Secuencias

El objetivo de este control es implementar un TAD genérico sencillo para el manejo de secuencias, así como familiarizarse con la gestión de memoria dinámica en clases C++

1) Trabajo a realizar

Debe implementarse un TAD genérico *Secuencia*<T> que incluya las siguientes operaciones:

- Un constructor por defecto `Secuencia()` que construya una secuencia vacía.
- Una operación mutadora `pon` que tome como argumento un elemento `e` y lo añada al final de la secuencia.
- Una operación observadora `elem` que tome como argumento un índice `i` y devuelva el elemento que está en la posición *i* (los elementos comienzan a indexarse desde 0). Esta operación es parcial: deberá levantar una excepción de tipo `ElementoInvalido` en caso de que el elemento solicitado no exista.
- Una operación observadora `num_elems` que devuelva el número de elementos de la secuencia.

Como representación debe utilizarse un array dinámico. Inicialmente dicho array tendrá un tamaño de `2` elementos. Cada vez que se llene, su tamaño deberá duplicarse.

Dado que la implementación estará basada en el manejo de memoria dinámica, deberán incluirse los constructores y métodos adicionales necesarios para garantizar el correcto funcionamiento, así como para evitar cualquier pérdida de memoria.

2) Código de apoyo

Se proporcionan los siguientes archivos:

- `Secuencia.h`. Este archivo debe completarse con la implementación del TAD genérico pedido.
- `main.cpp`. Programa de prueba. Este archivo no debe modificarse. El programa mantiene una secuencia de enteros, inicialmente vacía, y lee y ejecuta *comandos* hasta que termina. Los comandos leídos son de los siguientes tipos:
  - `pon` *e*: Añade el elemento *e* al final de la secuencia, e imprime el número de elementos que tiene la secuencia tras realizarse el añadido.
  - `consulta` *i*: Consulta el elemento *i*-esimo, y lo imprime por pantalla. Si tal elemento no existe, imprime `ERROR`.
  - `muestra`. Imprime la secuencia actual.
  - `limpia`. Fija la secuencia actual a la secuencia vacía. Imprime `OK`.
  - `termina`. Termina la ejecución.

  A continuación, se muestra un ejemplo de entrada / salida:

Entrada	**Salida**

`pon 5`
`pon 6`
`pon 7`
`muestra`
`consulta 1`
`consulta 3`
`limpia`
`muestra`
`pon 20`
`muestra`
`termina`
	`1`
	`2`
	`3`
	`5 6 7`
	`6`
	`ERROR`
	`OK`
	
	`1`
	`20`