

IMPORTANTE:

- Crear un proyecto con su Nombre y Apellido.
- Agregar la librería de Pilas (en su versión original).
- Realizar todas las funciones que se indican.
- Añadir comentarios a su código identificando cada inciso.
- Desarrolle un main() e invoque únicamente a las funciones que compilen correctamente. La resolución del mismo es obligatoria y sumará puntos extras en caso de necesitarlos.

Sea la siguiente estructura de datos:

```
typedef struct
{
    int id;
    int código;
    int rubro; // 1 – Televisores / 2 – Lavarropas / 3 – Cocinas / 4 - Calefactores
    char marca[30];
    char modelo[30];
    float precio;
} stArticulo;
```

Obtenido	Valor	Inciso
	10	1. Utilizando el archivo de artículos que ya está cargado ("articulos.dat"), hacer una función que pase a un arreglo de artículos , todos los artículos de un determinado rubro. La función recibe por parámetro el arreglo de artículos, la dimensión del mismo y el id_rubro . Debe retornar la cantidad de celdas cargadas en el arreglo.
	5	2. Hacer una función que muestre el arreglo de artículos por pantalla. Modularizar .
	15	3. Hacer una función que ordene el arreglo de artículos por el método de selección , usando el código como criterio de ordenación. Modularizar .
	10	4. Crear una función que devuelva la cantidad de artículos cargados de un archivo. NO debe recorrerlo todo para contar los registros.
	15	5. Hacer una función que reciba como un número de registro (posición dentro del archivo) . La función debe mostrar por pantalla los datos del artículo almacenado en esa posición . Debe verificar que la cantidad de registros almacenados en el archivo sea mayor que el número ingresado (o sea, no ir más allá de los límites del archivo). Se debe invocar a la función anterior (antes de hacer fopen en esta función) .
	15	6. Hacer una función que, utilizando el arreglo cargado previamente , pase a dos nuevos archivos ("articulosEconomicos.dat" y "articulosCostosos.dat") todos los artículos, el filtrado se realiza a partir de un precio enviado por parámetro. La función recibe toda la información necesaria para trabajar. Dentro de la función trabaje con 2 variables de tipo FILE*, una para cada archivo.
	10	7. Hacer una función que muestre por pantalla el contenido del archivo . Modularizar .
	10	8. Hacer una función que lea el archivo de artículos y pase los id a una pila. Recordar que la pila tiene un límite de 50 elementos.
	10	9. Hacer una función que elimine los elementos de las posiciones pares de la pila.
	----	10. Hacer un main () que demuestre un correcto funcionamiento de las funciones.

Tabla de puntuación:

Obtenido	10	20	30	40	50	60	70	80	90	100
Nota	1	2	3	4	5	6	7	8	9	10
	DESAPROBADO					APROBADO				