



UNLaM

Dto. Ingeniería e Investigaciones Tecnológicas

E. A. R. Nro. 2– Rehacer de la Segunda Evaluación de Aprendizaje.

Debe resolver la siguiente problemática eligiendo la mejor estrategia.

Parte 1,-.POO.- Un muy simple y muy conceptual ejercicio de Programación Orientada a Objetos en Lenguaje C++. Debe completar la declaración de la clase Pers (en `pers.h`) que modela parte de los atributos de información de una persona. Una vez hecho debe asegurarse que el proyecto compila sin errores (entrando por el menú desplegable `[Build] / [Compile current file]`). Luego, complete la definición de la clase (en `pers.cpp`). Una vez hecho, compile y ejecute. En la carpeta del proyecto verá que hay un archivo de texto con la salida esperable. No debe modificar ni `main.c` ni `main.h`. Debe enviar los archivos "`pers.cpp`" y "`pers.h`" del modo indicado al final de este documento. La evaluación consiste en que la salida que se produce sea la que se muestra y se espera. Además, se verá la correctísima declaración y definición de la clase, sus métodos y funciones miembro.

Una nota aclaratoria: será preferible que el atributo de información de la clase pers (persona) para el apellido y nombre sea `string`, de lo contrario `char *`, tal como se indica en `pers.h`.

NOTE que en el proyecto entregado, se ha comenzado a declarar (con serios errores salvo en los atributos), una clase para modelar una persona. De este modo, no podrá compilar ni generar el ejecutable hasta que lo haya corregido y completado.



Parte 2.-TDA.- Se dispone de una función que permite cargar información sintética (simulada) de los exámenes finales de alumnos. Esta información está compuesta de: legajo (el DNI), apellido(s) y nombre(s), código de la materia y calificación obtenida, almacenados en orden cronológico en la lista.

Se requiere:

- La primitiva que muestre la lista del primero al último, similar a la desarrollada en clase.
- La primitiva que elimine los nodos duplicados (del mismo alumno), generando un listado agrupando por alumno y en el mismo orden en que estaban cargados y que devuelva la cantidad de nodos eliminados.
- La primitiva que elimine los nodos únicos, generando, al igual que la anterior, un listado en el mismo orden en que estaban cargados en la lista y que devuelva la cantidad de nodos eliminados.

Para que las primitivas anteriores puedan cumplir su tarea, se deben resolver dos funciones de manejo de información:

- La que permite mostrar la información del examen final antes mencionada, de modo que si recibe un puntero con NULL muestre los títulos, de lo contrario la información, similar a las vistas en el curso.
- La que permite comparar dos exámenes por su clave única y que como toda función de comparación devuelve 0 (cero) si son iguales, negativo si el primero es menor que el segundo, de lo contrario algún valor positivo. Coincidente con lo dado en el curso.

Bajo ningún concepto debe ordenar la lista.

La salida de su programa debe coincidir exactamente con la que hace el proyecto que se le entrega. No debe alterar los archivos `funciones.h` ni `main.h`. Las únicas modificaciones en `main.c` son para que invoque sus primitivas.



UNLaM

Dto. Ingeniería e Investigaciones Tecnológicas

Dispone de un proyecto que al ejecutarlo hace todo lo pedido, con lo que puede comparar los resultados que obtiene con los resultados esperados.



Consideraciones Generales (idénticas a las evaluaciones anteriores)

Descargue el proyecto provisto, genere una *carpeta* en su computadora y en ella descomprímalo.

Verá que se generan dos *subcarpetas*, una para plataforma (o compilador) de 32 bits y otro para plataforma de 64 bits.

En la carpeta que corresponde a su IDE (**EAR2-Proyecto64** o **EA2-Proyecto32**) abra el proyecto que corresponde a su plataforma (habitualmente doble click en el archivo **EA-2.cbp** o en su defecto una vez abierto el IDE, "arrastrándolo" en el mismo).

Proceda a compilarlo.

Si en el cuadro de diálogo inferior le aparecen errores:

... ¡¡¡es porque se equivocó de proyecto!!! Vaya al otro y compile:

Cuando compile el proyecto, asegúrese que el IDE genera el ejecutable con las opciones como se ve en la figura anterior.

```
gcc.exe -Wall -g -c C:(...etcétera)
```

```
gcc.exe -Wall -g -c C:(...etcétera)
```

```
gcc.exe -o bin\Debug\(...etcétera) .\lib-EAR_2_Release_bib.a
```

De otro modo, podrá tener algunos problemas en los que no le podremos ayudar.

DEBE DEVOLVER TAN SOLO los archivos **funciones.c**, **pers.cpp** y **pers.h** y al comienzo de cada uno de estos archivos, debe poner en una línea de comentario:

DNI-APELLIDO,Nombre-(curso-comisión), por ejemplo:

```
/**/* 22.333.444-PEREZ_DEL_RIO, JuanManuel-(07-2299) */**/*
```

Estos archivos deben estar comprimidos únicamente en formato **.zip (*)**. Para ello copie los tres archivos en una misma ubicación, selecciónelos y con botón derecho elija de la ventana emergente la opción: **[Enviar a] / [Carpeta comprimida (en zip)]** (o **[Send to] / [Compressed (zipped) folder]**)

EL archivo comprimido debe estar renombrado con su número de dni sin puntos, por ejemplo: **22333444.zip**