Ejercicios Tema 02. Arrays y Matrices

Ejercicio 1. Implemente los siguientes métodos que reciben un array de enteros como parámetro y (1) lo rellenan con el valor de la posición (índice en el array), (2) con un valor pasado como parámetro en todas sus posiciones, o (3) con el valor de potencias de dos (2^{indice}).

```
public static void completarArrayPosicion(int [] array)
public static void completarArrayValor(int [] array, int valor)
public static void completarArrayPotenciasDos(int [] array)
```

Ejercicio 2. Implemente un método que reciba como parámetro un array de reales (`double`) y compruebe si un `valor` real (`double`), también dado como parámetro, está contenido en el array. El método devolverá un entero que indique la posición en la que está el valor a encontrar o -1 en caso de no estar contenido en el array.

```
public static int buscarEnArray(double [] array, double valor)
```

Ejercicio 3. Implemente un método que reciba un array de booleanos (`boolean`) y que devuelva el número de valores que están a `true`. Para procesar el array de booleanos se debe hacer, recorriendo el array desde el final hasta el principio.
Implemente otro método que, dado un valor entero como parámetro, devuelva un array de booleanos (`boolean`) con tamaño ese valor dado como parámetro (`tamano`) y rellenado con las posiciones pares a `true` y las impares a `false`.

```
public static int numeroDeVerdaderos(boolean [] array)
public static boolean [] arrayBooleanos(int tamano)
```

Ejercicio 4. Implemente dos métodos que reciben una matriz de enteros (`int`) como parámetro y la rellenan por filas o por columnas con números consecutivos empezando desde el 0.

```
public static void completarMatrizPorFilas(int [][] matriz)
public static void completarMatrizPorColumnas(int [][] matriz)
```

Ejercicio 5. Implemente un método que reciba como parámetro una matriz de reales (`double`) y devuelva la suma de todos sus elementos

```
public static double obtenerSumaElementosMatriz(double [][] matriz)
```

Ejercicio 6. Implemente un método que reciba como parámetro una matriz de reales (`double`) y un factor de multiplicación (`factor`), y devuelva una nueva matriz cuyos elementos serán los contenidos en la matriz recibida como parámetro multiplicados por el factor dado.

```
public static double [][] matrizMultiplicada(double [][] matriz, double factor)
```

Ejercicio 7. Implemente un método que cree una matriz irregular (tal y como se muestra en la siguiente figura) y la muestre en la consola.

```
public static void mostrarMatrizIrregular()
```

1	2	3	4	5
1	2	3	4	
1	2	3		
1	2			
1				

Ejercicio 8. Implemente un método que reciba como parámetro un valor entero que sea el número de las dos dimensiones (`dimension`) de una matriz cuadrada. El método creará y devolverá la correspondiente matriz cuadrada, rellenndo sus posiciones para que construya la matriz identidad (elemento neutro del producto de matrices) tal y como se indica en la siguiente figura.

```
public static int [][] generarMatrizIdentidad(int dimension)
```

1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1

Ejercicio 9. Implemente un método que reciba por parámetro una matriz cuadrada de enteros (`int`) y devuelva una matriz (también de enteros (`int`) y con solo dos filas) con los valores de las diagonales principales de dicha matriz.

```
public static int [][] obtenerDiagonales(int [][] matriz)
```

Ejercicio 10. Implemente de otra forma el *ejemplo 6* de matrices visto en clase de teoría en el que se muestra qué fila de la matriz de enteros (`int`) pasada como parámetro tiene el mayor valor de suma de todos sus elementos. Utilice `Integer.MIN_VALUE` que es una constante con el menor valor que puede tomar un entero (`int`). Implementar también otro método en Java, siguiendo este último esquema (`Integer.MIN_VALUE`), que muestre qué columna de la matriz tiene el mayor valor de suma.

```
public static void filaConMayorSuma(int [][] matriz)
public static void filaConMayorSumaOtroMetodo(int [][] matriz)
public static void columnaConMayorSuma(int [][] matriz)
```