

Para esta actividad, ambas secciones se entregan a mano.

**Sección 1: Ordena los siguientes arreglos, mostrando cada paso, utilizando el Selection Sort y el Bubble Sort.**

1.

2	4	1	0	10
---	---	---	---	----

2.

27	72	43	-52	13	78	11
----	----	----	-----	----	----	----

3.

1	2	3
---	---	---

**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:

```
public static void selectionSort(int[] array).
```

El método deberá acomodar los elementos descendientemente, es decir, de mayor a menor; utilizando el algoritmo Selection Sort.

**Ejemplo:**

```
selectionSort({1,5,4,3,6,7,8}) → Resultado: {8,7,6,5,4,3,1}
```

3. Crea un método estático con la siguiente firma:

```
public static void findMostFrequent(int[] array).
```

El método deberá regresar el elemento que más veces aparezca en el arreglo. Asume que siempre recibirás un arreglo con un elemento que se repita por lo menos 1 vez más que los demás elementos.

Ejemplo:

```
findMostFrequent(new int[]{5, 2, 5, 2, 5 }) → 5  
findMostFrequent(new int[]{1, 2, 3, 3, 4 }) → 3  
findMostFrequent(new int[]{1, 1, 1}) → 1
```

4. Crea un método estático con la siguiente firma

```
public 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"}
```