

Archivos y TDAs

—

Ejercicio 1

Se acerca el mundial, y Bart no se quiere quedar afuera a la hora de llenar el álbum de figuritas.

Como es medio despistado, nos pide ayuda para completarlo.

En principio, Bart tiene un archivo de figuritas con el siguiente formato

```
nombre_jugador;equipo;numero
```

Se tiene el siguiente TDA álbum:

```
#ifndef ALBUM_FIGURITAS_H
#define ALBUM_FIGURITAS_H

#include <stdbool.h>

#define MAX_NOMBRE 100
#define MAX_EQUIPO 100

typedef struct figurita {
    int numero;
    char nombre_jugador[MAX_NOMBRE];
    char equipo[MAX_EQUIPO];
} figurita_t;

typedef struct album {
    figurita_t* figuritas;
    int cantidad_figuritas;
} album_t;

/*
Post: Crea un álbum vacío.
Devuelve un puntero al álbum o NULL en caso de error.
*/
album_t* album_crear();
```

```

/*
Pre: `album` fue creado usando album_crear().
Post: Agrega una figurita al álbum.
      Devuelve true si se pudo agregar,
      false en caso contrario.
*/
bool agregar_figurita(album_t* album, figurita_t figurita);

/*
Pre: `album` fue creado usando album_crear().
Post: Busca en el álbum la figurita con el `numero` indicado.
      Si `figurita` no es NULL y existe, guarda en él una copia.
      Devuelve true si la figurita está en el album, false en caso contrario.
*/
bool obtener_figurita(album_t* album, int numero, figurita_t* figurita);

/*
Pre: `album` fue creado usando album_crear().
Post: Elimina del álbum la última figurita agregada y, si
      `figurita` no es NULL, guarda en él una copia de la figurita quitada.
      Devuelve true si se eliminó, false si el álbum está vacío o hubo error.
*/
bool quitar_figurita(album_t* album, figurita_t* figurita);

/*
Pre: `album` fue creado usando album_crear().
Post: Destruye el álbum y libera toda la memoria utilizada.
*/
void album_destruir(album_t* album);

#endif

```

Además, para abrir los sobres de figuritas se tiene la siguiente biblioteca:

```

#ifndef SOBRE_H
#define SOBRE_H

#include <stdbool.h>
#include "album.h"

```

```

typedef struct sobre {
    figurita_t* figuritas;
    int cantidad_figuritas;
    bool abierto;
} sobre_t;

/*
Pre: `sobre` fue inicializado con `cantidad_figuritas` > 0 y `abierto` en false.
Post: Abre el sobre y carga en el vector `figuritas` una cantidad de figuritas
      igual a la indicada en `cantidad_figuritas`.

      Las figuritas se obtienen aleatoriamente desde un archivo
      csv interno.

      El sobre debe estar cerrado para poder abrirse.

      Devuelve true si se pudieron cargar las figuritas,
      false en caso contrario.

      Se debe liberar la memoria de las figuritas usando free.
*/
bool abrir_sobre(sobre_t* sobre);

#endif

```

1. Dado el archivo de figuritas de Bart, que se recibe por **línea de comandos**, crear un TDA álbum y guardar las figus ahí.
2. Bart recibió un regalo de su tía Selma: unos sobres con figus. Los sobres están en un archivo con el siguiente formato:

```
cant_figus;estado
```

donde estado puede ser 'A'(abierto) o 'C' (cerrado).

Este archivo se va a recibir **opcionalmente** como argumento por línea de comando después del archivo de figus de Bart.

- Crear un procedimiento que lea el archivo de sobres y guarde en el TDA álbum sólo las figuritas que no tenga repetidas.
- Cambiar el procedimiento para que aquellas figuritas que estén repetidas las guarde en un archivo llamado repes_bart.csv.

3. Milhouse tiene un archivo csv con sus repetidas (repetidos_milhouse.csv).
Crear un procedimiento que, dadas las repetidas de Milhouse y el tda
álbum de Bart cree dos archivos de figuritas:
late.csv que guardará las figuritas de Milhouse que Bart ya tiene y
nola.csv que guarde las figs de Milhouse que Bart no tiene aún.
4. Bart se dio cuenta que pegó mal todas las figuritas de Australia.
Crear una función que sirva para quitar todas las figuritas del álbum
de un equipo determinado.
5. Se pide que al finalizar el programa, se guarden las figs actuales
en el archivo de figs.