Práctica 9: TAD PILA

Sea el **TAD PILA** visto en clase con la siguiente **especificación en C++**:

```
void iniciarPila(pila & P);
/*
{Pre: }
{Post: Inicia P como una pila vacía}
*/

void apilar(pila & P,telemento d);
/*
{Pre: la pila P ha debido ser previamente inicializada }
{Post: Apila en la pila P el elemento d}
*/

bool pilaVacia(pila P);
/*
{Pre: la pila P ha debido ser previamente inicializada }
{Post: Si la pila P está vacía devuelve el valor verdad. En caso
    contrario devuelve el valor falso}
*/

telemento cima(pila P);
/*
{Pre: la pila P ha debido ser previamente inicializada y no está
    vacía}
{Post:Devuelve el elemento más reciente en la pila P y no modifica P}
*/

void desapilar(pila & P);
/*
{Pre: la pila P ha debido ser previamente inicializada y no está
    vacía}
{Post: Modifica la pila P eliminando el último elemento apilado}
*/
```

Os proporcionamos los ficheros `pila.h` (con la especificación del TAD) y `pila.cpp` (con una implementación que vosotros como usuarios no conocéis). **Nota:** la implementación que os damos es para pilas que contienen números enteros: `typedef int telemento;`

Comienza descargando los ficheros `pila.h` y `pila.cpp` y añádelos a un proyecto de NetBeans.

1. La primera parte de la práctica consiste en diseñar los siguientes subprogramas que **usen** el TAD PILA (es decir, solo pueden utilizar las operaciones definidas en el TAD PILA, no la implementación de dichas operaciones o la definición concreta del tipo pila).

    a. Diseñar una **acción** para crear una pila con enteros leídos por teclado (terminar al introducir el entero 0).

    b. Diseñar una **acción** para mostrar el contenido de una pila de enteros.

    c. Diseñar una **función** que devuelva el número de enteros que hay en una pila.

    d. Diseñar una **acción** que invierta el contenido de una pila de enteros en otra.

    e. Construir una **acción** que copie el contenido de una pila de enteros en otra.

    f. Construir una **función** que decida si dos pilas de enteros son o no iguales.

    g. Diseñar una **función** que dado un entero decida si es o no capicúa. (Utilizar el tipo pila).

2. Ahora vais a trabajar como **implementadores** del TAD PILA. Os informamos de que la implementación que os habíamos proporcionado antes era estática (haciendo uso de registros y vectores). Ahora vosotros debéis dar una **implementación dinámica del TAD PILA** (mediante listas de celdas enlazadas). Para dar dicha implementación deberéis:

    - Redefinir el tipo `pila` definido en el fichero `pila.h`.
    - Utilizar la nueva representación del tipo `pila` para implementar las operaciones definidas en el fichero `pila.cpp`.

3. Comprobar que todos los subprogramas que habíais implementado como usuarios del TAD PILA en el apartado 1 funcionan correctamente con vuestra implementación dinámica del TAD PILA.