

## Práctica 5:

### Listas

1. Escriba los siguientes programas. Nota: No utilice los metodos `index`, `min`, `max`, `reverse`.

- a. Dada una lista de números `list` y un número `n`, determine en qué índice de la lista `list` se encuentra el número `n`. En caso de no encontrarlo, el programa mostrará `-1`. Ejemplos

```
list = [11,25,3,-4,95]
n = 3
# El programa debería mostrar
2
```

```
list = [1,2,3,4,5]
n = 10
# El programa debería mostrar
-1
```

- b. Dada una lista de números, calcule el mínimo y el máximo de la lista. **Nota:** es posible hacerlo recorriendo una única vez la lista o recorriéndola dos veces. Piense las ventajas y desventajas de ambos métodos.

```
numeros = [11,25,3,-4,95]
# El programa debería mostrar
El mínimo es -4
El máximo es 95
```

- c. Dada una lista de números, cree una nueva lista sumando 1 a cada elemento.

```
numeros = [0,1,2,3,4]
# El programa debería mostrar
[1,2,3,4,5]
```

- d. Dada una lista `palabras` de cadenas de texto, devuelva una nueva lista formada sólo por las cadenas de `palabras` que empiezan con "a".

```
palabras = ["arbol", "barco", "artificial", "casa", "dado", "a"]
# El programa debería mostrar
["arbol", "artificial", "a"]
```

- e. Dada una lista de números, calcule, por un lado, la suma de los elementos que se encuentran en un índice par en la lista y, por otro lado, el producto de los elementos de posiciones con índice impar.

```
numeros = [0,1,2,3,4,5]
# El programa debería mostrar
La suma de índices pares es 6
El producto de índices impares es 15
```

- f. Dada una lista cualquiera, cree una nueva lista con los mismos elementos pero en el orden inverso.

```
numeros = [0,1,2,3,4,5]
# El programa debería mostrar
[5,4,3,2,1,0]
```

- 
2. Escriba un programa que dada una lista de números `list` devuelva otra lista cuyos elementos sean las sumas acumuladas de los elementos de `list` en cada posición. Es decir, una nueva lista donde el primer elemento es el mismo que en la lista original `list`, el segundo elemento es la suma del primer y segundo elementos de `list`, el tercer elemento es la suma del resultado anterior con el tercer elemento de la lista original y así sucesivamente. Por ejemplo, dada la lista `[1, 2, 3]`, la suma acumulada debería ser `[1, 3, 6]`.
  3. Escriba un programa que dada una lista determine si tiene algún elemento repetido e imprimirlos. Puede asumir que un numero se repite a lo sumo una vez. **Pista:** Utilice slicing de listas.
  4. Calcule los `n` primeros números de la secuencia de Fibonacci en una lista. Es decir, el programa comenzara con la lista `[0,1]` y computará iterativamente el siguiente número de la secuencia.
  5. El objetivo de este ejercicio es recolectar datos de personas y almacenarlos en una especie de base de datos. A través de lo diferentes ítems, los iremos guiando en la creación de la misma.

- a. Escribir un programa que le pida al usuario ingresar por consola los siguientes datos por separado: Nombre, Apellido, Localidad, Edad, DNI, Carrera universitaria en curso, año de inicio de la carrera. Guardar los datos en una lista llamada `datos_personales` e imprimirlos en pantalla.

```
ejemplo: ['María', 'Gonzalez', 'Chascomús', 24, 50708934, 'Ing. civil', 2019]
```

- b. A partir de la lista del ejercicio anterior, genere un programa que calcule los años que la persona lleva cursando la carrera y lo agregue a la lista. Por ejemplo, si la persona inició su carrera en el 2019, la cantidad de años cursados a agregar es 4.

```
ejemplo: ['María', 'Gonzalez', 'Chascomús', 24, 50708934, 'Ing. civil', 2019, 4]
```

Finalmente, imprima una frase que indique el nombre, el apellido y cantidad de años cursados de la persona

- c. Utilice el código de los ítems de arriba y automatice la recolección de datos para una cantidad de personas desconocidas. El resultado de este ejercicio debe ser una lista llamada `basedatos` donde se almacenarán las listas `datos_personales` de cada individuo que se recolecten
- d. Realice un programa que modifique para cada persona la carrera en curso por la palabra "TUIA".

