

**Relación de prácticas de la asignatura
METODOLOGÍA DE LA PROGRAMACIÓN
Segundo Cuatrimestre
Curso 2019-2020**

1º Grado en Informática

Práctica 3: Recursividad, Ficheros y Argumentos en Línea de órdenes

Objetivos

- Practicar conceptos básicos sobre recursividad, ficheros de texto y ficheros binarios en C.
- Usar argumentos en línea de órdenes para que los programas puedan recibir argumentos en el momento de ser ejecutados.

Recomendaciones

- Pasa a tus funciones el nombre del fichero como parámetro y ábrelo y ciérralo dentro de la función.
- Organiza los ejercicios 8 y 11 en al menos tres ficheros *.c* (*main.c*, *ficheros.c*, *util.c*) y dos ficheros *.h* (*ficheros.h* y *util.h*).

Temporización

3 sesiones de prácticas

- 1ª sesión: Ejercicios de recursividad
- 2ª sesión: Ejercicios de ficheros de texto
- 3ª sesión: Ejercicios de ficheros binarios

Recursividad

1. Escribe una solución recursiva que calcule el algoritmo de *Euclides*, usado para calcular el máximo común divisor de dos enteros. El algoritmo de *Euclides* se describe del siguiente modo:

```
mcd(x, y) = x, si y = 0  
mcd(x, y) = mcd(y, mod(x,y)) si y > 0 y x >= y
```

2. Construye una función recursiva que calcule la división entera de dos números mediante el métodos de restas sucesivas. Implementa un pequeño programa para probarla.
3. Dada una cadena *c*, diseña una función recursiva que cuente la cantidad de veces que aparece un carácter *x* en *c*.
Ejemplo: para *c* = “elementos de programacion” y *x* = 'e', el resultado es 4.
4. Codifica una función recursiva que permita sumar los dígitos de un número. Implementa un programa para probarla. Ejemplo: Entrada: 124 Resultado: 7.

Ficheros de texto

5. Codifica un programa en C que, utilizando funciones, abra un fichero de **texto**, cuyo nombre se pedirá al usuario, y genere un nuevo fichero llamado *mayusculas-nombreFicheroDeEntrada.txt*, que tenga el mismo contenido que el fichero original pero en mayúsculas.
6. Codifica un programa en C que, utilizando funciones, cree un fichero **texto** con números enteros generados aleatoriamente en un intervalo especificado por el usuario. El programa le preguntará al usuario el nombre del fichero a crear, el número de elementos que contendrá el fichero y los extremos superior e inferior del intervalo.
7. Codifica un programa en C que, utilizando funciones, calcule la media de los elementos pares que hay en un fichero de texto generado por el ejercicio anterior. El nombre del fichero se pasará como argumento en la línea de órdenes.
8. Construye un programa que gestione mediante ficheros de **texto** el *stock* de libros de una librería. Para cada libro se almacenarán tres líneas en un fichero de texto (*stock*):
 - en la primera línea el *título*,
 - en la segunda línea el *autor*,
 - y en la tercera línea el *precio* y las *unidades* disponibles del libro.El programa contará con un **menú** que permitirá realizar las siguientes operaciones:
 - Comprobar la existencia de un determinado libro en el *stock* buscando por su título.
 - Introducir un nuevo libro en el *stock*.
 - Contar el número de libros (títulos) diferentes que hay en el *stock*. Considera que no puede haber títulos de libros repetidos en el fichero.
 - Listar los libros en el *stock* almacenándolos previamente en un vector dinámico.
 - Vender *n* unidades de un libro buscándolo por su título. Si hay menos de *n* unidades en el *stock*, se venderán solo las unidades disponibles.
 - Borrar aquellos registros con 0 unidades disponibles.
 - Salir.El nombre del fichero se pasará como argumento en la línea de órdenes.

Ficheros binarios

9. Codifica un programa en C que, utilizando funciones, cree un fichero **binario** con números enteros generados aleatoriamente en un intervalo especificado por el usuario. El programa guardará los números en un vector dinámico antes de volcarlo a disco. El programa recibirá 4 parámetros como argumentos en la línea de órdenes: nombre del fichero a crear, número de elementos que contendrá el fichero y los extremos superior e inferior del intervalo.
10. Codifica un programa en C que, utilizando funciones, lea números enteros desde un fichero binario generado en el ejercicio anterior, almacene sus valores en un vector dinámico y calcule la media de los números pares. El nombre del fichero se le preguntará al usuario.
11. Construye un programa que gestione, mediante ficheros **binarios**, el *stock* de libros de una librería. Para cada libro se almacenará la siguiente estructura:

```
struct libro {  
    char titulo[100];  
    char autor[50];  
    float precio;  
    int unidades;  
}
```

El programa contará con un **menú** que permitirá realizar las siguientes operaciones:

- Comprobar la existencia de un determinado libro buscando por su título.
- Introducir un nuevo libro en el *stock*.
- Contar el número de libros (títulos) diferentes que hay en el *stock*. Considera que no puede haber títulos de libros repetidos en el fichero.
- Listar los libros en el *stock* almacenándolos previamente en un vector dinámico
- Vender n unidades de un libro buscándolo por su título. Si hay menos de n unidades en el *stock*, se venderán solo las unidades disponibles.
- Borrar aquellos registros con 0 unidades disponibles.
- Salir.

El nombre del fichero se pasará como argumento en la línea de órdenes.