/\* TDA Lista

\* Implementación Simplemente Enlazada

\* Archivo : ListaPuntajesTotal.h

\* Versión : 1.1

\*/

#ifndef \_\_LISTAPUNTAJESTOTAL\_H\_\_

#define \_\_LISTAPUNTAJESTOTAL\_H\_\_

#ifndef NULL

#define NULL 0

#endif

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Definiciones de Tipos de Datos \*/

/\*--------------------------------\*/

/\* tipo enumerado para realizar comparaciones \*/

enum ResultadoComparacionPuntajesTotal {

MAYORPUNTAJETOTAL,

IGUALPUNTAJETOTAL,

MENORPUNTAJETOTAL,

};

/\* Tipo de Informacion que esta contenida en los Nodos de la

Lista, identificada como PuntajesTotal. \*/

typedef struct{

int posicion;

int idJugador;

float puntaje;

}PuntajesTotal;

/\* Tipo de Estructura de los Nodos de la Lista. \*/

struct NodoListaPuntajesTotal {

PuntajesTotal dato; // dato almacenado

NodoListaPuntajesTotal\* sgte; // puntero al siguiente

};

/\* Tipo de Puntero a los Nodos de la Lista, el cual se usa para recorrer

la Lista y acceder a sus Datos. \*/

typedef NodoListaPuntajesTotal\* PtrNodoListaPuntajesTotal;

/\* Tipo de Estructura de la Lista \*/

struct ListaPuntajesTotal{

PtrNodoListaPuntajesTotal primero; // puntero al primer nodo de la lista

};

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Definicion de Primitivas \*/

/\*--------------------------\*/

/\*

pre : la lista no debe haber sido creada.

post: lista queda creada y preparada para ser usada.

lista : estructura de datos a ser creado.

\*/

void crearListaPuntajesTotal(ListaPuntajesTotal &lista);

/\*----------------------------------------------------------------------------\*/

/\*

pre : lista Creada con crearListaPuntajesTotal().

post: Devuelve true si lista esta vacia, sino devuelve false.

lista : lista sobre la cual se invoca la primitiva.

\*/

bool listaVaciaPuntajesTotal(ListaPuntajesTotal &lista);

/\*----------------------------------------------------------------------------\*/

/\*

pre : lista Creada con crearListaPuntajesTotal().

post: devuelve la representacion de lo Siguiente al último Nodo de la lista,

o sea el valor Null, que en esta implementacion representa el final de

la lista.

return representación del finpuntajeTotal de la lista.

\*/

PtrNodoListaPuntajesTotal finPuntajesTotal();

/\*----------------------------------------------------------------------------\*/

/\*

pre : lista Creada con crearListaPuntajeTotal().

post: devuelve el puntero al primer elemento de la lista, o devuelve finPuntajeTotal() si

esta vacia

lista : lista sobre la cual se invoca la primitiva.

return puntero al primer nodo.

\*/

PtrNodoListaPuntajesTotal primero(ListaPuntajesTotal &lista);

/\*----------------------------------------------------------------------------\*/

/\*

pre : lista Creada con crearListaPunatajeTotal().

post: devuelve el puntero al nodo proximo del apuntado, o devuelve finPuntajeTotal()

ptrNodo apuntaba a finPuntajeTotal() o si lista esta vacia.

lista : lista sobre la cual se invoca la primitiva.

prtNodo : puntero al nodo a partir del cual se requiere el siguiente.

return puntero al nodo siguiente.

\*/

PtrNodoListaPuntajesTotal siguiente(ListaPuntajesTotal &lista, PtrNodoListaPuntajesTotal ptrNodo);

/\*----------------------------------------------------------------------------\*/

/\*

pre : lista Creada con crearListaPuntajeTotal().

ptrNodo es un puntero a un nodo de lista.

post: devuelve el puntero al nodo anterior del apuntado, o devuelve finPuntajesTotal() si

ptrNodo apuntaba al primero o si lista esta vacia.

lista : lista sobre la cual se invoca la primitiva.

prtNodo : puntero al nodo a partir del cual se requiere el anterior.

return puntero al nodo anterior.