## Tuplas

6. Cartas como tuplas.
  - a. Proponer una representación con tuplas para las cartas de la baraja francesa.
  - b. Escribir un programa de poker que reciba cinco cartas de la baraja francesa e informe (devuelva el valor lógico correspondiente) si esas cartas forman o no un poker (es decir que hay 4 cartas con el mismo número).
7. El tiempo como tuplas.
  - a. Proponer una representación con tuplas para representar el tiempo.
  - b. Escribir un programa que reciba dos tiempos dados y devuelva su suma.
8. Escribir un programa que dada una fecha expresada como la terna (Día, Mes, Año) (donde Día, Mes y Año son números enteros) calcule el día siguiente al dado, en el mismo formato.

- 
9. Escribir un programa que dada una fecha expresada como la terna (Día, Mes, Año) (donde Día y Año son números enteros, y Mes es el texto "Ene", "Feb", . . . , "Dic", según corresponda) calcule el día siguiente al dado, en el mismo formato.
10. Un analista económico utiliza un programa que toma desde una tupla valores de la cotización en pesos del dolar blue en los últimos 5 días:

```
dolarBlue = ( "189", "1930", "187", "210", "202" )
```

con el propósito de utilizarlos en un determinado informe. El programador se ha dado cuenta que necesita modificar la segunda posición del mismo, porque hay un error de entrada del dato. ¿Qué solución propone para enmendar este problema?.

11. Dada una lista de tuplas donde el elemento en la posición 0 es el DNI de la persona y en la posición 1 es el ingreso, identificar todas las personas cuyo ingreso es:
- Menor que el salario mínimo, vital y móvil (SMVM)
  - Entre un SMVM y dos SMVM
  - Entre dos SMVM y 4 SMVM
  - Mayor a 4 SMVM

Crear una lista de tuplas donde el primer elemento indica el rango de ingresos y el segundo elemento la cantidad de personas en ese segmento.

## Diccionarios

12. Escribe un programa que lea una cadena y devuelva un diccionario con la cantidad de apariciones de cada palabra en la cadena. Por ejemplo:

```
"la peluca de la abuela"  
#Debería mostrar  
{ "la": 2, "peluca": 1, "de": 1, "abuela": 1 }
```

13. Escribir un programa que toma una lista de tuplas con el formato (Alumno, Nota) donde alumno es una cadena con el nombre del alumno y nota es un número.  
Ejemplo: [('Perez', 8), ('Gonzalez', 10), ('Fernandez', 10)]

- a. Crear un diccionario donde las claves son los alumnos y el valor asociado una lista que contenga las notas del alumno.

```
Ejemplo_dict = { 'Perez': [8], 'Gonzalez': [10], 'Fernandez': [10] }
```

- b. Crear un diccionario donde las claves son las notas y el valor asociado una lista que contenga los alumnos que obtuvieron esa nota en un parcial

```
Ejemplo_dict2 = { 8: [ 'Perez' ], 10: [ 'Gonzalez', 'Fernandez' ] }
```

14. Escribir un programa que recibe un diccionario que tiene como llave a alumnos y los valores asociados una lista con las notas de parciales de los alumnos. El programa debe calcular el promedio de cada alumno e imprimirlo en pantalla. Por ejemplo para

```
alumnos = { "Juan": [6,9,8], "Manuel": [9,10,9], "Martin": [5,6,7] }
```

Debería mostrar

```
El promedio de Juan es 7.666666
El promedio de Manuel es 9.333333
El promedio de Martin es 6
```

