



## MyArray

Implemente una clase **MyArray** con las siguientes funcionalidades

- Un campo **items** privado que almacene un array de números enteros. Este campo no debe ser visible desde fuera de la definición de la clase.
- Una propiedad **Count** que contenga el tamaño de **items**. Esta propiedad no puede ser modificada fuera de la definición de la clase.
- Un constructor que reciba un array de enteros y de valores a las propiedades **items** y **Count**.
- Una forma de chequear cuál es el entero que está en la posición *i* de **items**. El elemento *i* de **items** no puede ser modificado de esta manera.
- Un método para invertir el orden de los elementos de **items**.
- Un método para, dado un entero *pos*, eliminar el elemento de la posición *pos* de **items**. Note que el tamaño de **items** debe disminuir en 1.
- Un método para, dado un entero *x*, eliminar todas las ocurrencias de *x* en **items**. Note que el tamaño de **items** disminuye.
- Un método para, dado un entero *pos* y un entero *x*, insertar *x* en la posición *pos*. Note que el tamaño de **items** aumenta en uno.

- Un método para, dado un entero  $n$ , rote el array  $n$  veces (hacia la derecha si es positivo, hacia la izquierda si es negativo).
- un método que dado un número entero  $k$  invierta cada subarray de `items` de longitud  $k$  que comience en posiciones múltiplo de  $k$ . Por ejemplo, para  $a = 1,2,3,4,5,6,7,8,9,10$  y  $k = 3$ , se deberán invertir los subarrays  $1,2,3$ ,  $4,5,6$  y  $7,8,9$ , por lo que deberá obtenerse el array  $3,2,1,6,5,4,9,8,7,10$
- Un método para ordenar `items` utilizando el algoritmo de ordenación por burbuja. (Si no lo conoce, búsquelo en Wikipedia).
- Un método para ordenar `items` utilizando el algoritmo de ordenación por inserción.
- Un método para ordenar `items` utilizando el algoritmo de ordenación por Conteo (Counting Sort).
- Un método que reciba un objeto  $a$  de tipo `MyArray` y retorne la suma del `items` propio con la del `items` de  $a$ .
- Un método que reciba otro objeto  $a$  de tipo `MyArray`, ordene tanto `items` propio como `items` de  $a$  y convierta al `items` propio en la mezcla ordenada de ambos `items`.

## Poly

Implemente una clase `Poly` que represente un polinomio. Debe contener lo siguiente

- Una propiedad (`Degree`) que contenga el grado del polinomio. Esta propiedad no puede ser modificada fuera de la definición de la clase.
- Un array `Coefficients` que tenga los coeficientes del polinomio. En la posición 0 estará el coeficiente de la variable de grado 0. Este campo no puede ser visible fuera de la definición de la clase.
- Un método que muestre el polinomio en la consola. Note que los coeficientes iguales a 0 no deben ser mostrados.
- Un método que reciba un entero  $x$  y calcule el valor del polinomio para ese  $x$ .
- Un método que reciba un escalar  $e$  y multiplique el polinomio por  $e$ .
- Un método estático que reciba dos objetos de clase `Poly` y retorne un nuevo objeto de clase `Poly` que represente la suma de las dos entradas.