

## Estructuras

1. Dadas las siguientes declaraciones de un programa:

```
#define N 80
...
struct persona{
    int codigo;
    char nombre[20];
    char telef[10];
} p1;
struct listin{
    int tam;    //cantidad de valores que hay en el vector datos
    struct persona datos[N];
} mio, tuyo;
...
int v1[5], v2[10], m, i;
char cad1[20], cad2[20];
```

//suponemos que hay almacenado algún valor válido en las variables que sean necesarias.

- Indicar cuáles de las siguientes sentencias de la función **main()** son correctas, si fuesen incorrectas indicar el por qué:
  - a) mio=tuyo;
  - b) mio.tam=tuyo.tam;
  - c) p1.codigo=mio.tam;
  - d) mio.persona[1]=p1;
  - e) p1.telef[3]='5';
- Escribir sentencias que:
  - f) Escriba el teléfono de la primera persona del vector **datos** de la variable **tuyo**
  - g) Escriba el tamaño de la lista **mio**
  - h) Almacene el contenido de la variable **m** en el **codigo** de la tercera **persona** de la lista **tuyo**

2. Dadas las siguientes declaraciones:

```
struct complejo{
    float real;
    float imaginario;
};
struct complejo p1;
struct complejo puntos[50];
```

Indicar cuáles de las siguientes sentencias son correctas (si fuesen incorrectas indicar el por qué):

- a) puntos[4] = puntos[3];
- b) puntos.real[2] = 33.3;
- c) puntos[2].real = p1.real;
- d) p1 = puntos[0];
- e) printf("%f", puntos[1]);

3. Dada la siguiente **declaración de estructuras** de un programa en C, que representan una lista de fechas de este siglo:

```
#define N 20
struct fecha{
    int dia, mes, anio;
};
struct lista{
    int tam;    // Cantidad de valores del array siguiente
    struct fecha vfechas[N];    // Array de fechas
};
```

**3.1.** Escribir una función que **añada al final** de una lista de fechas una nueva fecha.

**3.2.** Escribir una función para **iniciar** mediante **lecturas** una lista del tipo anterior. El usuario introducirá por teclado máximo N (20) fechas, cada fecha en una línea con tres valores enteros que representan un día, mes y un año. En la última línea se almacenarán tres ceros. Utilizar las definiciones del ejercicio anterior.

**3.3.** Diseñar una función que a partir de una fecha y una lista, **busque** la fecha **en una lista** y nos devuelva el lugar en el que se encuentra. Si no existe devolverá el valor -1.

**3.4.** Escribir una función que a partir de una lista del tipo anterior, **escriba** las **fechas almacenadas en la lista**: una fecha en cada línea, con el formato dd/mm/aa. Si la lista está vacía se indicará mediante un mensaje.

- 4.** Dada la siguiente declaración de estructuras de un programa en C, que representa los datos de una operación bancaria (ingresar, retirar, transferir) realizada por un cliente:

```
struct fecha{
    int dia, mes, anio;
};
struct opBanco{
    char nombre[50];           // Nombre y apellidos
    char codigo[15];           // Código de cuenta
    char oper;                 // Operación: 'i' ingresar, 'r' retirar,
                               // 't' transferir
    float importe;             // Importe en euros
    struct fecha fechaOp;      // fecha de la operación bancaria
};
```

**4.1.** Diseñar una función para **iniciar mediante lecturas** de un **fichero de texto** una variable de tipo struct **opBanco**. Para ello se tienen 4 líneas en el fichero:

La primera línea contiene el nombre completo del cliente.

La segunda, el código de cuenta.

La tercera, el tipo de operación (carácter 'i', 'r' o 't') seguido de un blanco y del importe.

La cuarta, el día mes y año.

**4.2.** Diseñar una función para **escribir por pantalla** los elementos de una variable de tipo **opBanco**. Para ello se escribirán 4 líneas:

La primera línea contiene el nombre completo del cliente.

La segunda el código de cuenta.

La tercera el tipo de operación (ingresar, retirar o transferir) y el importe.

La cuarta el día mes y año.

**4.3.** Definir los tipos de datos necesarios para trabajar con una **lista** de operaciones bancarias (**struct opBanco**), con un máximo de **N** (valor constante) operaciones bancarias.

Diseñar una función que **añada al final** de una lista de operaciones bancarias una nueva operación.