

Lista de funciones y procedimientos utilizados en las listas

procedure createEmptyList(VAR L:tList);

OBJETIVO: Crea una lista vacía.

ENTRADA: Lista

SALIDA: Lista vacía

POSTCONDICIÓN: La lista queda inicializada y no contiene elementos.

procedure CreateNodo(l:tItem; var P:tPosL);

{SOLO DYNAMICLIST}

OBJETIVO: Crea el elemento deseado de la lista

ENTRADA: Partido y numero de votos con la posición donde queremos introducirlo

SALIDA: Lista con ese nuevo dato.

function isEmptyList(L:tList):boolean;

OBJETIVO: Determina si la lista está vacía.

ENTRADA: La lista.

SALIDA: TRUE si la lista está vacía, FALSE en caso contrario.

function first(L:tList): tPosL;

OBJETIVO: Obtener el primer elemento de la lista.

ENTRADA: La lista

SALIDA: La posición del primer elemento de la lista.

PRECONDICIÓN: La lista no está vacía.

function last(L: tList): tPosL;

OBJETIVO: Devolver la posición del último elemento de la lista.

ENTRADA: La lista.

SALIDA: La posición del último elemento de la lista.

PRECONDICIÓN: La lista no está vacía.

function next(P:tPosL; L:tList):tPosL;

OBJETIVO: Obtener la posición en la lista del elemento siguiente a la posición indicada o NULL si la posición no tiene siguiente.

ENTRADA: La lista y la posición deseada.

SALIDA: Devuelve la posición siguiente a la indicada, si no NULL.

PRECONDICIÓN: La posición indicada es una posición válida en la lista.

function previous (P:tPosL; L:tList): tPosL;

OBJETIVO: Devolver la posición de la lista del elemento anterior a la posición indicada o NULL si la posición no tiene anterior.

ENTRADA: La lista y la posición deseada.

SALIDA: Devuelve la posición anterior a la indicada si tiene, si no NULL.

PRECONDICIÓN: La posición indicada es una posición válida en la lista.

function insertItem(l:tItem; P:tPosL; VAR L:tList): boolean;

3.1

OBJETIVO: Inserta elemento en la lista antes de la posición indicada. Si la posición es NULL se añade al final. Devuelve true si

el elemento fue insertado; false en caso contrario.

ENTRADA: El elemento, la posición donde queremos insertarlo y la lista.

SALIDA: Devuelve la lista con el elemento insertado en la posición indicada (si la posición es NULL la añade al final), y devuelve true si el elemento se insertó, false en caso contrario.

PRECONDICIONES: La posición indicada es una posición válida en la lista o bien nula (NULL).

POSTCONDICIONES: Tanto la posición del elemento eliminado como aquellas de los elementos de la lista a continuación del mismo pueden cambiar de valor.

procedure deleteAtPosition(P:tPosL; VAR L:tList);

OBJETIVO: Elimina de la lista el elemento que ocupa la posición indicada.

ENTRADA: La posición a borrar y la lista.

SALIDA: La lista con la posición eliminada.

PRECONDICIÓN: La posición indicada es una posición válida en la lista.

POSTCONDICIÓN: Tanto la posición del elemento eliminado como aquellas de los elementos de la lista a continuación del mismo pueden cambiar de valor.

function getItem(P:tPosL; L:tList): tItem;

OBJETIVO: Devuelve el contenido del elemento de la lista que ocupa la posición indicada.

ENTRADA: La posición del elemento que queremos obtener y la lista.

SALIDA: El dato buscado en la lista.

PRECONDICIÓN: La posición indicada es una posición válida en la lista.

procedure updateVotes(nv: tNumVotes; P:tPosL; VAR L:tList);

OBJETIVO: Modifica el número de votos del elemento situado en la posición indicada.

ENTRADA: numero de votos que queremos modificar con su posición correspondiente y la lista.

SALIDA: el numero de votos actualizado.

PRECONDICIÓN: La posición indicada es una posición válida en la lista.

POSTCONDICIÓN: El orden de los elementos de la lista no se ve modificado.

function findItem(name:tPartyName; L:tList): tPosL;

OBJETIVO: Devolver la posición del primer elemento de la lista cuyo nombre de partido se corresponda con el indicado (o NULL si no existe tal elemento).

ENTRADA: El nombre del partido y la lista.

SALIDA: La posición de la lista donde está ese partido.