

Ejercicio 1

Como el laboratorio de Dexter tiene muchas habitaciones, decidió distribuir a sus robots de manera que cada uno trabaje en una sola habitación. Se tiene un archivo de texto llamado **inventario_robots.csv**, **ordenado alfabéticamente por nombre de habitación y luego por id del robot**. Cada línea del archivo contiene la siguiente información:

```
habitacion;id_del_robot;estado
```

1. Se necesita controlar el rendimiento de estos robots, para lo cual Dexter quiere listarlos agrupados por habitación, y luego por id del robot de manera ascendente, mostrando su estado. Al llegar al último robot de la habitación, se necesita mostrar la cantidad total de activos en dicha habitación.

Ejemplo:

```
Pecera
056423: Activo.
980564: En descanso
102141: Activo.
En la habitacion Pecera se encuentran 2 robots activos.
```

```
Sala de máquinas
252564: Activo.
411029: Activo.
696915: En descanso.
717090: En descanso.
717890: Activo.
En la habitacion Sala de máquinas se encuentran 3 robots activos.
```

Luego de ver el resultado, Dexter se dio cuenta que es poco humano tratar a los robots por su **id**, por lo que decidió ponerle un nombre a cada uno. Si ahora se cuenta con un archivo **binario de acceso aleatorio** que contiene todos los nombres de los robots, **ordenado por id del robot**, donde dichos **id** son consecutivos y empiezan desde el 1, con la siguiente estructura:

```
typedef struct robot {
    int id;
    char nombre[MAX_NOMBRE];
} robot_t;
```

2. ¿Cómo modificarías tu programa para mostrar en el listado del ítem anterior, el **nombre** del robot en vez del **id**?

Ejercicio 2

Billy y Mandy planean molestar a Puro Hueso (como hacen siempre) pero esta vez, de una manera distinta. Se les ocurrió hablar por mensajes, y que éstos esten cifrados. Por lo que volcaron en el archivo binario **mensaje_a_cifrar.dat** frases, con la respectiva letra a quitar de cada una de ellas. La estructura hallada en el archivo es la siguiente:

```
typedef struct mensaje {
    char frase[MAX_FRASE];
    char letra;
} mensaje_t;
```

Donde *letra* simboliza la letra a quitar de la frase.

Ejemplo:

Si se recibe la frase **esto es algo1mendez** y la letra **e**, el mensaje cifrado quedaría: "sto s algo1mndz".

Se pide

1. Crear un procedimiento que a partir de **mensaje_a_cifrar.dat**, elimine la letra indicada, y escriba el mensaje cifrado en un archivo de texto **mensaje_cifrado.txt**.

Ejercicio 3

A las chicas superpoderosas les encanta llenar álbumes de figuritas, tanto, que por día cada una se compra 5 paquetes de figuritas.

Una tarde lluviosa, decidieron llevar control de todas las figuritas que tenían, por lo cual cada una empezó a anotarlas ordenadamente de menor a mayor, por número en un vector. Como Burbuja es muy distraída, se le volaron todas las figuritas y se le cayeron en un charco de agua. Pero como tiene muy buenas hermanas, decidieron darle las figuritas que tengan repetidas a ella, para que no se ponga triste.

Se pide

1. A partir de los 2 vectores recibidos de Bombón y Bellota, crear un procedimiento que cree un nuevo vector con las figuritas repetidas, para que se las puedan dar a Burbuja.
2. Si las chicas quieren tener anotadas todas las figuritas que tienen sin repetir, qué cambiarías del algoritmo anterior? Cómo se llama esa operación?

Ejercicio 4

Raramente, *Ed*, *Edd* y *Eddy* están teniendo una racha de buena suerte, han conseguido ahorrar mucho dinero y poder comprar muchos caramelos.

```
typedef struct caramelo {  
    int radio;  
    char sabor[MAX_SABOR];  
    char color[MAX_COLOR];  
} caramelo_t;
```

Sin embargo, esta buena suerte no duró mucho para Edd ya que se encuentra con dolor de muela. Este no va a permitir que sus amigos se terminen todos los caramelos, por lo que su dentista le permitió comer caramelos de hasta un diámetro de 10 cm.

Además, a él solo le gustan los de sabor **Sandía**, **Cereza** y **Limón**, y respecto a su color come cualquiera, salvo que sea **marrón**.

Si estos tres amigos guardaron sus caramelos en una matriz de **caramelo_t**.

Se pide

1. Crear una rutina **recursiva** que obtenga la cantidad de caramelos que Edd podrá comer a pesar de su dolor de muela.

Ejercicio 5

Justificar las siguientes falsedades: 1. Siempre es mejor tener los datos en archivos, en lugar de vectores. 2. Es mejor utilizar búsqueda binaria que búsqueda lineal. 3. Para usar operaciones entre vectores, no es necesario que estos estén ordenados. 4. *fwrite* y *fprintf* son equivalentes.