

MUY IMPORTANTE. LEER CON DETENIMIENTO.

1. El ejercicio será **realizado a mano** en un folio con letra clara utilizando un bolígrafo.
2. En la parte **superior izquierda** de cada folio se pondrá, en líneas distintas, el **Nombre completo**, **Nº del Dni** y **Firma del alumno**.
3. Se **tomará una fotografía con calidad por cada folio del examen**. Dicha fotografía deberá abarcar la totalidad del folio con los **datos de alumno** y la **respuesta del ejercicio**.
4. **No se tendrán en cuenta** cualquier parte de la fotográfica que **no se entienda por falta de claridad**.
5. **Toda fotografía que no cumpla lo indicado anteriormente, no se corregirá**.
6. Todas las fotografías se subirán a una tarea en Moodle en el tiempo fijado por el profesor.

EJERCICIO. Duración 60 minutos.

Dadas las siguientes estructuras de datos implemente los métodos **Cargar** y **DescartarDatos** de la clase **GestionDatos**

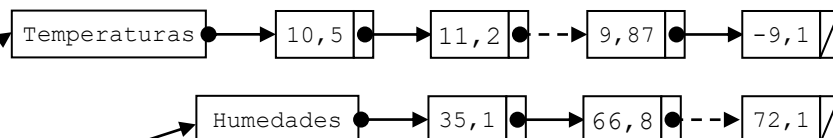
<pre>//Nodo Lista datos completa struct TNodeD { float Temperatura; float Humedad; TNodeD *Sig; };</pre>	<pre>//Nodo lista Temperaturas struct TNodeT { float Temp; TNodeT *Sig; };</pre>	<pre>//Nodo Lista Humedades struct TNodeH { float Hume; TNodeH *Sig; };</pre>
--	--	---

```
class GestionDatos //Clase que gestiona una lista simplemente circular con datos de
{ //Temperaturas y Humedades.
    TNodeD *ListaDatos;
public:
    GestionDatos(); //Constructor que inicializa a NULL el puntero ListaDatos.

    void Cargar(TNodeT **LTemp, TNodeH **LHume);
    void DescartarDatos(float TMin, float TMax, float HMin, float HMax);
};
```

Métodos a Implementar

```
void Cargar(TNodeT **LTemp, TNodeH **LHume);
```



Este método recibe dos punteros a dos listas simplemente enlazadas, una con un conjunto de temperaturas (**LTemp**) y otra con un conjunto de humedades (**LHume**). El método deberá crear la lista circular **ListaDatos** a partir de la información que contienen estas dos listas teniendo en cuenta:

- a) Las listas pasadas por parámetro no tiene por qué tener el mismo número de nodos.
- b) Se cargarán datos desde estas listas mientras ambas tengan datos.
- c) Ambas listas deberán ser eliminadas antes de finalizar el método, por este motivo **LTemp** apunta al puntero de la lista que contiene las temperaturas y **LHume** apunta al puntero de la lista que contiene humedades.

```
void DescartarDatos(float TMin, float TMax, float HMin, float HMax);
```

Este método **eliminará** todos los datos almacenados en la lista circular apuntada por **ListaDatos** si la media de las temperaturas almacenadas en la lista circular no está entre los parámetros **TMin** y **TMax** o si la media de las humedades almacenadas en la lista circular no está entre los valores de los parámetros **HMin** y **Hmax**.