

Campus Pérez Zeledón
Programación I
Tema: Matrices

1. Ejercicios

1.1. Matriz Frame

Diseñar el algoritmo correspondiente a un programa, que: Crea una tabla bidimensional de longitud 5x5 y nombre 'frame'. Carga la tabla con dos únicos valores 0 y 1, donde el valor uno ocupará las posiciones o elementos que delimitan la tabla, es decir, las más externas, mientras que el resto de los elementos contendrán el valor 0.

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Visualiza el contenido de la matriz en pantalla.

1.2. Ordenamiento Burbuja

Usando el método burbuja, realice un ordenamiento por medio de dos criterios. La llave primaria en la columna i, llave secundaria en la columna j.

1.3. Columna Identica

Realice un método que dada una matriz $M[n][m]$ con entradas de tipo entero determine si la matriz presenta dos columnas idénticas.

1.4. Ordenamiento de columnas por suma

Dada una matriz, calcule la suma $S_j = \sum_{i=0}^{n-1} M_{i,j}$ de los valores de la columna j-ésima. Posterior, ordene las columnas de dicha matriz de menor a mayor suma por columna.

1.5. Sudoku válido

Determinar si un 9 x 9 tablero de Sudoku es válido. Sólo es necesario validar las celdas completadas de acuerdo con las siguientes reglas:

1. Cada fila debe contener los dígitos 1-9 sin repetición.
2. Cada columna debe contener los dígitos 1-9 sin repetición.

3. Cada una de las nueve 3 x 3 subcasillas de la cuadrícula debe contener los dígitos 1-9 sin repetición.

Nota:

- Un tablero de Sudoku (parcialmente lleno) podría ser válido pero no necesariamente solucionable.
- Sólo es necesario validar las celdas completadas de acuerdo con las reglas mencionadas.

Ejemplo 1:

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

1.6. Girar imagen

Se le proporciona un $n \times n$ matriz 2D que representa una imagen, gire la imagen 90 grados (en el sentido de las agujas del reloj).

Tienes que rotar la imagen in situ , lo que significa que tienes que modificar la matriz 2D de entrada directamente. NO asigne otra matriz 2D y realice la rotación.

■ **Ejemplo 1**

1	2	3
4	5	6
7	8	9

→

7	4	1
8	5	2
9	6	3

▪ Ejemplo 2

5	1	9	11
2	4	8	10
13	3	6	7
15	14	12	16

⇒

15	13	2	5
14	3	4	1
12	6	8	9
16	7	10	11

1.7. Búsqueda de palabras

Dada una $m \times n$ cuadrícula de caracteres **board** y una cadena **word**, devuelve true si **word** existe en la cuadrícula.

Una palabra se puede construir a partir de letras de celdas secuencialmente adyacentes, donde las celdas adyacentes son vecinas horizontal o verticalmente. La misma celda con letra no se puede utilizar más de una vez.

▪ Ejemplo 1:

A	B	C	E
S	F	C	S
A	D	E	E

- **Entrada:** palabra = ABCCED
- **Salida:** verdadero

▪ Ejemplo 2:

A	B	C	E
S	F	C	S
A	D	E	E

- **Entrada:** palabra = ABCB
- **Salida:** falso

1.8. El triángulo de Pascal

Dado un número entero **numRows**, devuelve el primer número de filas del triángulo de Pascal.

En el triángulo de Pascal , cada número es la suma de los dos números directamente encima de él, como se muestra:



- **Entrada:** numRows = 5
- **Salida:** `[[1],[1,1],[1,2,1],[1,3,3,1],[1,4,6,4,1]]`

1.9. Transformar vector en matriz

De acuerdo al ejercicio anterior del **triángulo de Pascal**, el cual da como resultado un vector, realice una función que pueda pasar los valores numericos a una representación de matriz, donde los valores vacios para formar el triángulo, sean representados por un 0.