15. Repita el ejercicio 3 pero utilizando un diccionario. Ahora un número puede repetirse varias veces.
16. Dadas dos listas donde la primera contiene nombres de personas y la segunda sus edades, generar un diccionario a partir de ellas. Investigar cómo la función `zip()` puede ayudar en la resolución
17. Dado un diccionario donde la clave es el DNI (formato cadena) y el valor es la cantidad de dosis de vacunas contra el covid, filtrar del mismo las personas que tienen 2 dosis y guardarlas en la estructura de datos que crea conveniente. Esta información será utilizada para recordarles que ya tienen disponible la tercera dosis.
18. Un investigador de historia de la informática encontró un documento sin membrete, que data de mitad de siglo XX, e incluye un mensaje en código morse que sospecha explicaría importantes sucesos históricos de la posteridad. Por lo tanto, para avanzar en su investigación es necesario descifrarlo, pero no sabe el código. Dado el siguiente diccionario que permite descifrar un mensaje morse, ayude al investigador en su tarea mediante un programa que le permita decodificar el mensaje.

```
morse = {
    'A': '.-',      'B': '-...',    'C': '-.-.',
    'D': '-..',     'E': '.',      'F': '..-.',
    'G': '--',      'H': '....',    'I': '..',
    'J': '.---',    'K': '-.-',     'L': '-...',
    'M': '---',     'N': '-.',      'O': '---',
    'P': '-.-.',    'Q': '-.-.-',   'R': '.-.',
    'S': '...',     'T': '- ',     'U': '...-',
    'V': '...-',    'W': '-.-',     'X': '-.-.-',
    'Y': '-.-.-',   'Z': '---..',   '1': '.----',
    '2': '..-.-',   '3': '...-.-',   '4': '....-',
    '5': '.....',  '6': '-.-.-',   '7': '-----',
    '8': '---..',   '9': '-----',   '0': '-----',
    '.': '.-.-.-',  ',': '---.-',   ':': '-----',
    ';': '-.-.-',   '?': '...-.-',  '!': '-.-.-',
    '"': '....-',   '" "': '-----', '+': '.-.-.',
    '-': '-----', '/': '-.-.-',  '=': '....-',
    '_': '---.-',   '$': '.....',   '@': '....-',
    '&': '....',     '(': '-.-.-',    ')': '-.-.-'
}
```

Código:

一 二 三 四 五 六 七 八 九 十 十一 十二 十三 十四 十五 十六 十七 十八 十九 二十 二十一 二十二 二十三 二十四 二十五 二十六 二十七 二十八 二十九 三十 三十一 三十二 三十三 三十四 三十五 三十六 三十七 三十八 三十九 四十 四十一 四十二 四十三 四十四 四十五 四十六 四十七 四十八 四十九 五十 五十一 五十二 五十三 五十四 五十五 五十六 五十七 五十八 五十九 六十 六十一 六十二 六十三 六十四 六十五 六十六 六十七 六十八 六十九 七十 七十一 七十二 七十三 七十四 七十五 七十六 七十七 七十八 七十九 八十 八十一 八十二 八十三 八十四 八十五 八十六 八十七 八十八 八十九 九十 九十一 九十二 九十三 九十四 九十五 九十六 九十七 九十八 九十九 一百

- 
19. Cada profe de programación 1 tiene un diccionario donde guarda el legajo de cada alumne en la clave y una lista con las notas de cada persona en el valor. Al cerrar el cuatrimestre, necesitan pasarle a alumnado un solo diccionario con estos datos, generarlo. Revisar el método `.update()` para ver como puede ayudarte a resolver el problema. Los diccionarios de cada comisión se ven así

```
com1 = {'legajo_11': [6, 7, 8], ..., 'legajo_1m': [3, 8, 9]}
com2 = {'legajo_21': [6, 7, 8], ..., 'legajo_2n': [3, 8, 9]}
com3 = {'legajo_31': [6, 7, 8], ..., 'legajo_3j': [3, 8, 9]}
com4 = {'legajo_41': [6, 7, 8], ..., 'legajo_4k': [3, 8, 9]}
```

El diccionario final debería verse así:

```
prog_1 = {'legajo_11': [6, 7, 8], ..., 'legajo_1m': [3, 8, 9],
          'legajo_21': [6, 7, 8], ..., 'legajo_2n': [3, 8, 9],
          'legajo_31': [6, 7, 8], ..., 'legajo_3j': [3, 8, 9],
          'legajo_41': [6, 7, 8], ..., 'legajo_4k': [3, 8, 9]}
```