```c

#include <stdlib.h>
#include <stdio.h>
#include <assert.h>
#include <string.h>

#include "Bool.h"
#include "Alumno.h"

struct Node_Type
{
    Alumno alumno;

    struct Node_Type* next;
    struct Node_Type* prev;
};

typedef struct Node_Type Node;

struct DLL_Type
{
    Node* first;
    Node* last;
    Bool empty;
};

typedef struct DLL_Type DLL;

Node* newNode (Alumno* alumno)
{
    Node* n = (Node*) malloc (sizeof (Node));
    if (n) {
        Alumno_Assign (&n->alumno, alumno);
        // n.alumno := alumno;

        n->next = n->prev = NULL;
    }
    return n;
}

void deleteNode (Node* n)
{
    if (n) { free (n); }
}


DLL* DLL_Create ()
{
    DLL* lista = (DLL*) malloc (sizeof (DLL));
    if (lista) {
        lista->first = lista->last = NULL;
        lista->empty = TRUE;
    }
    return lista;
}

void DLL_Destroy (DLL* this)
{
    if (this) {

        // borra todos los nodos existentes
        while (this->first) {
            Node* tmp = this->first->next;
            deleteNode (this->first);
            this->first = tmp;
        }

        free (this);
        // borra la lista
    }
```

```
 71  }
 72
 73  Bool DLL_Insert (DLL* this, Alumno* alumno)
 74  {
 75      Node* n = newNode (alumno);
 76
 77      if (!n) { return FALSE; }
 78
 79      if (this->empty) {
 80          this->first = this->last = n;
 81          this->empty = FALSE;
 82      }
 83      else {
 84          this->last->next = n;
 85          n->prev = this->last;
 86          this->last = n;
 87      }
 88
 89      return TRUE;
 90  }
 91
 92  Bool DLL_Remove (DLL* this, Alumno* alumno)
 93  {
 94    if (this->first != this->last) {
 95  #if 0
 96      // No borré el siguiente código para que vean la diferencia entre usar
 97      // código en bruto para hacer una asignación, y código funcionalmente
 98      // abstracto (es decir, en una función):
 99
100      char* nombre = Alumno_GetNombre (&this->first->alumno);
101      int semestre = Alumno_GetSemestre (&this->first->alumno);
102      float promedio = Alumno_GetPromedio (&this->first->alumno);
103
104      Alumno_SetNombre (alumno, nombre);
105      Alumno_SetSemestre (alumno, semestre);
106      Alumno_SetPromedio (alumno, promedio);
107  #endif
108      Alumno_Assign (alumno, &this->first->alumno);
109      // alumno := this.first.alumno
110
111
112      Node* tmp = this->first->next;
113      deleteNode (this->first);
114      this->first = tmp;
115      this->first->prev = NULL;
116      return TRUE;
117    }
118    else if (this->empty == FALSE) {
119  #if 0
120      char* nombre = Alumno_GetNombre (&this->first->alumno);
121      int semestre = Alumno_GetSemestre (&this->first->alumno);
122      float promedio = Alumno_GetPromedio (&this->first->alumno);
123
124      Alumno_SetNombre (alumno, nombre);
125      Alumno_SetSemestre (alumno, semestre);
126      Alumno_SetPromedio (alumno, promedio);
127  #endif
128
129      Alumno_Assign (alumno, &this->first->alumno);
130      // alumno := this.first.alumno
131
132      deleteNode (this->first);
133      this->first = this->last = NULL;
134      this->empty = TRUE;
135      return TRUE;
136    }
137    else { return FALSE; }
138
139  }
140
```

```
141  Alumno DLL_Remove2 (DLL* this)
142  {
143    Alumno tmp;
144
145    // ...
146
147    return tmp;
148  }
149
150  //-----------------------------------------------------------------------
151  //  Driver program
152  //-----------------------------------------------------------------------
153  int main(void)
154  {
155      DLL *miLista = DLL_Create ();
156      assert (miLista);
157
158      Alumno unAlumno;
159
160      Alumno_SetNombre (&unAlumno, "FCO");
161      Alumno_SetSemestre (&unAlumno, 2);
162      Alumno_SetPromedio (&unAlumno, 10.0);
163      Alumno_Print (&unAlumno);
164
165      DLL_Insert (miLista, &unAlumno);
166
167      Alumno_SetNombre (&unAlumno, "");
168      Alumno_SetSemestre (&unAlumno, 1);
169      Alumno_SetPromedio (&unAlumno, 5.0);
170      Alumno_Print (&unAlumno);
171
172      DLL_Remove (miLista, &unAlumno);
173      Alumno_Print (&unAlumno);
174
175      DLL_Destroy (miLista);
176
177      return 0;
178  }
179
180
```