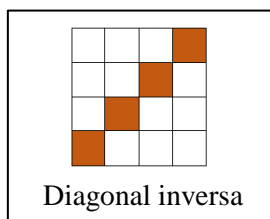
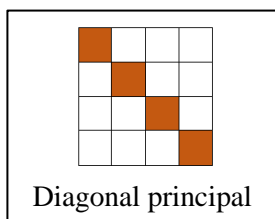


## Ejercicios arrays

### Ejercicio nº 1.-

Se desea codificar un programa en Java, que nos permita declarar un array de 5 filas y 5 columnas, para almacenar números enteros, después a través de un menú de opciones, realizar las siguientes tareas:

1. El usuario debe rellenar la matriz completamente.
2. Obtener la suma de los valores de una fila que previamente ha indicado el usuario (Se debe comprobar que la fila que solicita existe en la matriz).
3. Obtener la suma de los valores de una columna que previamente ha indicado el usuario. (Se debe comprobar que la columna que solicita el usuario existe en la matriz).
4. Obtener la suma de los valores de la diagonal principal del array.
5. Obtener la suma de los valores de la diagonal inversa del array.
6. Obtener la media de todos los valores del array



**NOTA IMPORTANTE.** Hasta que no se haya realizado la primera opción, es decir, hasta que no se hayan introducido valores en el array, el resto de opciones no se podrán ejecutar, simplemente se mostrará un mensaje donde se diga que debes rellenar el array.

### Ejercicio nº 2.-

Se desea codificar un programa en Java que nos permita simular el típico juego del 3 en raya, donde habrá dos jugadores que tendrán que completar el 3 en raya, los signos para representar la fichas serán: «X» y «O», cuando haya una posición vacía en el tablero habrá un «-»

El tablero de juego será un array de 3x3 de char.

El juego terminará cuando uno de los jugadores haga 3 en raya o si no hay posiciones libres para colocar fichas, en ese momento el programa indicará quién ha sido el jugador ganador, o que no se puede continuar introduciendo fichas.

El juego debe pedir las posiciones donde el jugador actual quiere poner su ficha, esta debe ser una posición válida y debe encontrarse libre.

Intentar utilizar la forma modular en lo que sea posible.

X		O	Ha ganado el jugador que juega con las fichas «X»
	X	O	
		X	

### **Ejercicio nº 3.-**

Se desea codificar un programa en Java que nos permita tabular los resultados de una encuesta que vamos a realizar a 10 personas. En esta encuesta indicaremos el sexo (1=masculino, 2=femenino), si trabaja (1=si trabaja, 2=no trabaja) y su sueldo (si tiene un trabajo, sino será un cero) estará entre 600 y 2000 € (valor entero). Una vez realizada la encuesta se quiere obtener la siguiente información:

1. Porcentaje de hombres (tengan o no trabajo)
2. Porcentaje de mujeres (tengan o no trabajo)
3. Porcentaje de hombres que trabajan
4. Porcentaje de mujeres que trabajan
5. Sueldo medio de los hombres que trabajan
6. Sueldo medio de las mujeres que trabajan

Puedes utilizar los métodos que veas necesario.

### **Ejercicio nº 4.-**

Se desea codificar un programa en Java que nos permita llevar la agenda de eventos que tiene programados una determinada persona para un año concreto. La agenda nos permitirá introducir un evento programado para una determinada fecha de ese año, por ejemplo: el usuario introducirá el mes y el día, (noviembre, 24), la agenda deberá comprobar que para esa fecha el usuario no tiene programado aun evento alguno, en ese momento le pedirá que introduzca el evento a añadir (por ejemplo, “concierto”), si el usuario ya tuviera programado algún evento para ese día la agenda le informará de dicha situación, dándole la opción de modificar el evento o no.

Así mismo, el programa nos permitirá consultar los eventos que tenemos programados para un día concreto, en este caso se introducirá el mes y el día a consultar y el programa nos mostrará el evento programado, o nos dirá que para esa fecha “no hay eventos programados”

### **Ejercicio nº 5.-**

Deseamos codificar un programa en Java que nos permita identificar si una matriz de  $N \times N$  elementos es un cuadrado mágico o no. Una matriz es un cuadrado mágico si las sumas de cada una de las columnas, las sumas de cada una de las filas y las sumas de la diagonal principal y la suma de la diagonal inversa dan el mismo valor.

8	1	6
3	5	7
4	9	2

La tabla que se muestra es un cuadrado mágico, si hacemos las sumas indicadas obtenemos como resultado 15 en todos los casos.

### **Ejercicio nº 6.-**

Deseamos codificar un programa en Java que nos permita gestionar la puntuación de una liga de fútbol, en la que compiten 1º equipos a una sola vuelta. Los resultados se almacenan en una tabla de dos dimensiones, de forma que cada elemento (i, j) de la tabla guarda los tantos que ha marcado el equipo «i» al equipo «j» en el encuentro. Por ejemplo, un «8» en la posición (2,6) significa que en el encuentro entre el equipo «2» y el equipo «6» el equipo «2» ha metido 8 goles al equipo «6».

Los nombres de los equipos y la puntuación obtenida por cada uno de los equipos se almacenarán en arrays, los nombres en un array de 10 posiciones (una por equipo) y las puntuaciones en otro array, con los puntos totales obtenidos por cada equipo.

El programa nos deberá permitir, a través de un menú de opciones, realizar las siguientes tareas:

1. Inicializar la competición, para ello se introducirán los nombres de los equipos participantes, inicializando sus puntuaciones a 0.
2. Introducir los resultados de un partido de fútbolín, para ello, se introducirán los equipos que han participado y los goles que ha marcado cada uno de los equipos.
3. Obtener la clasificación en un momento dado de la competición, para ello debes tener en cuenta que se reparten: (3 puntos por partido ganado, 1 punto por partido empatado y 0 puntos por partido perdido). En la clasificación se mostrará:

#### P A R T I D O S

	-----				
Equipo	Jugados	Ganados	Empatados	Perdidos	Puntos totales

#### Ejercicio nº 7.-

La V Edición de la Vuelta Ciclista a los Campos Sorianos consta de 5 etapas en las que participan 15 ciclistas. Los resultados de cada etapa se almacenan en una estructura de Array, que almacena los tiempos (hh:mm:ss) que ha empleado cada ciclista en la etapa correspondiente

##### Tiempos de la etapa 1

	hh	mm	ss
1			
2			
3			
...			
15			

El programa deberá, a través de un menú de opciones, realizar las tareas siguientes:

1. Inicializar la edición, para ello se introducirán en un array los nombres de los 15 ciclistas participantes en la competición.
2. Introducir los tiempos empleados en cada etapa por los corredores que la han finalizado, si un ciclista no finaliza la etapa se le introducirá como tiempo invertido 0, y no podrá tomar la salida en las siguientes etapas.
3. Obtener la clasificación final de la Edición, proclamando el palmarés de la misma, mostrando la información:

			Tiempo empleado		
			-----		
Puesto	Dorsal	Nombre Ciclista	Horas	Minutos	Segundos