/\* TDA ListaJugador

\* Implementación Simplemente Enlazada

\* Archivo : ListaJugador.h

\* Versión : 1.1

\*/

#ifndef LISTAJUGADOR\_H\_INCLUDED

#define LISTAJUGADOR\_H\_INCLUDED

#ifndef NULL

#define NULL 0

#endif

#include "Jugador.h"

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

/\* Definiciones de Tipos de Datos \*/

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

/\* tipo enumerado para realizar comparaciones \*/

enum ResultadoComparacionJugador {

MAYORJugador,

IGUALJugador,

MENORJugador

};

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

ListaJugador, identificada como Jugador. \*/

//typedef Jugador Jugador;

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

struct NodoListaJugador {

Jugador dato; // dato almacenado

NodoListaJugador\* sgte; // puntero al siguienteListaJugador

};

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

la ListaJugador y acceder a sus Datos. \*/

typedef NodoListaJugador\* PtrNodoListaJugador;

/\* Tipo de Estructura de la ListaJugador \*/

struct ListaJugador{

PtrNodoListaJugador primeroListaJugador; // 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 crearListaJugador(ListaJugador &lista);

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

/\*

pre : lista Creada con crearListaJugador().

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

lista : lista sobre la cual se invoca la primitiva.

\*/

bool listaVaciaJugador(ListaJugador &lista);

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

/\*

pre : lista Creada con crearListaJugador().

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 finListaJugador de la lista.

\*/

PtrNodoListaJugador finListaJugador();

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

/\*

pre : lista Creada con crearListaJugador().

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

esta vacia

lista : lista sobre la cual se invoca la primitiva.

return puntero al primer nodo.

\*/

PtrNodoListaJugador primeroListaJugador(ListaJugador &lista);

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

/\*

pre : lista Creada con crearListaJugador().

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

ptrNodo apuntaba a finListaJugador() 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 siguienteListaJugador.

return puntero al nodo siguienteListaJugador.

\*/

PtrNodoListaJugador siguienteListaJugador(ListaJugador &lista, PtrNodoListaJugador ptrNodo);

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

/\*

pre : lista Creada con crearListaJugador().

ptrNodo es un puntero a un nodo de lista.

post: devuelve el puntero al nodo anteriorListaJugador del apuntado, o devuelve finListaJugador() si

ptrNodo apuntaba al primeroListaJugador 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 anteriorListaJugador.

return puntero al nodo anteriorListaJugador.

\*/

PtrNodoListaJugador anteriorListaJugador(ListaJugador &lista, PtrNodoListaJugador ptrNodo);

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

/\*

pre : lista creada con crearListaJugador().

post: devuelve el puntero al ultimoListaJugador nodo de la lista, o devuelve finListaJugador() si

si lista esta vacia.

lista : lista sobre la cual se invoca la primitiva.

return puntero al último nodo.

\*/

PtrNodoListaJugador ultimoListaJugador(ListaJugador &lista);

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

/\*

pre : lista creada con crearListaJugador().

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.

\*/

PtrNodoListaJugador adicionarPrincipioListaJugador(ListaJugador &lista, Jugador dato);

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

/\*

pre : lista creada con crearListaJugador().

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 finListaJugador() no inserta

nada y devuelve finListaJugador().

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.

\*/

PtrNodoListaJugador adicionarDespuesListaJugador(ListaJugador &lista, Jugador dato, PtrNodoListaJugador ptrNodo);

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

/\*

pre : lista creada con crearListaJugador().

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.

\*/

PtrNodoListaJugador adicionarFinalListaJugador(ListaJugador &lista, Jugador dato);

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

/\*

pre : lista creada con crearListaJugador().

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 finListaJugador(). Si ptrNodo apunta al primeroListaJugador, el nodo

insertado sera el nuevo primeroListaJugador.

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.

\*/

PtrNodoListaJugador adicionarAntesListaJugador(ListaJugador &lista, Jugador dato, PtrNodoListaJugador ptrNodo);

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

/\*

pre : lista creada con crearListaJugador(), no vacia. ptrNodo es distinto de finListaJugador().

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 colocarDatoListaJugador(ListaJugador &lista, Jugador &dato, PtrNodoListaJugador ptrNodo);

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

/\*

pre : lista creada con crearListaJugador(), no vacia. ptrNodo es distinto de finListaJugador().

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 obtenerDatoListaJugador(ListaJugador &lista, Jugador &dato, PtrNodoListaJugador ptrNodo);

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

/\*

pre : lista creada con crearListaJugador().

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

esta vacia o si ptrNodo apunta a finListaJugador().

lista : lista sobre la cual se invoca la primitiva.

ptrNodo : puntero al nodo que se desea eliminar.

\*/

void eliminarNodoListaJugador(ListaJugador &lista, PtrNodoListaJugador ptrNodo);

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

/\*

pre : lista creada con crearListaJugador().

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

accion alguna.

lista : lista sobre la cual se invoca la primitiva.

\*/

void eliminarNodoPrimeroListaJugador(ListaJugador &lista);

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

/\*

pre : lista creada con crearListaJugador().

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

sino no realiza accion.

lista : lista sobre la cual se invoca la primitiva.

\*/

void eliminarNodoUltimoListaJugador(ListaJugador &lista);

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

/\*

pre : lista creada con crearListaJugador().

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 eliminarListaJugador(ListaJugador &lista);

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

/\* Definición de Operaciones Adicionales \*/

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

/\*

pre : lista fue creada con crearListaJugador().

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 finListaJugador().

lista : lista sobre la cual se invoca la primitiva.

dato : elemento a localizar.

return puntero al nodo localizado o finListaJugador().

\*/

PtrNodoListaJugador localizarDatoListaJugador(ListaJugador &lista , Jugador dato);

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

/\*

pre : lista fue creada con crearListaJugador() 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.

\*/

PtrNodoListaJugador insertarDatoListaJugador(ListaJugador &lista, Jugador dato);

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

/\*

pre : la lista fue creada con crearListaJugador().

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 eliminarDatoListaJugador(ListaJugador &lista, Jugador dato);

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

/\*

pre : la lista fue creada con crearListaJugador().

post : reordena la lista.

lista : lista sobre la cual se invoca la primitiva.

\*/

void reordenarListaJugador(ListaJugador &lista);

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

/\*

pre : la lista fue creada con crearListaJugador().

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

lista : lista sobre la cual se invoca la primitiva.

\*/

int longitudListaJugador(ListaJugador &lista);

#endif // LISTAJUGADOR\_H\_INCLUDED