

# MEMORIA DINÁMICA

**Explicado en clase teoría :** Todo el temario.

**Objetivos:** Practicar con memoria dinámica.

**Tareas a realizar por el alumno:**

*(Los siguientes ejercicios están sacados del tema 8 del libro de apuntes)*

1. Realiza un programa que solicite al usuario un tamaño de vector y reserve memoria dinámica para un vector de enteros con el tamaño indicado. A continuación rellene el vector con valores solicitados al usuario. Realiza también una función que muestre el contenido de un vector y utilízala para comprobar que los valores solicitados al usuario se han leído correctamente.
2. La serie de Fibonacci está formada por la secuencia de números: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ... Por definición, los dos primeros números de la serie son 0 y 1 y los números siguientes se calculan como la suma de los dos números previos de la secuencia. Realiza un programa que solicite al usuario un número  $n$  y calcule y muestre en la pantalla los primeros  $n$  números de la serie de Fibonacci. Almacena los números de la serie en un vector dinámico.
3. Se almacena en una estructura la siguiente información sobre una persona: nombre, edad y localidad de nacimiento. Realiza una función que tome como parámetros de entrada un vector de personas, su longitud y una localidad y devuelva como salida un vector con los datos de las personas cuya localidad de nacimiento coincide con la localidad recibida como parámetro. El vector devuelto debe albergarse en memoria dinámica. Realiza también una función *main* que permita probar la función.
4. Realiza las siguientes funciones para el trabajo con vectores de *string*:
  - Función que recibe como entrada un tamaño y devuelve un vector ubicado en memoria dinámica de ese tamaño.
  - Función que recibe como entrada un vector de *strings* y su tamaño y muestre en la salida estándar los *strings*. Cada *string* debe aparecer en una línea distinta.
  - Función que recibe un vector de *strings* ubicado en memoria dinámica, su tamaño y un nuevo tamaño. La función debe trasladar el vector original a otro vector en memoria dinámica que tenga el nuevo tamaño. El nuevo tamaño puede ser menor o mayor que el tamaño previo. Se debe liberar la memoria ocupada por el vector original y se debe devolver la dirección de inicio en memoria dinámica del nuevo vector.
  - Función que reciba un vector de *strings* y devuelva una copia del vector.
  - Función *main* que permita probar las funciones anteriores.