

1. Asuma la definición parcial de una clase Lista, que va a representar una lista simple para almacenar las notas de los estudiantes de un curso:

```
class Nodo:
    def __init__(self, nom, valor):
        self.next = None
        self.nombre = nom
        self.nota = valor
class Lista:
    def __init__(self):
        self.head = None
        self.largo = 0
```

- a. Escribir los métodos `apene(nombre, nota)` que inserta un estudiante al final de la lista y `printL()` que muestra todos los estudiantes con sus notas del curso.
 - b. Escriba un método `borrar(valor)`, que recibe el nombre de un estudiante y lo elimina de la lista.
2. Escriba una función iterativa llamada `cambie(mat1, mat2)` que recibe dos matrices de tamaño $n \times m$ y cambie los valores iguales en las dos matrices por -1 en ambas matrices.

```
>>> cambie([[0,1,2],[5,5,4]], [[9,1,3],[8,5,7]])
([[0,-1,2],[5,-1,4]], [[9,-1,3],[8,-1,7]])
```

3. Escriba una función iterativa `cambie_repetidos(vector)` que reciba un vector y sustituya todos los valores repetidos por -1 (salvo la primera ocurrencia del elemento repetido).

```
>>> cambie_repetidos([1, 2, 5, 3, 1, 5, 1])
[1, 2, 5, 3, -1, -1, -1]
```

4. Escriba una función iterativa llamada `ceros(mat)` que recibe una matriz de tamaño `nxm` y devuelve las posiciones de la matriz que contienen ceros.

```
>>> ceros([[0, 1, 2, 4], [9, 8, 0, 0], [5, 0, 3, 2]])  
[[0, 0], [1, 2], [1, 3], [2, 1]]
```

5. Se va a recibir como entrada una matriz de tamaño `nxn` o cuadrada, lo cual debe verificarse. Debe programarse una función iterativa `diag_anti(matriz)` que permita obtener la diagonal y la antidiagonal de la matriz de entrada, en forma similar al ejemplo siguiente: (25 pts)

```
>>> diag_anti([[1, 2, 3], [4, 5, 6], [7, 8, 9]])  
[[1, 5, 9], [3, 5, 7]]
```

6. Escriba una función iterativa `descomponer(vector)` que reciba un vector de tamaño `n` y obtenga dos nuevos vectores, de forma que el primer elemento del vector va a ser el primero del primer vector resultante, el segundo elemento del vector va a ser el primero del segundo vector resultante y así sucesivamente. (25 pts)

```
>>> descomponer([1, 3, 5, 7, 2, 4, 6])  
[[1, 5, 2, 6], [3, 7, 4]]
```

7. Escribir una función iterativa llamada `obtener_parcial` que reciba como argumentos un vector de números y una posición inicial y una posición final y obtenga un vector conteniendo los elementos encontrados entre las posiciones dadas. Así si el vector es `[10, 20, 5, 4, 6, 8, 1]` y las posiciones son 2 y 4 el vector resultante sería `[5, 4, 6]`.

8. Escriba una función iterativa `merge(vector1, vector2)` que realiza la intercalación de los dos vectores de entrada. Los vectores DEBEN ser del mismo tamaño.

```
>>> merge([2, 4, 6], [1, 3, 5])  
[2, 1, 4, 3, 6, 5]  
>>> merge([8, 6, 2], [1, 3, 5, 1, 2, 4, 6])  
Error
```