

Ejercicios de matrices

1. Implementa un programa que permita realizar operaciones sobre matrices de $N \times N$. El programa debe permitir al usuario la selección de las siguientes operaciones:
 - a) Sumar 2 matrices.
 - b) Multiplicar 2 matrices.
 - c) Trasponer una matriz.
 - d) Mostrar una matriz señalando cuáles son los puntos de silla (los *puntos de silla* de una matriz son aquellos elementos de la misma que cumplen ser el mínimo de su fila y el máximo de su columna).

Habrás también un subprograma para cargar de archivo y una matriz, y otro para mostrar en la consola una matriz.

2. Una pantalla de texto del PC tiene 25 filas, de 80 columnas cada una de ellas. En cada posición de pantalla se guarda el carácter representado en dicha posición, el color con que se escribe el carácter, el color de fondo donde se visualiza el carácter y si el carácter parpadea o no. Diseña las estructuras de datos necesarias considerando que los colores posibles son: negro, azul, verde, amarillo y rojo.

Construye un subprograma que, dada una fila, devuelva el color de fondo más utilizado en esa fila. Y otro que determine en qué fila y columna se encuentra el primer carácter parpadeante.



3. Escribe un programa que tras leer de archivo una matriz de orden $N \times M$ de valores enteros, indique si el máximo y el mínimo de los valores de la matriz coinciden en la misma fila o en la misma columna. Si hubiera varios máximos o mínimos se tomarían las posiciones de la primera aparición.

El archivo de entrada está formado por N líneas; en cada línea hay M enteros.

4. Queremos programar el juego de las 3 en raya:

X	O	
	X	X
	O	O

- Define una estructura de datos para representar el tablero de juego.
 - Implementa un subprograma para inicializar el tablero con blancos.
 - Implementa un subprograma para dibujar el tablero junto con el contenido de cada posición ('X', 'O', ' ').
 - Implementa un subprograma que devuelva true o false, dependiendo de si todo el tablero está ocupado o existen casillas libres.
 - Implementa la función Booleana `TresEnRaya(tableroJuego, jugador)` que devuelva true en caso de que un jugador tenga la jugada de tres en raya y false en caso contrario.
5. El pirata Pata de Palo desea conseguir el Gran Tesoro sin morir ahogado. La *Isla*, que consiste en un tablero cuadrado de NxN, rodeado de agua y con dos puentes en dos esquinas, se puede representar así:

A	G	U	A	
G				A
U				U
A				G
	A	G	U	A

- Escribe un procedimiento `inicializarTablero()` que coloque sobre el tablero al pirata Pata de Palo y el Gran Tesoro en casillas distintas, determinadas de una forma aleatoria. El agua se ha de colocar en los bordes y los puentes en las esquinas indicadas.
- Escribe un subprograma `direccionPirata()` que nos dé la dirección en la que avanzará el pirata (Norte, Sur, Este u Oeste). Un número aleatorio entre 1 y 4 decidirá qué dirección toma el pirata.
- Escribe un procedimiento `dibujarTablero()` que dibuje el tablero con el formato de la figura.
- Escribe un programa que permita al pirata **Pata de Palo** intentar buscar el **Gran Tesoro** en menos de 50 movimientos. El pirata recorre los cuadros de uno en uno según la dirección. Al final ha de indicarse si el pirata ha encontrado el tesoro o si se ha ahogado (ha caído al agua).

6. Se quiere desarrollar un subprograma que, dadas dos matrices cuadradas de enteros, M de 10 x 10 y P de 3 x 3, compruebe si entre todas las submatrices 3x3 que se pueden formar en la matriz M, desplazándose por filas o columnas, existe al menos una que coincida con la matriz P. En ese caso, se indicarán la fila y la columna de la matriz M en la que empieza la submatriz que coincide. Si hay varias, bastará con indicar la primera.
7. El juego de las 4 en línea consta de un tablero formado por siete columnas y seis filas. En una partida participan dos jugadores, uno con fichas blancas y otro rojas. Inicialmente todas las posiciones del tablero están libres. Cada jugador coloca alternativamente una ficha en una columna. La ficha colocada cae por su propio peso hasta el fondo de la columna correspondiente (primera fila de la columna libre); por ejemplo, en la figura si el jugador Rojo coloca una ficha en la columna 2, la ficha se coloca en la fila 3. La partida la gana el jugador que coloque en primer lugar cuatro de sus fichas en línea horizontal, vertical o en diagonal. La partida queda en tablas si ninguno de los jugadores es capaz de alinear cuatro fichas después de llenar el tablero.

6	L	L	L	L	L	L	L
5	L	L	L	L	L	L	L
4	L	L	L	■	L	L	L
3	L	L	■	□	■	L	L
2	L	■	□	□	□	■	L
1	L	■	■	□	□	□	■
	1	2	3	4	5	6	7

- a) Codifica las estructuras de datos necesarias para jugar una partida.
- b) Escribe un subprograma `inicializarJuego()` que quite todas las fichas del tablero y prepare el tablero de juego.
- c) Escribe un subprograma `colocarFicha()` que dado un jugador (blanco o rojo) y una columna (1 a 7), coloque la ficha en la posición correspondiente.
- d) Escribe un subprograma `presentarTablero()` que visualice en la pantalla el estado del tablero (como en la figura, excepto la rejilla).
- e) Escribe un subprograma `comprobarGanador()` que, dado un jugador (blanco o rojo) y una determinada casilla (1 a 6, 1 a 7), determine si hay cuatro fichas del mismo jugador alineadas en horizontal, vertical o diagonal.