

Nombre y apellidos: \_\_\_\_\_

Carné: \_\_\_\_\_

## Parcial II (50 %)

1. Suponga que dado un número de carné, tal como 1710897, se puede obtener una secuencia de números  $S$ , de la siguiente forma:  $S = [17, 10, 89, 7]$ . Sea  $R$  una secuencia definida como  $R = [15, 27, 75, 34, 89, 57]$ . Se tiene que la secuencia  $T$ , es la concatenación de las secuencias  $S$  y  $R$ , es decir,  $T = S + R = [17, 10, 89, 7, 15, 27, 75, 34, 57]$ . La secuencia  $T$  no debe tener elementos repetidos, por lo que si en  $R$  y  $S$  tienen un elemento en común, entonces solo se agrega uno de los elementos a  $T$ . Se quiere insertar en una tabla de hash la secuencia de claves de  $T$ . La tabla de hash tiene  $m = 7$  casillas y la resolución de colisiones se realiza mediante el método de encadenamiento. La función de hash es  $h(k) = k \bmod m$ , para una clave  $k$ . Se quiere que muestre la tabla de hash resultante después de la inserción de la secuencia de claves. Para la inserción de cada clave debe indicar la casilla en la tabla en la cual la clave es insertada y debe mostrar la tabla resultante de la inserción. **Para la secuencia S, deben usar su carné** (Valor 13 %).
2. Suponga que se tiene una tabla de hash en la cual la resolución de colisiones se realiza mediante el método direccionamiento abierto. Sea  $m$  el número de casillas de la tabla de hash. Se tiene que  $m = 2^q$ , donde  $q$  es un número entero mayor que 1. La función de hash a usar es  $h(k) = k \bmod m$  para una clave  $k$ . Explique las debilidades de esta función de hash para la tabla de hash descrita. Si es posible, muestre ejemplos. (Valor 12 %).
3. Usando apuntadores, se define la estructura de datos lista doblemente enlazada, como una lista enlazada, la cual está compuesta por elementos de tipo celdas y un apuntador a la cabeza (head) o inicio de la lista. Cada una de las estructuras de tipo celda, contiene tres componentes: un campo llamado *clave* de tipo entero, un campo *prev* que es un apuntador a un elemento tipo celda y un campo un campo *next*, que también es un apuntador a un elemento tipo celda. El estructura de la lista doblemente enlazada se muestra en Figura 1.  
  
Se quiere que realice un procedimiento llamado **EliminarTodasOcurrencias**, que elimine todas las ocurrencias de un elemento dado  $x$ , en la lista enlazada. La parte a) de la Figura 1 muestra una lista enlazada  $L$  y la parte b) muestra la lista  $L$  después de aplicar el procedimiento **EliminarTodasOcurrencias**, con el elemento 1. La especificación del procedimiento se da a continuación.  
  
(Valor 13 %).

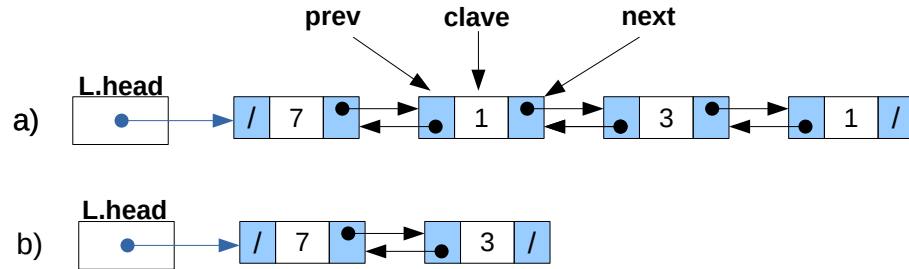


Figura 1: Ejemplo de dos listas doblemente enlazadas. La Figura a) presenta una lista  $L$  con cuatro elementos. La Figura b) muestra la lista  $L$ , después de aplicar el procedimiento **EliminarTodasOcurrencias** con el elemento 1.

```

proc EliminarTodasOcurrencias ( in out  $L$  : Lista; in  $x$  : Entero )
{ Pre:   True }
{ Post: La lista de salida  $L$  contiene todos de la lista inicial de entrada
 $L_0$ , excepto cualquier ocurrencia del elemento  $x$ .
La lista  $L$  mantiene el orden de los elementos de  $L_0$ .
En caso de que  $x$  no se encuentre en  $L_0$ , entonces  $L$  y  $L_0$  son la misma lista. }

```