# ARRAYS UNIDEMSIONALES

1. Leer 5 números y mostrarlos en orden inverso al introducido
2. Leer 5 números por teclado y a continuación realizar la media de los números positivos, la media de los negativos y contar el número de ceros.
3. Leer 10 números enteros. Debemos mostrarlos en el siguiente orden: el primero, el último, el segundo, el penúltimo, el tercero, etc.
4. Leer por teclado dos arrays de 10 números enteros y mezclarlos en una tercera de la forma: el 1º de A, el 1º de B, el 2º de A, el 2º de B, etc
5. Leer por teclado una serie de 10 números enteros. La aplicación debe indicarnos si los números están ordenados de forma creciente, decreciente, o si están desordenados
6. Diseñar una aplicación que declare un array de 10 elementos enteros. Leer mediante el teclado 8 números. Después se debe pedir un número y una posición, insertarlo en la posición indicada, desplazando los que estén detrás
7. Leer 10 enteros. Guardar en otro array los elementos pares del primero, y a continuación los elementos impares
8. Leer dos series de 10 enteros, que estarán ordenados crecientemente. Copiar (fusionar) los dos arrays en una tercero, de forma que sigan ordenados.
9. Leer 10 enteros ordenados crecientemente. Leer N y buscarlo en la tabla. Se debe mostrar la posición en que se encuentra. Si no está, indicarlo con un mensaje

# ARRAYS N-DIMENSIONALES

1. Crear un array **bi**dimensional de tamaño 5x5 y rellenarla de la siguiente forma: la posición T[n,m] debe contener n+m. Después se debe mostrar su contenido.
2. Crear y cargar dos matrices de tamaño 3x3, sumarlas y mostrar su suma (guardada en otra matriz).
3. Crear un array bidimensional de tamaño 7x7 y rellenarlo de forma que los elementos de la diagonal principal sean 1 y el resto 0.
4. Se pretende realizar un programa para gestionar la lista de participaciones en una competición de salto de longitud. El número de plazas disponible es de 10. Sus datos se irán introduciendo en el mismo orden que vayan inscribiéndose los atletas. Se utilizará un array unidimensional para los nombres de 10 elementos, y una array bidimensional que contenga 10 filas (atletas) con 3 columnas (marcas) con valores reales (float).Diseñar el programa que muestre las siguientes opciones:

1- Inscribir un participante. 2- Mostrar listado de datos.

3- Mostrar listado por marcas. 4- Finalizar el programa.

Si se selecciona 1, se introducirán los datos de uno de los participantes: Nombre, mejor marca del 2017, mejor marca del 2018 y mejor marca del 2019.

Si se elige la opción 2, se debe mostrar un listado por número de dorsal.

La opción 3 mostrará un listado ordenado por la marca del 2017, de mayor a menor.

Tras procesar cada opción, se debe mostrar de nuevo el menú inicial, hasta que se seleccione la opción 4, que terminará el programa

# MÉTODOS ESTÁTICOS

Crea una clase llamada: Encargo

1. Diseñar un método estático maximo que tenga como parámetros dos números enteros, y que devuelva el máximo de los dos.
2. Diseñar un método estático maximo que devuelva el máximo de un array de elementos enteros. El método estará sobrecargado del 1.
3. Diseñar un método estático: opera\_array, a la que se le pasa dos arrays, el número de elementos útiles y que operación se desea realizar: sumar, restar, multiplicar o dividir (mediante un carácter: 's', 'r', 'm', 'd'). El método estático debe devolver un array con los resultados.
4. Diseña un método estático: primitiva, que toma como parámetros dos arrays. La primera con los 6 números enteros comprendidos entre (1 y

49) de una apuesta de la primitiva, y la segunda con los 6 números ganadores. El método estático debe devolver el número de aciertos.

# MÉTODOS ESTÁTICOS RECURSIVOS

1. Diseña un método estático recursivo que calcule an.

2. Diseña un método estático que devuelva el factorial de n (entero) recursivamente.

3. Diseña un método estático que calcule el n-ésimo término de la serie de Fibonacci. En esta serie el n-ésimo valor se calcula sumando los dos valores anteriores. Es decir fibonacci(n) = fibonacci(n-1)+fibonacci(n-2), siendo fibonacci(0)=1 y fibonacci(1)=1.

4. Diseña un método estático que calcule el valor máximo de un array de forma recursiva.

5. Diseña un método recursivo que calcule la suma de los n primeros números naturales.

6. Diseña un método recursivo que calcule número de cifras tiene un número entero.

7. Diseña un método recursivo que calcule el cociente entre dos números enteros.