\*/

PtrNodoListaPuntajesTotal anterior(ListaPuntajesTotal &lista, PtrNodoListaPuntajesTotal ptrNodo);

/\*----------------------------------------------------------------------------\*/

/\*

pre : lista creada con crearListaPuntajesTotal().

post: devuelve el puntero al ultimo nodo de la lista, o devuelve finPuntajeTotal() si

si lista esta vacia.

lista : lista sobre la cual se invoca la primitiva.

return puntero al último nodo.

\*/

PtrNodoListaPuntajesTotal ultimo(ListaPuntajesTotal &lista);

/\*----------------------------------------------------------------------------\*/

/\*

pre : lista creada con crearListaPuntajesTotal().

post: agrega un nodo nuevo al principio de la lista con el dato proporcionado

y devuelve un puntero a ese elemento.

lista : lista sobre la cual se invoca la primitiva.

dato : elemento a adicionar al principio de la lista.

return puntero al nodo adicionado.

\*/

PtrNodoListaPuntajesTotal adicionarPrincipio(ListaPuntajesTotal &lista, PuntajesTotal puntajes);

/\*----------------------------------------------------------------------------\*/

/\*

pre : lista creada con crearListaPuntajesTotal().

post: agrega un nodo despues del apuntado por ptrNodo con el dato

proporcionado y devuelve un puntero apuntado al elemento insertado.

Si la lista esta vacía agrega un nodo al principio de esta y devuelve

un puntero al nodo insertado. Si ptrNodo apunta a finPuntajeTotal() no inserta

nada y devuelve finPuntajeTotal().

lista : lista sobre la cual se invoca la primitiva.

dato : elemento a adicionar.

ptrNodo : puntero al nodo después del cual se quiere adicionar el dato.

return puntero al nodo adicionado.

\*/

PtrNodoListaPuntajesTotal adicionarDespues(ListaPuntajesTotal &lista, PuntajesTotal puntajes, PtrNodoListaPuntajesTotal ptrNodo);

/\*----------------------------------------------------------------------------\*/

/\*

pre : lista creada con crearListaPuntajeTotal().

post: agrega un nodo al final de la lista con el dato proporcionado y devuelve

un puntero al nodo insertado.

lista : lista sobre la cual se invoca la primitiva.

dato : elemento a adicionar al final de la lista.

return puntero al nodo adicionado.

\*/

PtrNodoListaPuntajesTotal adicionarFinal(ListaPuntajesTotal &lista, PuntajesTotal puntajes);

/\*----------------------------------------------------------------------------\*/

/\*

pre : lista creada con crearListaPuntajeTotal().

post: agrega un nodo con el dato proporcionado antes del apuntado por ptrNodo

y devuelve un puntero al nodo insertado. Si la lista esta vacia no

inserta nada y devuelve finPuntajeTotal(). Si ptrNodo apunta al primero, el nodo

insertado sera el nuevo primero.

lista : lista sobre la cual se invoca la primitiva.

dato : elemento a adicionar.

ptrNodo : puntero al nodo antes del cual se quiere adicionar el dato.

return puntero al nodo adicionado.

\*/

PtrNodoListaPuntajesTotal adicionarAntes(ListaPuntajesTotal &lista, PuntajesTotal puntajes, PtrNodoListaPuntajesTotal ptrNodo);

/\*----------------------------------------------------------------------------\*/

/\*

pre : lista creada con crearListaPuntajeTotal(), no vacia. ptrNodo es distinto de finPuntajeTotal().

post: coloca el dato proporcionado en el nodo apuntado por ptrNodo.

lista : lista sobre la cual se invoca la primitiva.

dato : elemento a colocar.

ptrNodo : puntero al nodo del cual se quiere colocar el dato.

\*/

void colocarDatoPuntajesTotal(ListaPuntajesTotal &lista, PuntajesTotal &puntajes, PtrNodoListaPuntajesTotal ptrNodo);

/\*----------------------------------------------------------------------------\*/

/\*

pre : lista creada con crearListaPuntajeTotal(), no vacia. ptrNodo es distinto de finPuntajeTotal().

post: devuelve el dato del nodo apuntado por ptrNodo.

lista : lista sobre la cual se invoca la primitiva.

dato : elemento obtenido.

ptrNodo : puntero al nodo del cual se quiere obtener el dato.

