adjunto, .py

Os dejamos la información.

PARTE 1:

- Haz que funcione el código, y se muestren los 3 prints con las 3 listas

PARTE 2:

- Haz que el código principal (sin contar las listas) sea de 1 línea

lst = ["Blanco", "Verde", "Amarillo", "Rojo", "Azul"]

lst2 = ["Blanco", "Negro", "Rojo", "Gris", "Naranja"]

def mostrar\_lista(lst):

for e in lst:

print(e)

def get\_coincidencias(lst, lst2):

comparacion = []

for i in lst:

if lst[i] in lst2:

comparacion.append(lst[i])

return comparacion

mostrar\_lista(lst[2])

print("LST2")

mostrar\_lista(lst2)

lista\_comparada = get\_coincidencias(lst)

print("LST COMPARADA")

mostrar\_lista(lista\_comparada)

"""

"""

-----------------------------------------------------------

EJERCICIO 4 ----------------------------------------------

-----------------------------------------------------------

(2 puntos)

Dadas las listas:

lst1 = [3, 8, 6, 5, 7, 7, 2, 7, 7, 10, 3, 4, 3, 1]

lst2 = [3, 1, 3, 6, 7, 0, 3, 9, 4, 5, 4, 2, 9, 2, 0]

]

4.1 - Compararemos ambas listas, mostrando por pantalla cuál de las 2 es mas larga.

4.2 - Realizaremos la media de cada una, indicando el resultado y cuál tiene la media más grande.

4.3 - Recorreremos la lista 1 y compararemos cada elemento de la lista, con el que tenga la misma posición de la otra. Indicando que lista posee el elemento con un valor más pequeño.

4.4 - Ordenaremos ambas listas de la misma forma y las mostraremos una de mayor a menor y la otra al revés de forma invertida.

"""