

IMPORTANTE:

- Crear un proyecto con su Apellido y nombre.
- Realizar todas las funciones que se indican, añadiendo comentarios a su código identificando con el número de ejercicio e inciso tanto a la/s función/es vinculadas con su resolución, como a la parte del Main correspondiente al ejercicio e inciso. Si no se identifica, la resolución no tendrá validez.

Una empresa que patrocina a distintos Youtubers con canales dedicados a juegos nos ha encomendado realizar un pequeño sistema.

Los datos para trabajar se encuentran almacenados en el archivo subido al campus, el cual hay que descargar y copiar dentro de la carpeta del proyecto del Parcial. Se trata de un archivo de estructuras "stYoutuber" que responde a la siguiente estructura de datos:

```
typedef struct {  
    int id;  
    char nombreCanal[40];  
    char rubro[30];           //los rubros de juegos a los que se dedica cada canal son (y estan escritos  
                             // tal cual: "fantasia", "creativos", "accion", "aventura", "retro", "construccion"  
    int cantSuscriptores;     //cant suscript total del canal  
    int cantVistasSemestre;   //cant de vistas que recibieron sus videos en el último semestre  
} stYoutuber;
```

Tenemos que desarrollar la siguiente funcionalidad:

1. Deberá recorrerse el archivo de Youtubers y construir con dichos datos una **LISTA SIMPLEMENTE ENLAZADA** de Youtubers. Para realizar esto, se deberá:
 - crear la estructura necesaria,
 - inicializar lo que corresponda,
 - y codificar las funciones que sean necesarias, **modularizando de forma correcta**.
 - Los datos deben ser agregados **en el lugar de la lista en el cual resulte más EFICIENTE realizar el proceso de agregado** (al final?, al principio?, o en orden?),
 - y la función que los agrega a la lista debe utilizar como parámetro un **PUNTERO DOBLE** a la lista.

(20 puntos)

2. Hacer una función que muestre la **LISTA SIMPLEMENTE ENLAZADA** obtenida en el punto anterior, **modularizando de forma correcta**.

- La lista deberá ser mostrada de manera **RECURSIVA**, mostrando por pantalla un mensaje que identifique cuando se ha llegado al final de la misma.
- Para ahorrar tiempo, ya se nos ha dado la porción de código correspondiente al mostrado por pantalla de los datos:

```
printf("\n ***** YOUTUBER ***** \n");  
printf("Id del Youtuber.....: %d ",p.id);  
printf("\nNombre del Canal.....: %s ",p.nombreCanal);  
printf("\nRubro del Canal (tipo de juegos).....: %s ",p.rubro);  
printf("\nCantidad de Suscriptores en miles.....: %d ",p.cantSuscriptores);  
printf("\nCantidad de vistas del semestre en miles.: %d \n\n",p.cantVistasSemestre);
```

(8 puntos).

3. En relación a la **LISTA SIMPLEMENTE ENLAZADA** obtenida en primer punto, realizar una función que permita modificar el campo cantVistasSemestre de un Youtuber a elección del usuario. Para ello debemos realizar lo siguiente:
- Una función auxiliar **RECURSIVA** que **busque** el nodo de la lista en el cual se encuentra el Youtuber a modificar.
 - Una función principal que invoque a la anterior y **modifique** el campo cantVistasSemestre del Youtuber elegido reemplazándolo por el nuevo dato (pensar en qué lugar de nuestro código debemos pedir al usuario que ingrese el Youtuber a modificar y el nuevo dato de la cantidad de vistas).

(16 puntos)

4. Debemos codificar las funciones necesarias para poder **generar una LISTA DOBLEMENTE ENLAZADA** con todos los Youtubers de la Lista Simplemente Enlazada que pertenezcan al rubro “fantasía”, **BORRANDO los mismos de la Lista Simple original**. Para ello se deberá:
- crear la estructura necesaria,
 - inicializar lo que corresponda,
 - y codificar las funciones que sean necesarias, **modularizando de forma correcta**.
 - En la Lista Doblemente Enlazada los datos deben ser agregados **en el lugar de la lista en el cual resulte más EFICIENTE realizar el proceso de agregado** (al final?, al principio?, o en orden?),
 - A los fines de borrar los nodos de la Lista Simple Original correspondientes a los Youtubers del rubro “fantasía”, **deberá codificarse la función auxiliar de “borrarNodo”, la cual deberá recibir por parámetro un PUNTERO DOBLE a la lista**.

(20 puntos).

5. Hacer una función que muestre la **LISTA DOBLEMENTE ENLAZADA** obtenida en el punto anterior, **modularizando de forma correcta** y teniendo en cuenta el código ya realizado en el punto 2.
- **La lista deberá ser mostrada AL REVÉS de manera ITERATIVA**. Para ello, pensar qué parámetros deberá recibir.

(10 puntos)

6. Como el Youtuber con más cantidad de suscriptores de la **LISTA DOBLEMENTE ENLAZADA** obtenida en el punto 4 será premiado con un plus equivalente al 20% de su cantidad de suscriptores, debemos realizar:
- Una **función auxiliar** que **busque** el Youtuber de la lista con **mayor** cantidad de suscriptores.
 - Una **función principal** que invoque a la anterior y **calcule** el plus a abonar.

(16 puntos).

7. Hacer una función main()
- Para hacer esto, cree las variables que considere necesarias e invoque las funciones (de forma directa o indirecta) como corresponde en cada caso.
 - Muestre los resultados cada vez que sea necesario.
 - A fin de identificar cada inciso, comente su código indicando a qué apartado corresponde, por ejemplo: // Ejercicio 3.a

(10 puntos)