

## Algunos ejercicios resueltos de estructuras (Tema 3)

A continuación se presenta el código fuente de algunos problemas del tema 3. Comparadlos con vuestras soluciones y plantead en clase cualquier duda o modificación.

### Ejercicio extra: *Asignación de escaños en unas elecciones*

Esta es una posible solución al ejercicio de la asignación de escaños. Se han considerado 50 provincias y 5 partidos, aunque estos parámetros son fácilmente modificables al estar establecidos como constantes. Además, el número de escaños asignados en cada provincia es variable, siendo elegido aleatoriamente entre 4 y una constante llamada MAX\_ESCAÑOS.

Nótese que la función main() es excesivamente larga y debería ser descompuesta en subfunciones. No se ha hecho para no dificultar la interpretación del algoritmo.

```
#include <stdio.h>
#include <stdlib.h>

#define PROVINCIAS 50
#define PARTIDOS 5
#define MAX_ESCANOS 20

int main()
{
    int votos[PROVINCIAS + 1][PARTIDOS + 1]; // Asignamos uno más: no usaremos la fila 0 ni la columna 0
    int esc[PROVINCIAS + 1][PARTIDOS + 1]; // Para los escaños tampoco usaremos la fila 0 ni la columna 0
    int aux[MAX_ESCANOS + 1][PARTIDOS + 1]; // Matriz auxiliar para el cálculo de escaños
    int prov, part, escano, e; // Contadores para los bucles
    int maximo, esc_max, part_max; // Variables auxiliares para calcular los máximos
    int num_escanos; // Número de escaños por provincia
    int totales[PARTIDOS + 1]; // Para calcular los votos totales nacionales de cada partido

    printf("Fundamentos programación 1º ASI\n");
    printf("Ejercicio del tema 3: REPARTO DE ESCAÑOS EN UNAS ELECCIONES\n");
    printf("-----\n\n");

    // Inicializamos la matriz de escaños a 0
    for (prov = 1; prov <= PROVINCIAS; prov++)
        for (part = 1; part <= PARTIDOS; part++)
            esc[prov][part] = 0;

    // Inicializamos el vector de votos totales a 0
    for (part = 1; part <= PARTIDOS; part++)
        totales[part] = 0;

    // Asignamos aleatoriamente los votos
    for (prov = 1; prov <= PROVINCIAS; prov++)
        for (part = 1; part <= PARTIDOS; part++)
            votos[prov][part] = rand() % 1000 + 1;
            esc[prov][part] = 0;

    printf("He asignado aleatoriamente los votos. Pulsa INTRO para ver el reparto...\n");
    getchar();

    // Mostramos por pantalla el resultado de la asignación de votos
    printf(" = VOTOS =          Partido 1   Partido 2   Partido 3   Partido 4   Partido 5\n");
    printf("-----\n");
    for (prov = 1; prov <= PROVINCIAS; prov++)
    {
        printf("Provincia %2i", prov);
        for (part = 1; part <= PARTIDOS; part++)
        {
            printf(" %4i", votos[prov][part]);
        }
        printf("\n");
    }

    // Calculamos los escaños en cada provincia
    for (prov = 1; prov <= PROVINCIAS; prov++)
```

```

{
    // Cada provincia tendrá un número de escaños variable entre 4 y MAX_ESCAÑOS
    num_escanos = rand() % (MAX_ESCANOS - 4) + 4;

    // Creamos la matriz auxiliar para la provincia "prov"
    for (part = 1; part <= PARTIDOS; part++)
    {
        for (escano = 1; escano <= num_escanos; escano++)
        {
            aux[escano][part] = votos[prov][part] / escano;
        }
    }

    // Mostramos la matriz auxiliar de cada provincia

    // Asignamos los escaños de esta provincia
    for (e = 1; e <= num_escanos; e++)
    {
        maximo = 0; // Vamos a buscar el mayor número en la matriz "aux"
        esc_max = 0;
        part_max = 0;
        for (part = 1; part <= PARTIDOS; part++)
        {
            for (escano = 1; escano <= num_escanos; escano++)
            {
                if (aux[escano][part] > maximo)
                {
                    // Hemos encontrado un nuevo máximo
                    maximo = aux[escano][part]; // Guardamos el nuevo máximo
                    esc_max = escano; // También guardamos su fila...
                    part_max = part; // ...y su columna
                }
            }
        }

        // El nuevo escaño corresponde al partido que haya tenido el máximo número de votos
        esc[prov][part_max]++; // Incrementamos en 1 el número de escaños de ese partido
        aux[esc_max][part_max] = 0; // Ponemos a 0 la entrada en la matriz auxiliar
        //para no volver a contar esos votos
    }
} // for prov

printf("\nHe calculado los escaños. Pulsa INTRO para ver el resultado...\n");
getchar();

// Mostramos por pantalla el resultado de la asignación de escaños
printf(" = ESCAÑOS =      Partido 1  Partido 2  Partido 3  Partido 4  Partido 5  Total\n");
printf("-----\n");
for (prov = 1; prov <= PROVINCIAS; prov++)
{
    printf("Provincia %2i      ", prov);
    num_escanos = 0;
    for (part = 1; part <= PARTIDOS; part++)
    {
        printf(" %4i      ", esc[prov][part]);
        // Vamos contabilizando los escaños de esta provincia
        num_escanos = num_escanos + esc[prov][part];
        // También contamos los escaños totales de cada partido
        totales[part] = totales[part] + esc[prov][part];
    }
    // Falta por escribir el total de escaños de esta provincia
    printf(" %4i\n", num_escanos);
}
// Por último, escribimos los totales nacionales
printf("TOTAL ESPAÑA      ");
num_escanos = 0;
for (part = 1; part <= PARTIDOS; part++)
{
    printf(" %4i      ", totales[part]);
    num_escanos = num_escanos + totales[part];
}
printf(" %4i\n", num_escanos);

printf("\n\n");

return 0;
}

