

Sección 1: Diseña un programa que permita al usuario interactuar con la consola, capturando ejemplos para los problemas de la Sección 2.

Sección 2: Resuelve los siguientes problemas.

1. Diseña un método estático con la siguiente firma `int[] findElement(int[] array, int key)` que retorne como resultado un arreglo que contenga el índice de cada uno de los elementos con valor **key** encontrados en el arreglo **array**. El tamaño del arreglo retornado deberá ser igual a la cantidad de elementos encontrados. En caso de no encontrar coincidencia alguna, deberá retornar un arreglo de tamaño 0.

Ejemplo:

- `findElement(new int[]{ 7, 3, 3, 11, 8, 3, 0}, 3) → Resultado: {1, 2, 5}`
- `findElement(new int[]{6, 3, 1, 2 }, 0) → Resultado: { }`

2. Crea un método estático con la siguiente firma `void bubbleSort(String[] array)`. El método deberá acomodar los elementos lexicográficamente de mayor a menor; utilizando el algoritmo Bubble Sort. Para esto, revisa el funcionamiento del método `compareTo()` de la clase `String`.

Ejemplo:

```
bubbleSort(new String[]{"Hola", "abeja", "ave", "avenida", "avena", "zoologico"}) →
```

```
Resultado: {"zoologico", "avenida", "avena", "ave", "abeja", "Hola"}
```

3. 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 la retorne ordenada 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}}) → Returns {{1,2,3},{4,5,6},{7,8,9}}
```