

Ejercicio 1

Robin es un joven titan que lee mucho, tiene una biblioteca llena de libros, como además también le gusta fomentar la lectura, presta sus libros a otros jóvenes. Cada vez que compra un libro lo anota en una planilla y obviamente su biblioteca difiere de esa planilla (por todos los que prestó).

Suponga que se tienen dos vectores de tipo `libro_t`, uno representa la planilla (todos los libros que compró) y otro representa su biblioteca actual.

```
typedef struct libro {
    char nombre[MAX_NOMBRE];
    char autor[MAX_NOMBRE];
    int cant_paginas;
} libro_t;
```

Ambos vectores están ordenados alfabéticamente por nombre.

Se pide

1. Crear una función que devuelva un vector con los libros prestados.

Ejercicio 2

Puro Hueso está cansado de no estar a la moda, y decidió investigar para comprarse un celular último modelo. Para esto, en un archivo anotó las especificaciones que cree necesarias que tenga, en el siguiente formato:

```
CAMARA=<numero>MP
BATERIA=<numero>mAh
TAMANIO=<numero>"
MARCA=<nombre>
```

Por suerte, CeluMania nos comentó que solo tenía en stock 15 celulares, y nos ofreció un archivo de texto con todos ellos. Cada línea del archivo tiene el siguiente formato:

```
<modelo>;<marca>;<procesador>;<batería>;<tamaño_pantalla>;<cámara>
```

Se pide

1. Imprimir por pantalla aquellos celulares que cumplan con al menos 3 de las especificaciones pedidas por Puro Hueso.

Ejercicio 3

Dexter está quiere continuar la partida de la batalla naval que estaba jugando contra su Computadora y necesita cargar la matriz del terreno con las jugadas. Para eso tiene guardados 2 archivos, por un lado tiene uno con las coordenadas de todos los barcos y por otro tiene un archivo con las coordenadas de los tiros que su contrincante le hizo. Ambos son archivos de texto y tienen el siguiente formato.

```
fila;columna
```

Se pide

1. Dados 2 archivos de texto, uno con los barcos y otro con los tiros, crear un procedimiento que devuelva la matriz cargada con la partida para que sigan jugando.

Ejercicio 4

Bombón, Burbuja y Bellota estaban jugando al TEG (juego que se basa en conquistar países), cuando a Bellota le dieron ganas de ir al baño. Al volver, se dio cuenta que no todo estaba como lo había dejado, había fichas cambiadas. Pero por suerte, fue más viva que las hermanas y sacó una foto al tablero antes de irse.

Se cuentan con dos matrices de `pais_t`, una con el tablero antes y otra actual.

```
typedef struct pais {
    int cantidad_fichas;
    char color_ficha;
} pais_t;
```

Se pide

1. Crear una función que deje a la matriz actual como antes y que devuelva la cantidad total de fichas que se tuvieron que sacar/poner en el cambio.

Acá hay una tabla con algunos cambios de ejemplo:

Actual	Antes	Cantidad de fichas cambiadas
5 Amarillo	7 Rojo	12 (se sacaron 5 amarillas y se pusieron 7 rojas)
4 Amarillo	2 Amarillo	2 (se sacaron 2 amarillas)
5 Azul	9 Azul	4 (se pusieron 4 azules)
8 Rojo	2 Azul	10 (se sacaron 8 rojas y se pusieron 2 azules)

Ejercicio 5

1. Escribir en pseudocódigo el algoritmo base para hacer recursividad. Qué partes fundamentales se requieren?
2. Qué son las variables? Qué diferencia y qué similitud tienen con las constantes?
3. Para qué casos recomendarías usar archivos, y para qué casos vectores? Por qué?