

Práctico 3 – Matrices

OBJETIVOS:

- Definir, declarar e inicializar arreglos bidimensionales.
- Desarrollar e implementar algoritmos que recorran y manipulen matrices.
- Fortalecer el uso de estructuras de control anidadas (bucles dobles).
- Aplicar buenas prácticas de codificación en el trabajo con datos estructurados.

Ejercicio 1

Implementar una función que muestre por consola el contenido de una matriz.

Firma a utilizar: `public void mostrarMatriz(int[][] mat);`

Ejercicio 2

Implementar una función que muestre la diagonal principal de una matriz

Firma a utilizar: `public void mostrardiagonal(int[][] mat);`

Ejercicio 3

Implementar una función que retorne el máximo valor de una matriz.

Firma a utilizar: `public int maximoMatriz(int[][] mat);`

Ejercicio 4

Implementar una función que muestre una columna de una matriz

Firma a utilizar: `public void mostrarColumna(int[][] mat, int columna);`

Ejercicio 5

Implementar una función que muestre una fila de una matriz

Firma a utilizar: `public void mostrarFila(int[][] mat, int fila);`

Ejercicio 6

Implementar una función que muestre las filas impares de una matriz

Firma a utilizar: `public void mostrarFilasImpares(int[][] mat);`

Ejercicio 7

Implementar una función que busque un elemento en una matriz y retorne true si el elemento existe en la matriz y false en otro caso.

Firma a utilizar: `public boolean buscarElementoEnMatriz(int[][] mat, int elemento);`

Ejercicio 8

Implementar una función que busque un elemento en una columna de una matriz y retorne true si el elemento existe en la matriz y false en otro caso.

Firma a utilizar: `public boolean buscarEnColumna(int[][] mat, int columna, int elemento);`

Ejercicio 9

Implementar una función que muestre por consola, la suma de cada columna de la matriz.

Firma a utilizar: `public void mostrarSumaCol(int[][] mat);`

Ejercicio 10

Implementar una función que indique cual es la fila que suma más. En caso de haber más de una, retornar la de índice menor

Firma a utilizar: `public int filaMayorSuma(int[][] mat);`

Ejercicio 11

Implementar una función que reciba una matriz y un número, e indique si existe una fila en donde se encuentre dicho número exactamente 3 veces.

Firma a utilizar: `public boolean existeFilaRepetida3 (int[][] mat, int valor);`

Ejercicio 12

Implementar una función que indique si existe una columna en donde existan al menos 3 números consecutivos que sean estrictamente ascendentes. Retornar número de columna (- 1 en caso de no existir ninguna).

Firma a utilizar: `public int existeColumnaAscendente (int[][] mat);`

Ejercicio 13

Implementar una función que determine si una matriz es simétrica horizontalmente. Es decir, si la primera fila es igual a la última, la segunda a la penúltima, y así sucesivamente.

Firma a utilizar: `public boolean esSimetricaHorizontal (int[][] mat);`

Ejercicio 14

Implementar una función que determine si existe alguna fila que sea palíndroma, es decir, que se lea igual de izquierda a derecha que de derecha a izquierda. Si existe, retornar el índice de la primera fila palíndroma encontrada; si no hay ninguna, retornar -1.

Firma a utilizar: `public int esPalindroma (int[][] mat);`