



## TRABAJO PRÁCTICO Nº4

Realizá cada ejercicio en lenguaje C siguiendo las indicaciones de la cátedra. En los casos en los que se requiera, desarrollá un planteo de solución (PS) y una prueba de escritorio (PE) antes de compilar y correr el programa. Los primeros ejercicios proponen una definición de la estructura de registro requerida y la muchos implican modificar lo que realizaste en los prácticos anteriores, por lo que si no los hiciste... ¡este es el momento!

**Ejercicio 1.** Escribí un programa que reciba del usuario una fecha en formato dd/mm/aaaa (dos dígitos para el día, dos para el mes y cuatro para el año) y almacene cada dato por separado en un registro. Al finalizar, debe imprimir la fecha en formato dd-mm-aaaa.

```
struct Fecha {  
    int dia;  
    int mes;  
    int ano;  
};
```

**Ejercicio 2.** Utilizando las funciones presentes en *time.h* obtené el día actual e imprimilo por pantalla con el formato "Hoy es dd/mm/yyyy".

**Ejercicio 3.** Utilizando lo realizado en los ejercicios 1, 2 y 3 del práctico 3 realizá un programa que reciba del usuario nombre, apellido y fecha de nacimiento (dd/mm/aaaa), calcule la edad y guarde todos los datos en un registro. Luego debe imprimir todos los datos en una sola cadena con el siguiente formato: "La persona *nombre apellido* nació el día *dd-mm-aaaa* y tiene *n* años de edad". Utilizá al menos una función para completar la estructura, otra para calcular la edad (obteniendo la fecha actual con lo realizado en el ejercicio 2) y otra para imprimir por pantalla la cadena.

```
struct Fecha {  
    int dia;  
    int mes;  
    int ano;  
};  
  
struct Persona {  
    char nombre[50];  
    char apellido[50];  
    struct Fecha fechaNacimiento;  
    int edad;  
};
```

**Ejercicio 4.** Modificá el programa del ejercicio anterior para que permita ingresar una serie de 5 personas en un arreglo con todos los datos requeridos. Al insertar una persona debe hacerlo en orden alfabético por apellido. Si dos personas tienen el mismo apellido debe ordenar por el nombre y si tienen el mismo apellido y nombre debe hacerlo por la fecha de nacimiento.

**Ejercicio 5.** Escribí un programa que permita mantener un catálogo de productos con la siguiente estructura:

```
struct Producto {  
    char codigo[10];  
    char nombre[50];  
    int stock;
```



```
        float precio;  
    };
```

El programa debe permitir:

1. Agregar un producto cuyo código no haya sido ingresado.
2. Buscar un producto por su nombre e imprimir en pantalla sus datos.
3. Imprimir en pantalla todo el catálogo.
4. Salir.

Utilizá un arreglo de 100 posiciones para el catálogo. Si se intentá insertar un producto y el arreglo está lleno, se debe dar un error.

**Ejercicio 6.** Modificá el programa del ejercicio anterior para que implemente el stock mediante una unión que contenga el dato de unidades (entero) o kilos/litros (flotante). Se debe agregar a la estructura un indicador que ayude a determinar cuál es el tipo de unidad utilizado en cada producto.