

Ejercicios adicionales 1er Parcial

1. Dada una lista simplemente enlazada cargada con valores enteros de manera ordenada (menor a mayor), eliminar los elementos que se encuentran **repetidos**.
2. Dadas dos listas A y B cargadas con números enteros de manera ordenada, insertar en una 3er lista la **intersección** de ambas.
3. Sea la siguiente estructura de las notas de exámenes de un alumno de una determinada materia:

```
Typedef struct {  
    int nroParcial;  
    char dni[9];  
    char asignatura[20];  
    float Nota;  
}Examen;
```

Tener en cuenta que la numeración de las notas debe ser del 1 al 10.

- a. Crear la estructura de una lista doblemente enlazada que contenga como campo un dato de tipo Examen.
 - b. Declarar una lista y cargar los datos ordenados por DNI.
 - c. Eliminar de la lista aquellos nodos que contengan un DNI en particular. (Tener en cuenta que puede existir más de 1).
 - d. Realizar un algoritmo que informe el promedio de nota por DNI de una asignatura en particular.
 - e. (Rekursivo) Verificar si existe un determinado DNI en una asignatura dada.
 - f. (Rekursivo) Mostrar todos los nodos con DNI, Nro. parcial y Nota de una asignatura en particular.
 - g. (Rekursivo) Devolver la cantidad de alumnos(DNI) de una asignatura en particular. Tener en cuenta de **NO** contar dos veces el mismo DNI, pensar bien la resolución.
4. Pasar a un arreglo dinámico del tamaño justo aquellos DNI que pertenezcan a una asignatura en particular. (Utilizar función del inciso 3g)
 5. Sea la siguiente estructura de una Pila con sus valores almacenados en memoria dinámica

```
Typedef struct  
{  
    int *valores;  
    int tope;  
}Pila;
```

Recrear las funciones principales de pilas.