

**Relación de prácticas de la asignatura
METODOLOGÍA DE LA PROGRAMACIÓN
Segundo Cuatrimestre
Curso 2018-2019**

1º Grado en Informática

Práctica 2: Memoria dinámica, Bibliotecas

Objetivos

- Practicar conceptos básicos sobre memoria dinámica. Se hará uso de directivas de compilación para incluir de forma adecuada los ficheros de cabecera.
- Se hará uso de bibliotecas ya creadas y se crearán otras nuevas.

Recomendaciones

- No se podrá hacer uso de memoria estática.
- Dividir todos los ejercicios en varios ficheros y utilizar las directivas de inclusión condicional de código.

Temporización recomendada

2 sesiones de prácticas

¿Qué hay que entregar?

El ejercicio 1 y el análisis y diseño de la función *minCol* (ejercicio. 4)

El análisis consistirá en el estudio del problema que plantea el ejercicio.

- Qué datos de entrada necesita y de qué tipo son
- Cómo van a llegar esos datos
- Qué resultado se va a obtener y de qué tipo es
- Cómo se obtiene el resultado a partir de los datos de entrada
- Cómo se va a presentar al usuario el resultado final
- Ejemplo de que la solución propuesta funciona, utilizando los nombres dados a los datos

El diseño incluirá un algoritmo en pseudocódigo o diagrama de flujo que resuelva el problema y que servirá como base para la posterior codificación. Recordad que **el diseño es independiente del lenguaje de programación utilizado**.

¿Cuándo hay que entregar el ejercicio 1 ?

| Grupo | GM1-GM2 | GM3 -GM4 | GM5 |
|-------|----------|----------|----------|
| Fecha | 12/03/19 | 13/03/19 | 15/03/19 |

¿Cuándo hay que entregar el análisis y el diseño de la función minCol?

| Grupo | GM1-GM2 | GM3 -GM4 | GM5 |
|-------|----------|----------|----------|
| Fecha | 19/03/19 | 20/03/19 | 22/03/19 |

Memoria Dinámica

- Supón una matriz dinámica (*float ** tabla*) de 2x3 elementos, con los siguientes valores.
 $\{\{1.1, 1.2, 1.3\}, \{2.1, 2.2, 2.3\}\}$
 - ¿Cual es el significado de *tabla*?
 - ¿Cual es el significado de *(tabla+1)*?
 - ¿Cual es el significado de **(tabla+1)*?
 - ¿Cual es el significado de **(tabla+1)+1*?
 - ¿Cual es el significado de **(tabla)+1*?
 - ¿Cual es el valor de **(*(tabla+1)+1)*?
 - ¿Cual es el valor de **(*(tabla)+1)*?
 - ¿Cual es el valor de **(*(tabla+1))*?
- Escribe una función que, dado un vector dinámico, su longitud y un número entero (*num*), devuelva dos vectores dinámicos. Uno contendrá los elementos mayores que *num* y otro los elementos menores o iguales que *num*. Implementa un pequeño programa para probar la función.
- 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;          /* Nº entero */
    float peso;          /* Expresado en kilos */
    int estatura;        /* Expresado en centímetros*/
};
```

El programa realizará secuencialmente las siguientes operaciones:

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

Deberás implementar al menos las siguientes funciones (también puede utilizar otras funciones auxiliares que considere oportunas):

- Función para reservar memoria para un vector de estructuras de jugador.
- 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*.
- Función para rellenar un vector de jugadores usando la función para leer un jugador.
- Función para listar los jugadores del equipo.
- 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 número de jugadores que quedan en el vector.

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

| | | | |
|---------|---|-------|-----|
| Pablo | 4 | 80.5 | 192 |
| Luis | 5 | 90.2 | 201 |
| Antonio | 6 | 112.0 | 214 |
| Rodrigo | 7 | 85.7 | 194 |
| Juan | 8 | 93.0 | 198 |
| Miguel | 9 | 101 | 205 |

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

| | | | |
|---------|---|------|-----|
| Luis | 5 | 90.2 | 201 |
| Rodrigo | 7 | 85.7 | 194 |
| Miguel | 9 | 101 | 205 |

- Escribe un programa que implemente las siguientes funciones sobre matrices dinámicas y las llame de manera secuencial mostrando, de manera adecuada, la salida por pantalla.
 - int **reservarMemoria (int nFil, int nCol).* Reserva memoria para una matriz de *nFil* filas y *nCol* columnas.
 - void rellenaMatriz (int **matriz, int nFil, int nCol).* Rellena una matriz con valores aleatorios en el intervalo [1,20].
 - void imprimeMatriz (int **matriz, int nFil, int nCol).* Imprime una matriz por pantalla. Usa la notación de aritmética de punteros para recorrer la matriz.
 - int * minCol (int **matriz, int nFil, int nCol).* Devuelve un vector dinámico con los mínimos de cada columna.
 - void liberarMemoria(int ***matriz, int nFil).* Libera la memoria de una matriz reservada dinámicamente. La función pondrá el puntero *matriz* a NULL antes de terminar.
- Escribe un programa que lea una frase y, a partir de ella, cree un vector dinámico de cadenas con las diferentes palabras de la frase. A partir de este vector, el programa deberá:
 - Calcular la longitud media de las palabras de la frase, así como la longitud mayor y menor (usando paso de parámetros por referencia).
 - Construir un vector dinámico con la frecuencia de aparición de cada longitud.

Realiza una adecuada modularización, implementando todas las funciones que sean necesarias.

Bibliotecas y Doxygen

- Crea una biblioteca (*libMatrices.a*) a partir de las cuatro funciones del ejercicio 4 (*reservarMemoria*, *liberarMemoria*, *rellenaMatriz* e *imprimeMatriz*) y su correspondiente fichero de cabecera. Reproduce los resultados del ejercicio 4, pero esta vez haciendo uso de la biblioteca creada (por tanto sólo necesitarás un *main()*, la inclusión del *.h* de la biblioteca y enlazar con ella). Utiliza Doxygen para documentar todas las funciones de la biblioteca.
- Implementa una función que permita multiplicar matrices dinámicas. Utiliza las funciones incluidas en la biblioteca *libMatrices.a* para implementar el programa que te permita probar la función.