

# EXAM

## Fifth day

---

*Lecturer: Concha Batanero*

The application to develop is about a discs collection. Please, take into account the next considerations:

- Please, send to me a zip file named: Exam\_your name\_your lastname.zip. For example:

Exam\_Concha\_Batanero.zip

The zip file must contain all the code files (.c and .h). It must be sent to me before 13:00.

- Read carefully the exercise.
- Do a **graphic**.
- Read again and make sure that the graphic is correct.
- Read as many times as you need. It is normal you have to read it several times.
- **Global variables are not allowed.**
- You must use the **Memory Manager** library and organize the code in several files.

## Discs collection application

It is wanted to develop an application in order to manage a discs collection. The following data type has been designed for this end:

```
typedef struct
{
    char title[MAX_CHARAC_TITLE];    // Example: "THE ESSENTIAL"
    int year;
    char nameAuthor[MAX_CHARAC_NAME];
    int numSongs;                    // The songs number
    char ** ppSong;                  // The songs name
}tDisc;
```

Each *tDisc* structure stores the data of a music disc. The *numSongs* member refers the number of elements of the array of pointers to *char* (*ppSong*).

The header file of the program contains the following declarations:

```
// All include directives that are needed

// Constants

#define    MAX_CHARAC_NAME    20
#define    MAX_CHARAC_TITLE    30

// The referred data type declaration

// Prototypes of the functions
```

Using this data type it is asked to write a program that builds the dynamic array of music discs, as well as all the dynamic arrays that every *tDisc* structure contains. The program must show the following menu and execute each of its options:

1. Add disc
2. Display information of the discs
3. Display the discs of an author using his name
4. Delete disc
5. Exit

### 1. Add disc

The option add a disc involves adding an element to the dynamic array of discs and read the data of the disc from keyboard, including the names of all his songs. Organize the code for this option in several functions. Think about the necessary actions to be carried out to be able to store all the information. Think also about the prototype of the functions for an efficient execution of the code.

## 2. Display information of the discs

The data of all discs must be must be displayed, including the name of all songs.

## 3. Display the discs of an author using his name

Program the function with the following prototype:

```
void PrintDiscsOfAuthor(tDisc *pDisc, int nDiscs);
```

The function receives a pointer to the discs array and the number of the discs. It reads from keyboard the name of the author, to which his discs will be displayed. If the author does not exist in the array, the function will print a message notifying it. Otherwise, all the discs of this author must be displayed.

## 4. Delete disc

Program the function *DeleteDisc()* that deletes a disc with a specific title of the discs array. His prototype is the following:

```
int DeleteDisc(tDisc *pDisc, int *pnDiscs);
```

The function receives the pointer to the dynamic array of discs. Also it receives by reference the number of discs. It returns the value *0* if the disc is correctly deleted and the value *-1* if the disc doesn't exist in the array. This function uses the function *LookForDisc()* which prototype is the next:

```
int LookForDisc(tDisc *pDisc, int nDiscs);
```

The function reads from keyboard the title of the disc and looks for it in the array that receives in its parameters. It returns the position of the disc in the array, if the disc is found. Otherwise, it returns *-1*.

Note: The running of this application must include the freeing of the memory when is necessary along the code.