```

### Ejercicio 3.24: *Liga de fútbol*

```
#include <stdio.h>

#define MAX_EQUIPOS 20          // N °máximo de equipos que puede manejar el programa

// Estructura de datos de cada equipo
struct s_equipo
{
    char nombre[50];
    int jug, gan, per, emp;
    int puntos;
};

// Prototipos de funciones
void introducir_datos(struct s_equipo equipos[MAX_EQUIPOS]);
void mostrar_datos(struct s_equipo equipos[MAX_EQUIPOS]);

int main(void)
{
    int opc;
    char txt[50];
    struct s_equipo equipos[MAX_EQUIPOS];    // Array donde almacenaremos los datos

    do
    {
        // Mostrar el menú de opciones
        printf("\n\nPROGRAMA DE LA LIGA DE FUTBOL\n\n");
        printf("MENU DE OPCIONES\n\n");
        printf("1 - Introducir datos\n");
        printf("2 - Mostrar datos\n");
        printf("3 - Salir del programa\n");

        // Leer la opción seleccionada por el usuario
        do
        {
            printf("\nElija una opción (1-3): ");
            gets(txt);
            opc = atoi(txt);
        }
        while ((opc < 1) || (opc > 3));

        // Llamar a la función correspondiente segn la opción elegida
        switch (opc)
        {
            case 1: introducir_datos(equipos); break;
            case 2: mostrar_datos(equipos); break;
        }
    }
    while (opc != 3);

    return 0;
}

// Leer por teclado los datos de los equipos y los inserta en el vector
void introducir_datos(struct s_equipo equipos[MAX_EQUIPOS])
{
    char aux[50];
    int i;

    for (i=0; i < MAX_EQUIPOS; i++)
    {
        // Leemos los datos de un equipo por teclado y los almacenamos en la posición "i" del array
        printf("Introduzca los datos del equipo nº %i\n", i);
        printf("  Nombre: ");
        gets(equipos[i].nombre);
        printf("  Jugados: ");
        gets(aux);
        equipos[i].jug = atoi(aux);
        printf("  Ganados: ");
        gets(aux);
        equipos[i].gan = atoi(aux);
        printf("  Perdidos: ");
```

```

        gets(aux);
        equipos[i].per = atoi(aux);
        equipos[i].emp = equipos[i].jug - equipos[i].gan - equipos[i].per;
        equipos[i].puntos = equipos[i].gan * 3 + equipos[i].emp * 1;
    }
}

// Recorre los datos del vector de equipos y muestra su contenido en la pantalla
void mostrar_datos(struct s_equipo equipos[MAX_EQUIPOS])
{
    int i;

    printf("\nEQUIPO          JUG GAN EMP PER PUNTOS\n\n");
    // Recorremos el array de datos y los mostramos en la pantalla
    for (i = 0; i < MAX_EQUIPOS; i++)
    {
        printf("%-20s %2i %2i %2i %2i %2i\n", equipos[i].nombre, equipos[i].jug, equipos[i].gan,
            equipos[i].emp, equipos[i].per, equipos[i].puntos);
    }
}

```

### Ejercicio extra: *Depósito de vehículos*

El encargado de un depósito de vehículos quiere un programa que le permita anotar la matrícula, la marca y el modelo de todos los coches que llegan al depósito, y luego quiere que le aparezca en la pantalla una lista con los coches ordenados por marca.

```

#include <stdio.h>
#include <string.h>

#define MAX_COCHES 3          // Número de coches del array

struct s_coche {              // Estructura para guardar los datos de cada coche
    char marca[20];
    char modelo[20];
    char matricula[10];
};

void introducir_datos(struct s_coche c[MAX_COCHES]);      // Prototipos
void ordenar_datos(struct s_coche c[MAX_COCHES]);
void mostrar_datos(struct s_coche c[MAX_COCHES]);

int main(void)
{
    struct s_coche coches[MAX_COCHES]; // Array de coches

    introducir_datos(coches);           // Lee los datos por teclado y los inserta en el array
    ordenar_datos(coches);              // Ordena los elementos del array según la marca del coche
    mostrar_datos(coches);              // Muestra en la pantalla el contenido del array
    return 0;
}

void introducir_datos(struct s_coche c[MAX_COCHES])
{
    int i;
    for (i=0; i<MAX_COCHES; i++)
    {
        printf("Introduzca los datos del coche nº %i\n", i+1);
        printf("Matrícula: "); gets(c[i].matricula);
        printf("Marca: "); gets(c[i].marca);
        printf("Modelo: "); gets(c[i].modelo);
    }
}

void ordenar_datos(struct s_coche c[MAX_COCHES]) // Usaremos el método de la burbuja
{
    int i, j;
    int comparacion;

```

```

struct s_coche aux;                                // Para hacer el intercambio de elementos

for (i=0; i<MAX_COCHES; i++)
    for (j=1; j<MAX_COCHES; j++)
    {
        // Comparamos la marca del coche j con el j-1
        comparacion = strcmp(c[j-1].marca, c[j].marca);
        if (comparacion > 0)                        // La marca del coche j es mayor que la del j-1
        {
            aux = c[j-1];                          // Intercambiamos los elementos j y j-1
            c[j-1] = c[j];
            c[j] = aux;
        }
    }
}

void mostrar_datos(struct s_coche c[MAX_COCHES])
{
    int i;
    printf("MARCA           MODELO           MATRICULA\n");
    printf("-----\n");
    for (i=0; i<MAX_COCHES; i++)
        printf("%-15s %-15s %-8s\n", c[i].marca, c[i].modelo, c[i].matricula);
}

```