

## Práctica utilización TADs

Se trata de comprobar que los TADs implementados en la práctica obligatoria de EDAI son correctos, y pueden utilizarse de manera análoga a como se utilizan los tipos de datos primitivos. Para ello se plantea la implementación de las siguientes funciones, todas ellas aparecen en la asignatura EDAI:

1. Probar la función que implementa el **equilibrio de símbolos** en una expresión, presentada en el tema 6 de EDAI, utilizando el TAD Pila. Esta función se proporciona implementada en el fichero de ayuda pruebaTADs.c
2. Codificar una función que permita determinar si una frase es **palíndromo**, se puede leer lo mismo de izquierda a derecha y de derecha a izquierda, una vez eliminados los blancos ( Ej.: “anula la luz azul a la luna”). Utilizar el TAD Pila y el TAD Cola, añadiendo cada carácter de la frase a las dos estructuras a la vez. La extracción simultánea de caracteres de ambas y su comparación determinará si la frase es palíndromo o no.
3. Modificar el algoritmo de ordenación **RadixSort** visto en tema 4 de EDAI de forma que utilice el TAD Lista implementado.

Para no penalizar el rendimiento lineal de este algoritmo, ni bajar el nivel de abstracción del TAD Lista, es necesario añadir la siguiente operación a la especificación dada en EDAI.

```
int traspasarNodo(tipoPosicion p, Lista *la, tipoPosicion q, Lista *lb)
```

Operación que traspasa el nodo de la posición  $p$ , indicada en el primer parámetro, de la lista referenciada por el parámetro **la**, a la posición  $q$ , indicada en el tercer parámetro, de la lista referenciada por el parámetro **lb**. Si **la** es la colección de elementos  $a_1, a_2, a_{p-1}, a_p, a_{p+1}, \dots, a_n$  y **lb** es la colección  $b_1, b_2, b_{q-1}, b_q, b_{q+1}, \dots, b_n$ , el resultado de la ejecución de esta función será

- La lista **la** debe quedar con los elementos  $a_1, a_2, a_{p-1}, a_{p+1}, \dots, a_n$ .
- La lista **lb** debe quedar con los elementos  $b_1, b_2, b_{q-1}, a_p, b_q, b_{q+1}, \dots, b_n$ . Si  $q$  es fin(lb) entonces  $a_p$  se añade al final.
- Si la operación se ejecuta con éxito devolverá cero. Si alguna de las listas o posiciones no son válidas un entero negativo.

Si se utiliza correctamente el TAD Lista accediendo a sus posiciones solamente con las operaciones especificadas, los parámetros de tipo posición serán correctos, se evitarán recorridos en las lista y obtendremos una operación de **orden constante**.

**Nota:** Como se ha estudiando en teoría la especificación de un tipo abstracto de datos es única, pero una vez definido puede implementarse de varias formas. Sólo es necesario respetar la especificación para que, fuera de la sección donde se define, pueda utilizarse como si se tratara de un tipo de datos primitivo (encapsulación).

En este sentido, se recuerda que existen enunciados adicionales de prácticas en el último tema de Estructuras de Datos y Algoritmos I, dedicado a los Tipos Abstractos de Datos. Las diferentes implementaciones que se proponen puede ayudar a una mejor comprensión de la utilización de la abstracción, por eso se recomienda su implementación.