

## Ejercicios del Tema 2 - laboratorio

### EJERCICIO 5 (main5\_plantilla.cpp)

Implementa un subprograma genérico que reciba una pila y escriba todos sus elementos, desde la base hasta la cima, separados por espacios.

Haz tres versiones:

- Dos iterativas (observa que se diferencian en el modo del parámetro):

```
template <class T>
void imprimeDesdeBaseIterativo (Pila<T> stack);

template <class T>
void imprimeDesdeBaseIterativoREFCTE(const Pila<T>& stack);
```

- Otra recursiva:

```
template <class T>
void imprimeDesdeBaseRecursivo(Pila<T> pila);
```

Implementa un programa principal donde pruebes las distintas versiones sobre pilas de enteros.

### EJERCICIO 7 (main7\_plantilla.cpp)

Implementa un subprograma genérico que reciba una pila y un número positivo  $n$  e invierta los  $n$  valores almacenados en la parte superior de la pila (asume que la pila tiene al menos  $n$  elementos).

```
template <class T>
void invierteN (Pila<T>& pila, int N);
```

Implementa un programa principal donde pruebes el subprograma con pilas de enteros. Para comprobar el resultado de la inversión, visualiza la pila invertida por pantalla implementando la sobrecarga del operador << que encontrarás en la plantilla.

```
template <class T>
ostream& operator<< (ostream& os, Pila<T> pila);
```

### EJERCICIO 12 (main12\_plantilla.cpp)

Extiende la implementación de la cola basada en lista enlazada disponible en el campus virtual con una nueva operación, numElemos, que devuelva el número de elementos almacenados en ella. Hazlo de forma que su complejidad sea  $O(1)$ .

Prueba el TAD extendido usando el programa principal proporcionado. Deberás completar la implementación de la sobrecarga del operador << que figura en la plantilla.

```
template <class T>
ostream& operator<< (ostream& os, Cola<T> cola);
```

Como trabajo adicional, puedes extender la implementación del TAD incorporando la sobrecarga del operador << como función amiga:

```
friend ostream& operator<<(ostream& out, const Cola<T>& c) {
    // incluye tu código aquí
}
```

Y pruébalo en el programa principal proporcionado (quitando previamente la sobrecarga de << que habrás implementado en la plantilla).