

Ejercicio 1

Scooby Doo y sus amigos, se dedican a capturar monstruos y villanos de todo tipo, y siempre llevaron registro de sus victorias y, para las últimas, le delegaron la responsabilidad a Shaggy de actualizar este registro.

Se dieron cuenta que en vez de hacer lo que le pidieron, varias veces armó un buen sandwich, y ahora quedarón algunas victorias sin pasar.

Cada victoria se almacena en una estructura del tipo:

```
typedef struct victoria {
    char nombre_enemigo[MAX_NOMBRE];
    bool es_monstruo;
    int dificultad;
} victoria_t;
```

Se pide

1. Dado el vector de victorias original y un vector que representa la victorias que Shaggy se "olvidó" de pasar, las agregue al vector original manteniendo el orden. Ambos vectores están ordenados alfabéticamente según el nombre del monstruo o villano.

Ejercicio 2

Número Uno se está por jubilar, y los chicos del barrio necesitan un nuevo líder (nananananananana lliideeerrr). Decidieron elegir al sucesor en base a la experiencia de cada postulante. Para esto hicieron un censo y obtuvieron un archivo de tipo **csv** en donde guardan las misiones de cada chico. Cada misión está guardada de la siguiente forma:

```
nombre_postulante;nombre_mision;dificultad
```

Se pide

1. Sabiendo que, Los chicos del Barrio no son más de 20 y que las misiones son muuuuuuchas. Crear un algoritmo que devuelva un vector de los postulantes, ordenado descendentemente por cantidad de misiones. **Crear las estructuras que considere necesarias**.

Ejercicio 3

La escuela primaria Endsville está teniendo problemas de organización y necesitan una forma de buscar a los alumnos que aprobaron el exámen general.

La escuela cuenta con estos dos archivos:

- Archivo de alumnos: Un **archivo binario de acceso secuencial** llamado `alumnos.dat` que contiene a elementos de tipo `alumno_t`:

```
typedef struct alumno {
    char nombre[MAX_NOMBRE];
    int id_maestro;
    int nota;
} alumno_t;
```

- El campo `id_maestro` tiene el id del maestro que le corrigió el exámen.

- Archivo de maestros: Un **archivo binario de acceso directo** llamado `maestros.dat` que contiene elementos de tipo `maestro_t`:

```
typedef struct maestro {
    int id;
    char nombre[MAX_NOMBRE];
    bool es_malo;
} maestro_t;
```

- El archivo está ordenado por `id` de forma ascendente. Los `id` son números enteros correlativos que comienzan desde el 1.

Se pide

1. Implementar una función que imprima el nombre de todos los alumnos que aprobaron (se aprueba con 6 o más) **junto con el nombre del maestro que le corrigió el exámen**.

Ejercicio 4

Los chicos del barrio entrenan todos los días arduamente para mantenerse en forma y poder derrotar a sus enemigos. Por esto es que, entre otras cosas, juegan a la batalla naval (número uno dice que ayuda a la estrategia). La batalla naval que juega consiste en dos terrenos que se pueden ver como matrices de caracteres. En un terreno se posicionan los barcos (B) y en otro las bombas (T).

Se pide

1. Implementar una función que dado estas dos matrices devuelva **cuántos barcos están vivos aún**.

Ejercicio 5

1. Cuáles son las ventajas y desventajas de trabajar con archivos?
2. Qué condición se tiene que dar para poder hacer una operación entre 2 vectores? Por qué? Qué pasa si no se aplica esta condición?
3. Explicar con tus palabras qué es programar.