1. Codifica un programa Java que lea un array de 15 elementos y visualice la suma de los elementos que ocupan la posición par. (0, 2, 4 …)

2. Codifica un programa Java que lea un array de 10 elementos y busque el menor, indicando la posición donde se encuentra.

3. Codifica un programa Java que lea las temperaturas del mes de Enero y calcule y visualice:

- Temperatura media
- La temperatura más baja
- Cuantos días han tenido temperaturas superiores a la media

4. Codifica un programa Java que lea un array de 10 elementos e intercambie el elemento mayor con el último.

5. Codifica un programa Java que lea un array con las notas de Programación de 20 alumnos, y que escriba :

- Nota media de la asignatura.
- Porcentaje de aprobados
- Cuantas notas inferiores a 2.
- Si ha habido algún 10.

6. Desarrolla un algoritmo que cargue un vector de 10 elementos y cambie el mayor elemento a la posición del menor y el menor a la posición del mayor.

7. Carga un vector de 15 elementos T(I) del 1 al 15 y a partir de él crea otro N(J) de 15 elementos de la siguiente forma:

N(1) = T(1)
N(2) = T(1) + T(2)
:
N(15) = T(1) + T(2) + T(3)+ ….T(15)

8. Introducir una secuencia de 10 números, cuyos valores están comprendidos entre 1 y 10. Queremos saber al final que números del 1 al 10 no han sido introducidos.

9. Escribir un método que reciba un array de 10 elementos y posteriormente un número entero y devuelva cuántas veces aparece ese número en el array.

10. Crea un array con los nombres de 10 alumnos, para finalizar antes de introducir los 20 nombres, pulsamos “F” o “f”.

11. Escribir un método que saque por pantalla el mayor y el menor elemento de un array de 10 elementos.

12. Codifica un programa Java que lea un array de 10 elementos v[].

- Crea un array nuevo con los 5 primeros elementos del array v[].
- Visualiza el array v ordenado, con ayuda de la clase Arrays método sort.
- Busca un valor en el array ordenado, utilizando el algoritmo de la **búsqueda binaria,** crear el método **buscar(v, valor)** al cual le pasaremos el array ordenado y el valor a buscar.

13. Codifica un programa Java que lea una matriz de 5 x 5 (creada con valores aleatorios entre 1 y 10) método **rellenarmatriz(m)** y :

a. Visualice los elementos de la diagonal principal. Método **diagonalprincipal(m)**
b. Visualice la suma de sus filas. Método **suma(m)**
c. Ponga a 0 los elementos de la columna 3. Método **cambioacero(m)**

14. Partiendo de una matriz de 5x5 valores aleatorios entre 1 y 100 : **Método rellenarmatriz(matriz);**

a. Visualiza los elementos de la diagonal secundaria. **Método diagonalSecundaria(matriz);**

b. Busca el mayor elemento e indica en qué fila se encuentra. **Método elementoMayor(matriz)**

c. Intercambia los elementos de la fila 3 por los de la fila 4. **Método IntercambiarFila3por4(matriz)**;

15. Codifica un programa que en un Array de 4x15 lea el número de la comarca (del 1 al 15), el número de habitantes (aleatorio entre 200 y 5000), la extensión en kilometros (aleatorio entre 20 y 100) y la densidad de población (hab/km) de 15 comarcas de la Comunidad Valenciana (Método **rellenarComarcas(comarcas**)) y posteriormente calcule:

a) Número de la comarca de menor extensión. Método **menorExtension(comarcas)**

b) ¿Hay alguna comarca con menos de 10.000 habitantes?. Método **menosDeDiezMilH(comarcas)**

c) Cuantas comarcas tienen una densidad de población superior a la media. Método **densidadSuperiorMedia(comarcas)**;

16. Codifica un programa que pida los datos de una matriz de dimensiones 2 X 2 método **rellenar(matriz)**, y compruebe si la matriz forma un cuadrado mágico (es decir si todas las filas suman el mismo número y todas las columnas también, y además ese número coincide con el de las filas). Método **comprobarSiEsMagico**.

17. Desarrolla un algoritmo que cargue en memoria los nombres y números de teléfono de 15 personas (método **rellenar**) y nos permita consultar secuencialmente su número de teléfono introduciendo el nombre de una persona determinada (método **busquedaSecuencial**). Para finalizar la consulta introduciremos una ‘N’ o ‘n’