



METODOLOGÍA DE LA PROGRAMACIÓN

Grado en Ingeniería Informática
Primer curso. Segundo cuatrimestre
Escuela Politécnica Superior de Córdoba
Universidad de Córdoba



Práctica 2: Memoria Dinámica.

NOMBRE Y APELLIDOS: **VÍCTOR MONSERRAT VILLATORO.**

GRUPO: **GM8.**

1.- ENUNCIADO.

Escribe un programa que permita gestionar los jugadores de baloncesto del equipo de una ciudad.

- Para ello se guardará la información de cada jugador en la siguiente estructura:

```
struct Ficha_jugador
{
    char nombre [50];
    int dorsal; /* N° entero */
    float peso; /* Expresado en kilos */
    int estatura; /* Expresado en centímetros*/
};
```

- El programa realizará secuencialmente las siguientes operaciones:

1. Crear un vector dinámico de jugadores.
2. Listar los jugadores registrados en el equipo, con las características de cada uno de ellos (dorsal, peso, estatura).
3. Borrar todos los jugadores con una 'a' en su nombre.
4. Listar de nuevo los jugadores.
5. Liberar memoria al terminar.

- Funciones que deberás implementar (también puede utilizar otras funciones auxiliares que considere oportunas):

- a) Función para reservar memoria para un vector de estructuras de jugador.
- b) Función para leer un nuevo jugador. La función pedirá al usuario los datos de un jugador y los devolverá en una estructura struct Ficha_jugador.
- c) Función para rellenar un vector de jugadores.
- d) Función para listar los jugadores del equipo.
- e) Función para borrar jugadores cuyo nombre contenga un carácter que se pasará como argumento.
 - Al terminar la ejecución, el vector de jugadores habrá reducido su tamaño usando la función realloc.
 - La función devolverá el n° de jugadores que ha borrado.

2.- DATOS DE ENTRADA.

Parámetros formales:

- Nombre: **nElementos**. Significado: **elementos que tendrá el vector (número de jugadores que hay)**. Tipo de dato: **entero (int)**. Restricciones: **ninguna**.
- Nombre: **jugadores**. Significado: **vector de estructuras Ficha_jugador que almacena los datos de todos los jugadores**. Tipo de dato: **estructura (struct **) Ficha_jugador****. Restricciones: **no ocupará más espacio que el asignado en memoria dinámica**.
- Nombre: **letra**. Significado: **será la letra que el programa comprobará si tienen o no los jugadores en su nombre para borrarlos**. Tipo de dato: **carácter (char)**. Restricciones: **ninguna**.

Parámetros locales:

- Nombre: **i**. Significado: **contador para recorrer el vector a través de un bucle**. Tipo de dato: **entero (int)**. Restricciones: **ninguna**.

3.- DATOS DE SALIDA.

- Nombre: **contador**. Significado: **contador de los jugadores que han sido borrados**. Tipo de dato: **entero (int)**. Restricciones: **ninguna**.

La función devuelve el parámetro contador a la función principal por valor, en la cual se mostrará por pantalla.

4.- DATOS AUXILIARES.

- Nombre: **ptr**. Significado: **apunta a la primera aparición de letra (o a NULL si no se encuentra) en cada llamada a la función strchr, que se ejecuta tantas veces como jugadores hay**. Tipo de dato: **puntero a carácter (char *)**. Restricciones: **ninguna**.

5.- DESCRIPCIÓN DEL ALGORITMO.

El algoritmo de la función consiste en recorrer el vector de jugadores e ir comprobando si en sus nombres está la letra que hará que sean borrados. Para ello, si se encuentra la letra en el nombre del jugador, se sobrescribirá este jugador con el siguiente y así sucesivamente hasta finalizar de recorrer el vector, el contador aumentará una vez por cada jugador borrado, y el número de elementos (nElementos) disminuirá una vez por cada jugador borrado también. Finalmente, el contador para el bucle (i), disminuirá para comprobar de nuevo al actual jugador que ocupa la posición que hemos borrado, y así hasta recorrer todos los jugadores.

Ejemplo:

Se desea borrar los jugadores cuyo nombre contenga el carácter 'a':

Pablo, Luis, Rodrigo, Juan, Miguel.

Vector después de realizar la eliminación de los jugadores con un 'a' en su nombre:

Luis, Rodrigo, Miguel.

6.- PSEUDOCÓDIGO DEL ALGORITMO.

```
[1]  Función borrar_jugadores (nElementos, Ficha_jugador **jugadores, letra)
[2]  Inicio
[3]  Para contador i
[4]    desde 0
[5]    hasta nElementos-1
[6]    [paso 1]
[7]    hacer
[8]      ptr ← Invocar strchr (jugadores [contador i].nombre, letra)
[9]      Si ptr ≠ NULO
[10]        entonces
[11]          Para contador i
[12]            desde i
[13]            hasta nElementos-1
[14]            [paso 1]
[15]            hacer
[16]              jugadores [i] ← jugadores [i+1]
[17]          Fin_para
[18]          contador ← contador +1
[19]          nElementos ← nElementos -1
[20]          contador i ← contador i -1
[21]        Fin_si
[22]  Fin_para
[23]  jugadores ← Invocar realloc (jugadores, nElementos · Invocar sizeof
    (Ficha_jugador))
[24]  Devolver contador
[25]  Fin_función
```

7.- CÓDIGO EN C.

```
int borrar_jugadores (int nEle, struct Ficha_jugador **jugadores, char letra)
{
    int i, j, contador=0;
    char *p;

    for (i=0; i<nEle; i++)
    {
        p=strchr (((*jugadores) [i]).nombre, letra);
        if (p!=NULL)
        {
            for (j=i; j<nEle-1; j++)
            {
                (*jugadores) [j] = (*jugadores) [j+1];
            }
            contador++;
            i--;
            nEle--;
        }
    }
    *jugadores = (struct Ficha_jugador*)realloc(*jugadores, (nEle)*sizeof(struct
```

```
Ficha_jugador));  
    return (contador);  
}
```