

Informática II - Prepa Tec Campus Eugenio Garza Lagüera
Laboratorio Primer Parcial

Sección 1: Ordena los siguientes arreglos a mano ascendentemente, mostrando cada uno de los pasos, utilizando los algoritmos de Selection Sort y Bubble Sort. Indica la cantidad de comparaciones e intercambios que tuvo que realizarse en cada algoritmo.

1.

45	1	88	0	35	99
----	---	----	---	----	----

2.

5		4	3	2
---	--	---	---	---

Sección 2: Codifica los siguientes ejercicios.

Problema 1: Diseña una clase llamada **TicTacToe** que sirva para modelar un juego de Tic-Tac-Toe (gato) para dos jugadores. El juego deberá validar las entradas y mostrar en consola el ganador del juego.

Diseña métodos para cada una de las acciones del juego. Por ejemplo:

- Determinar si existe un ganador o no en el tablero.
- Escribir en una posición del tablero.
- Imprimir el estado del tablero.


Problema 2: Escribe un método estático `char[] removeDuplicates(char[] in)` que retorne un nuevo arreglo de caracteres sin elementos duplicados. Siempre deberás mantener el primer elemento encontrado y eliminar los elementos subsecuentes.

Ejemplo:

```
removeDuplicates(new char[]{'b','d','a','b','f','a','g','a','a'}) → {'b','d','a','f','g'}.
removeDuplicates(new char[]{'a','b','a','a'}) → {'a','b'}.
```

Problema 3: Escribe un método estático `int[][] removeSmaller(int v, int[][] in)` que retorne una nueva matriz de enteros a partir del arreglo recibido **in**, pero con los valores menores a v eliminados. El nuevo arreglo retornado deberá tener la misma cantidad de filas que el arreglo **in**, pero cada fila deberá tener sólo las columnas necesarias.

removeSmaller(5, in)				
in =				
5	3	6	5	
1	2	3	5	
8	8	6	5	
-1	5	2	15	
0	0	0	0	

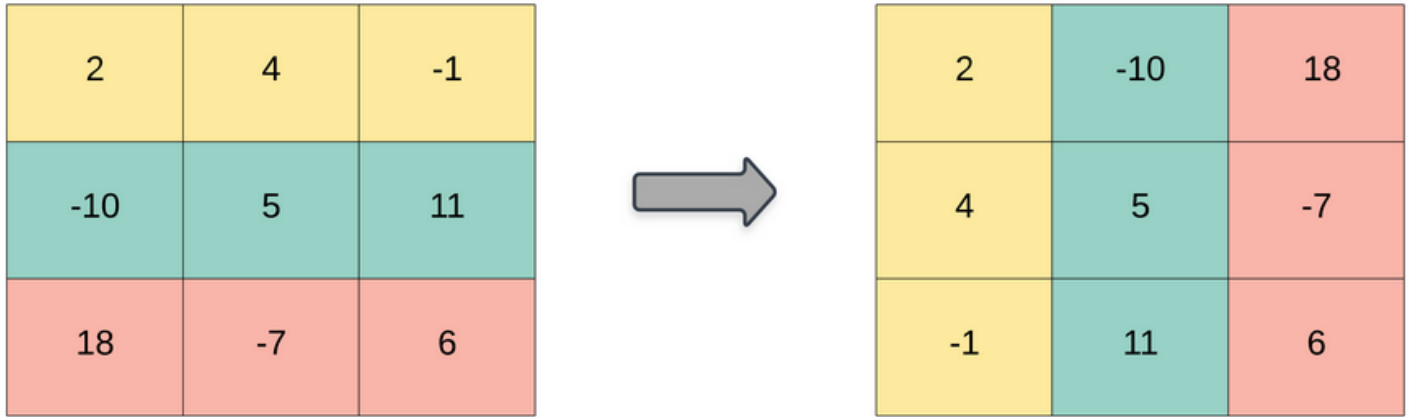


out =				
5	6	5		
5				
8	8	6	5	
5	15			

Prueba el método anterior diseñando 3 casos de prueba adicionales a los siguientes:

1. in es un arreglo no inicializado (null)
2. todos los valores de in son menores a v
3. <caso de prueba 3>
4. <caso de prueba 4>
5. <caso de prueba 5>

Problema 4: Dado un arreglo de enteros de dos dimensiones llamado *matrix*, regresa la transpuesta de *matrix*. La transpuesta de una matriz es la matriz invertida en su diagonal principal, intercambiando los índices de sus filas y columnas.



Puedes utilizar la siguiente liga para probar tu código: <https://leetcode.com/problems/transpose-matrix/>

```
int[][] transpose(int[][] matrix)
```

Problema 5: Crea un método estático void `sortMatrix(int[][] data)` que reciba como parámetro de entrada una matriz cuadrada de enteros (misma cantidad de filas y columnas) y lo ordene ascendentemente. Utiliza el algoritmo Selection Sort para este fin. Intenta no utilizar un arreglo auxiliar para resolver este problema.

Ejemplo: `sortMatrix(new int[][]{{3,7,1},{4,2,5},{9,8,6}})` → `{{1,2,3},{4,5,6},{7,8,9}}`

Problema Reto!

<https://leetcode.com/problems/cells-with-odd-values-in-a-matrix/>