Práctica 4: Transformación de un diseño recursivo a uno iterativo

1. Objetivos de esta práctica

- **Diseñar algoritmos recursivos**. Se diseñará un algoritmo recursivo sencillo.
- **Diseñar estructuras de datos.** Se transformará el diseño recursivo realizado en uno iterativo con idéntica funcionalidad.

2. Transformación de diseños recursivos en iterativos

2.1. Implementación de la función recursiva `mayoresR()`

Se debe diseñar e implementar una función exclusivamente recursiva, `mayoresR()`, especificada de la siguiente manera:

```
// Pre: De la entrada estándar se puede leer una secuencia de
//       enteros: a_1, a_2, ..., a_k tal que a_k < 0 ∧ ∀i ∈ [1, k−1].a_i ≥ 0.
// Post: Escribe por la salida estándar (en cualquier orden) todos
//      los enteros a_i tal que i < k ∧ a_i > −a_k. Devuelve a_k.
int mayoresR();
```

Listado 1: Especificación de la función `mayoresR()`

La función `mayoresR()` no tiene parámetros y no puede utilizar variables globales.

2.2. Implementación de la función iterativa `mayoresI()`

Aplicando las técnicas de transformación de diseños recursivos en iterativos y, si fuera necesario, haciendo uso del tipo de datos `PilaEnt` diseñado en la práctica anterior, se debe diseñar e implementar una función exclusivamente iterativa, `mayoresI()`, con la misma funcionalidad que `mayoresR()`.

Las funciones `mayoresR()` y `mayoresI()` se implementarán en un fichero denominado `mayores.cpp`. Estas funciones serán probadas mediante llamadas contenidas en el programa principal, `main()`, que se implementará en el mismo fichero.

A modo de ejemplo, la salida de un programa que invoca primero a la función recursiva y después a la iterativa podría ser:

```
$ ./mayores

Diseño recursivo. Introduce una secuencia de enteros:
4 7 2 9 6 -5
6 9 7
```

```
El último entero es: -5

Diseño iterativo. Introduce una secuencia de enteros:
4 7 2 9 6 -5
6 9 7
El último entero es: -5
```

La función `mayoresI()` no tiene parámetros y no puede utilizar variables globales.

2.3. Fichero `Makefile`

Se debe escribir un fichero **`Makefile`** que permita compilar el programa **`mayores`** mediante la orden:

```
$ make
```

3. Resultados del trabajo desarrollado en las cuatro primeras prácticas

Como resultado de las cuatro primeras prácticas, cada alumno dispondrá en su cuenta de un directorio (carpeta) denominado **`programacion2`** dentro del cual se encontrarán los directorios (carpetas) y ficheros que se detallan a continuación.

- Carpeta **`programacion2/funciones`** con los siguientes ficheros: **`pilaEnt.hpp`** y **`pilaEnt.cpp`**
- Carpeta **`programacion2/practica1`**, con los siguientes ficheros: **`tiempoReaccion.cpp`**, **`generarTabla.cpp`**, **`genNum.hpp`**, **`genNum.cpp`** y **`aproxPI.cpp`**.
- Carpeta **`programacion2/practica2`** con los siguientes ficheros: **`calculos.hpp`**, **`calculos.cpp`**, **`pruebasCal.cpp`** y **`Make_pruebasCal`**
- Carpeta **`programacion2/practica3`** con los siguientes ficheros: **`funcionesPilaEnt.hpp`**, **`funcionesPilaEnt.cpp`**, **`pruebasPila.cpp`** y **`Makefile`**.
- Carpeta **`programacion2/practica4`** con los siguientes ficheros:
  - Fichero **`mayores.cpp`** que contiene las funciones `mayoresR()`, `mayoresI()` y `main()`.
  - Fichero **`Makefile`** para compilar **`mayores.cpp`**.