

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 = []
        self.largo = 0
```



- a. En caso de requerir algún otro método adicional en alguna de las dos clases, debe escribirlo. Escribir los métodos `apene(self, nombre, nota)` que inserta un estudiante al final de la lista y `printL(self)` que muestra todos los estudiantes con sus notas del curso.
- b. Escriba un método `prome(self)`, que obtiene en forma recursiva, el promedio de notas de los estudiantes.
- c. Escriba un método `max(self)`, que obtiene en forma iterativa, el elemento mayor en la lista. En caso de requerir algún otro método adicional en alguna de las dos clases, debe escribirlo.
- d. Escriba un método `modificar(self, puntero, dato, ndato)`, que recibe el puntero al inicio de la lista, el dato actual y el dato a modificar, busque la primera aparición del elemento y lo modifique. En caso de requerir algún otro método adicional en alguna de las dos clases, debe escribirlo.

- e. OPCIONAL. Escriba un método `borra2(self, item)`, que elimina la segunda aparición del ítem dado en la lista. Si solo hay una aparición del elemento en la lista, no debe borrarlo. En caso de requerir algún otro método adicional en alguna de las dos clases, debe escribirlo.

2. Escribir una función iterativa `borde(matriz)` que recibe una matriz de dimensiones $n \times m$ (n = cantidad de filas, m = cantidad de columnas) de enteros y devuelve un vector con los elementos que se encuentran en el borde externo de la matriz (ver el ejemplo).

No pueden utilizarse formas directas de acceder una fila de la matriz. (20 pts)

```
>>> borde([1, 2, 3], [8, 10, 1], [5, 6, 7])  
[1, 2, 3, 1, 7, 6, 5, 8]
```

3. Se desea construir un objeto tipo camión, los atributos son: el número de placa, marca, año, estado del camión (libre, ocupado, reparación) capacidad de carga en kg. y cantidad de viajes realizados. Los métodos a implementar son: `mostrar()`, `reset()` que pone en cero la cantidad de viajes realizados, `enviar a reparación()` que actualiza el estado del camión a reparación, `recibir de reparación()` que actualiza el estado del camión a libre y `hacer viaje()` que aumenta en 1 la cantidad de viajes realizados. Además deben existir los métodos `get_capacidad()`, `get_estado()` y `get_viajes()` que retornan los datos correspondientes.

- a. Defina una clase `Camion` para el manejo de este objeto.
- b. Luego de tener la clase, defina fuera de la clase, una lista de al menos 5 instancias y escriba una función llamada `pedido`, para saber cuál o cuáles camiones podrán transportar una determinada mercadería. En este caso se recibirían como argumentos de la función: la lista de instancias, la cantidad de kg a transportar y se deben mostrar los datos de cuales camiones libres y con suficiente capacidad podrían cubrir el pedido.

4. Para representar los datos de un estudiante del TEC, se va a manejar un objeto Estudiante que permite almacenar: nombre, carnet, lista de cursos (va a iniciar en nulo al crear una instancia). La lista de cursos va a ser una lista de listas, donde cada curso se representa por un código de curso, semestre, año y nota. Un ejemplo de un curso sería [1101, 1, 2017, 60], que indicaría que el curso 1101 fue llevado por el estudiante en el semestre 1 del año 2017 y obtuvo una nota de 60. Los métodos serían:

`get_carnet()`: retorna el carnet del estudiante

`get_nombre()`: retorna el nombre del estudiante

`get_cursos()`: retorna la lista de cursos del estudiante

`agregar_curso(codigo, semestre, año, nota)`: verifica si el curso no existe en el semestre y año dados, en este caso lo incluye en la lista. En caso que el curso ya esté registrado, se debe mostrar un mensaje de error.

`mostrar()`: muestra el carnet, nombre y para cada curso: el código, semestre, año y nota obtenida por el estudiante.

- Defina una clase en Python para manejar el objeto Estudiante.
 - Defina una lista con 5 instancias de la clase Estudiante.
 - Con la lista de instancias del objeto Estudiante creada, defina una función externa a la clase llamada `promedio(carnet, lista)`, que reciba un carnet, la lista de instancias del objeto Estudiante y muestre los datos del estudiante, así como el promedio de notas. Si el carnet no existe, debe indicarse por medio de otro mensaje.
5. Se va a definir un objeto estudiante que tenga los siguientes datos: número de carnet, nombre, cursos (lista con los códigos de curso aprobados). El objeto va a contener los siguientes métodos: `mostrar()` que despliega (`print`) los datos de una instancia y `calculaNota(exa1, exa2)` que recibe la nota del estudiante en cada uno de los 2 exámenes, que tienen un valor de 50% y muestra la nota final del estudiante.
- Implementar esta clase en Python.
 - Defina una lista de 5 instancias de la clase Estudiante
 - Escriba una función `cursos(lista)` que reciba la lista de instancias anterior y muestre (`print`) el nombre y la cantidad de cursos llevados por cada estudiante.