

## Control 1

Tiempo disponible: 1 hora

Extiende la implementación basada en nodos doblemente enlazados del TAD Lista proporcionado con la siguiente operación:

```
void cortaPorK (unsigned int k, Lista<T>& destino)
```

que recibe un valor  $k \geq 0$  y una lista destino vacía, y quita en la lista receptora los elementos de que van desde la posición  $k$  hasta el final y los mete en la lista destino manteniendo su orden relativo. Se supone que el primer elemento de una lista tiene la posición 0, el segundo tiene la posición 1, y así sucesivamente. La operación no surtirá ningún efecto para valores de  $k$  que sean mayores o iguales que la longitud de la lista receptora.

De esta forma, si la lista receptora contiene los siguientes elementos (de principio a fin):

0 1 2 3 4 5 6

tras ejecutar dicha operación con un valor  $k = 2$ , los elementos de la lista receptora, de principio a fin, serán:

0 1

y los de la lista destino, de principio a fin, serán:

2 3 4 5 6

La operación *cortaPorK* no puede crear ni destruir nodos (es decir, no puede invocar ni directa ni indirectamente operaciones de manejo de memoria dinámica –new, delete), ni tampoco puede realizar asignaciones entre contenidos de nodos.

Además de implementar la operación, debes indicar razonadamente la complejidad de la misma.

Ten en cuenta que:

- Debes usar el archivo lista.h que se adjunta con este enunciado y no el visto en clase. En el archivo lista.h adjunto a este enunciado está la implementación del TAD lista basado en nodos doblemente enlazados vista en clase, extendida con métodos para imprimir por pantalla la lista de principio a fin y de fin a principio. Incluye también, al comienzo de la parte pública, el método 'cortaPorK' que es el que hay que implementar además de indicar razonadamente su complejidad.
- Se proporciona el archivo main.cpp en el que se implementa toda la lógica de entrada / salida necesaria. NO MODIFIQUES NADA EN ÉL.
- El archivo leeme.txt incluye información sobre el formato de la entrada / salida, así como algunos ejemplos de entrada junto con las salidas producidas.