

IMPORTANTE:

- Crear un proyecto con su Nombre y Apellido.
- Realizar todas las funciones que se indican.
- Añadir comentarios a su código identificando con el número de ejercicio e inciso a la/s función/es vinculadas a la resolución. 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;
```

En el sistema a desarrollar se utilizarán las siguientes estructuras:

```
typedef struct _nodoS {
    stYoutuber dato;
    struct _nodoS * siguiente;
} nodoS;

typedef struct _nodoD {
    stYoutuber dato;
    struct _nodoD * siguiente;
    struct _nodoD * anterior;
} nodoD;

typedef struct {
    nodoD * inicio;
    nodoD * fin;
}Fila;
```

Se nos pide desarrollar la siguiente funcionalidad:

1. Utilizando el archivo de Youtubers, hacer una función que recorra el mismo y construya una **FILA de Youtubers trabajada con LISTA DOBLE**. Para realizar esto, deberá **modularizar** y codificar las siguientes funciones:
 - a) Una función auxiliar que cree el nodo de la Lista Doble, la cual debe codificarse con **PUNTERO DOBLE**. (2 puntos)
 - b) Una función auxiliar que agregue el Youtuber a la Fila. Pensar bien en qué lugar de la Fila deberá ser agregado. (5 puntos)
 - c) Un subprograma o función principal, que, por cada registro leído del archivo, lo agregue a la Fila. Recordar inicializar lo que corresponda. (8 puntos)
2. Hacer una función que muestre la **FILA** obtenida en el punto anterior, **modularizando** de la siguiente manera: (8 puntos)
 - a) La función mostrar de **Fila** invocará a la función mostrar de Lista Doble.
 - b) La función mostrar de **Lista Doble** deberá **mostrar la lista AL REVÉS** de manera **ITERATIVA**. Para ello, pensar qué parámetros deberá recibir.
 - c) La función mostrar de Lista Doble deberá invocar a la función **mostrar un Youtuber**.

3. La empresa patrocina a cada Youtuber con un monto en dinero equivalente al 20% de su cantidad de vistas por semestre, y quiere calcular el monto total de patrocinios de cada rubro. Para eso, hay que realizar una **única función** que recorra **una sola vez** la **FILA**, y sumar el total a abonar en cada rubro (sumar el 20% del campo cantidad de vistas de cada Youtuber del rubro). **Deberán retornarse los totales a abonar en c/u de los 6 rubros; pensar cómo hacerlo. (20 puntos).**
4. La empresa también quiere calcular un PLUS por cantidad de vistas por semestre que abona sólo a los Youtubers del rubro “retro”. Para eso debemos hacer lo siguiente:
- a) Pasar a una **LISTA SIMPLE** todos los Youtubers del sector “retro”, **sin borrar los datos de la Fila**, y **ordenados de menor a mayor** de acuerdo a su cantidad de vistas por semestre. **(17 puntos).**
 - b) Realizar una función **RECURSIVA** que retorne la cantidad de Youtubers que tiene dicha **LISTA SIMPLE** que superen una cantidad de vistas por semestre ingresada por parámetro **(15 puntos).**
 - c) Realizar una función **RECURSIVA** que calcule el total a pagar en concepto de PLUS a **todos** los Youtubers de la **LISTA SIMPLE**, para lo cual deberá sumar \$10000 por cada Youtuber que NO supere el monto de vistas establecido, y \$20000 por cada Youtuber que sí lo supere. **(17 puntos).**
5. Hacer una función main() **(8 puntos)**
- 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