



Alumno:

Grupo:

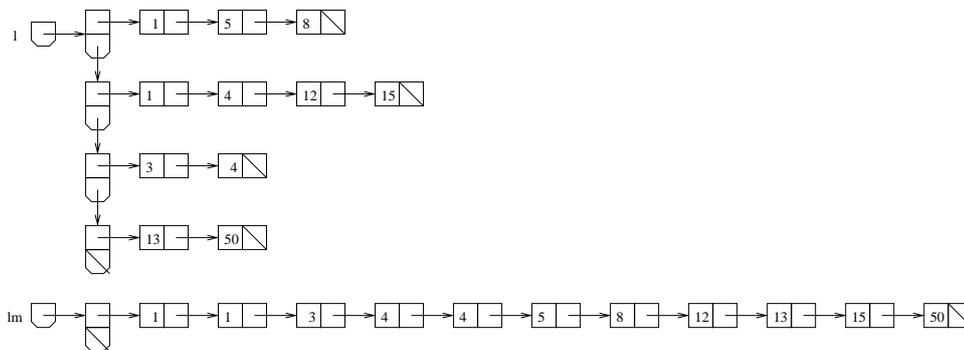
- 1. Codificar un algoritmo que tome como parámetro una lista de listas de números, donde los números están ordenados ascendentemente dentro de cada lista y produce como parámetro de salida una lista de números ordenados ascendentemente. (3.5 pts.)

TIPOS

```
REGISTRO NodoNum
  N dato
  NodoNum *sig
FIN
NodoNum *Lista
REGISTRO NodoLista
  Lista lst
  NodoLista *sig
FIN
NodoLista *Listal
```

ALGORITMO Mezclar(ES Listal l, S Listal lm)

Antes de la función el parámetro de salida es una lista vacía y la lista inicial como l en el ejemplo del dibujo; después de la función el parámetro lista inicial queda vacío y el de salida quedaría como el lm del dibujo



- 2. Una ListaVentana es una estructura lineal en la que hay un elemento distinguido, que es el que está en la ventana. Ese es el único elemento que, en un momento dado, puede ser directamente consultado, modificado o eliminado. Además, sólo se puede insertar un elemento en la posición siguiente al elemento distinguido. El nuevo elemento insertado pasa a ser el distinguido. La ventana se puede mover hacia adelante o hacia atrás un determinado número de posiciones. Si se intenta mover más allá de un extremo, se quedará en el elemento que esté en ese extremo. Los elementos están ordenados por su posición dentro de la lista, desde la posición 1 hasta la posición dada por el número de elementos. De una ListaVentana se puede saber en qué posición está la ventana y cuál es el número de elementos que tiene (3.5 pts.)

INTERFAZ CLASE ListaVentana TIPOS

```

TBase
METODOS
  Crear()
  Destruir()
  Consultar(S TBase el)
  Modificar(E TBase el)
  Borrar()
  Añadir(E TBase el)
  // n < 0 moverse hacia el principio
  // n > 0 moverse hacia el final
  Moverse(E Z n)
  N PosElemActual()
  N NumElemsTotal()
FIN ListaVentana

```

Se pide la implementación de la clase `ListaVentana` en base a pilas (con el interfaz visto en clase y manejadas a nivel de usuario). En una pila se tendrán los elementos que están desde la posición 1 hasta la ventana y en otra desde la posición siguiente hasta la última.

Como usuario (fuera de la clase) hacer un procedimiento:

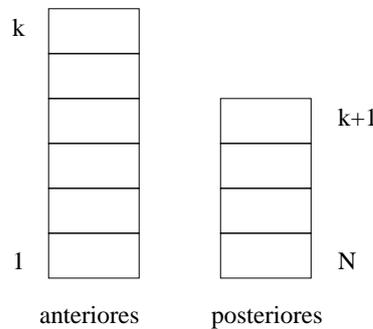
```

ALGORITMO BorrarElemPos(ES ListaVentana lv, E N n)

```

que borre el elemento que esté en la posición indicada por el parámetro. Si esa posición no existe, la lista no se modifica. Tras el borrado la ventana debe quedarse en la misma posición. Si la posición borrada es la de la ventana, la ventana pasa a la siguiente posición, pero si la ventana ya estaba al final de la lista, se queda en el que pasa a ser el último elemento.

En la figura se muestra el posible estado de una lista que contiene N elementos, estando numerado el primero con el número 1, y la actual es la número k .



3. Recorrer el perímetro de un árbol binario lleno. En el recorrido solicitado se pasa por todos los elementos de la rama más izquierda desde la raíz hasta la hoja de esa rama, después se sigue de izquierda a derecha por el resto de las hojas y se sube por la rama derecha hasta volver a la raíz. Por ejemplo, el recorrido del perfil de este árbol devolvería: **a, b, d, k, l, m, n, p, q, r, s, h, c**. Nótese que los elementos que están en los vértices sólo aparecen una vez. (3 pts.)