\*/

void obtenerDatoPuntajesTotal(ListaPuntajesTotal &lista, PuntajesTotal &puntajes, PtrNodoListaPuntajesTotal ptrNodo);

/\*----------------------------------------------------------------------------\*/

/\*

pre : lista creada con crearListaPuntajeTotal().

post: elimina el nodo apuntado por ptrNodo. No realiza accion si la lista

esta vacia o si ptrNodo apunta a finPuntajeTotal().

lista : lista sobre la cual se invoca la primitiva.

ptrNodo : puntero al nodo que se desea eliminar.

\*/

void eliminarNodoPuntajesTotal(ListaPuntajesTotal &lista, PtrNodoListaPuntajesTotal ptrNodo);

/\*----------------------------------------------------------------------------\*/

/\*

pre : lista creada con crearListaPuntajeTotal().

post: si la lista no esta vacia, elimina su nodo primero, sino no realiza

accion alguna.

lista : lista sobre la cual se invoca la primitiva.

\*/

void eliminarNodoPrimeroPuntajesTotal(ListaPuntajesTotal &lista);

/\*----------------------------------------------------------------------------\*/

/\*

pre : lista creada con crearListaPuntajeTotal().

post: si la lista no esta vacia elimina su nodo ultimo,

sino no realiza accion.

lista : lista sobre la cual se invoca la primitiva.

\*/

void eliminarNodoUltimoPuntajesTotal(ListaPuntajesTotal &lista);

/\*----------------------------------------------------------------------------\*/

/\*

pre : lista creada con crearListaPuntajeTotal().

post: elimina todos los Nodos de la lista quedando destruida e inhabilitada

para su uso.

lista : lista sobre la cual se invoca la primitiva.

\*/

void eliminarListaPuntajesTotal(ListaPuntajesTotal &lista);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Definición de Operaciones Adicionales \*/

/\*---------------------------------------\*/

/\*

pre : lista fue creada con crearListaPuntajeTotal().

post: si el dato se encuentra en la lista, devuelve el puntero al primer nodo

que lo contiene. Si el dato no se encuentra en la lista devuelve finPuntajeTotal().

lista : lista sobre la cual se invoca la primitiva.

dato : elemento a localizar.

return puntero al nodo localizado o finPuntajeTotal().

\*/

PtrNodoListaPuntajesTotal localizarDato(ListaPuntajesTotal &lista , PuntajesTotal puntajes);

/\*----------------------------------------------------------------------------\*/

/\*

pre : lista fue creada con crearListaPuntajeTotal() y cargada con datos ordenados de

menor a mayor respecto del sentido progresivo.

post: agrega a la lista el dato manteniendo el orden pero con multiples

valores iguales y devuelve un puntero al nodo insertado.

lista : lista sobre la cual se invoca la primitiva.

dato : elemento a insertar.

return puntero al nodo insertado.

\*/

PtrNodoListaPuntajesTotal insertarDato(ListaPuntajesTotal &lista, PuntajesTotal puntajes);

/\*----------------------------------------------------------------------------\*/

/\*

pre : la lista fue creada con crearListaPuntajeTotal().

post : elimina el dato de la lista, si el mismo se encuentra.

lista : lista sobre la cual se invoca la primitiva.

dato : elemento a eliminar.

\*/

void eliminarDatoPuntajesTotal(ListaPuntajesTotal &lista, PuntajesTotal puntajes);

/\*----------------------------------------------------------------------------\*/

/\*

pre : la lista fue creada con crearListaPuntajeTotal().

post : reordena la lista.

lista : lista sobre la cual se invoca la primitiva.

\*/

void reordenarPuntajesTotal(ListaPuntajesTotal &lista);

/\*----------------------------------------------------------------------------\*/

/\*

pre : la lista fue creada con crearListaPuntajeTotal().

post : devuelve la cantidad de datos que tiene la lista.

lista : lista sobre la cual se invoca la primitiva.

\*/

int longitudPuntajesTotal(ListaPuntajesTotal &lista);

#endif