```c

#include <stdio.h>
#include <stdbool.h>
#include "lectura.h"
#include "listaClase.h"



/*****************************************************************************
 Función: menu
    Escribe por pantalla el menú de la aplicación y recoge la opción elegida.
    Parámetro de salida (valor devuelto por la función): entero con la opción elegida.
 *****************************************************************************/
int menu();

/*****************************************************************************
 Función: recuperarInformacion
    Recupera la información contenida en el fichero binario cuyo nombre se pasa como
    argumento de main y se guarda en la lista apuntada por pl.
    Parámetro de entrada: nomFichero. Cadena de acaracteres con el nombre del fichero.
    Precondiciones: el nombre del fichero se pasa como argumento de main.
    Parámetro de entrada/salida: pl. Puntero a una estructura de tipo tListaClase con
        la lista inicializada como lista vacía, en la que se guarda la información
        contenida en el fichero.
    Parámetro de salida (valor devuelto por la función): booleano, true si hahabido error
                    al abrir el fichero o al escribir y false en caso contrario.
 *****************************************************************************/
 bool recuperarInformacion (char *nomFichero, tListaClase *pl);

 /*****************************************************************************
 Función: guardarInformacion
    Guarda la información de los alumnos de la lista l, en el fichero binario cuyo nombre
    se pasa como argumento de main.
    Parámetro de entrada:
        nomFichero. Cadena de caracteres con el nombre del fichero.
        l:estructura de tipo tListaClase con la lista de alumnos de clase.
    Precondiciones: el nombre del fichero se pasa como argumento de main.
                    l tiene que estar inicializada.
    Parámetro de salida (valor devuelto por la función): booleano, true si hahabido error
                    al abrir el fichero o al escribir y false en caso contrario.
 *****************************************************************************/
 bool guardarInformacion (char *nomFichero, tListaClase l);

int main (int argc, char *argv[]){
    tListaClase lista;
    int opcion;
    int pos;
    tNombre alumno;

    if (argc != 2)
        printf("Error. Se debe pasar como argumento el nombre del fichero en el se guarda/recupera la
informacion\n");
    else{
        inicializarLista(&lista);
        if(recuperarInformacion (argv[1], &lista))
            printf("No se ha podido recuperar la informacion del fichero %s\n", argv[1]);
        do{
            opcion= menu();
            switch (opcion){
                case 1: printf("\nEsta usted en la OPCION 1: Rellenar lista de clase\n");
                        printf ("\nSe han introducido un total de %d alumnos en la lista", rellenarLista(&
lista));
                        break;
                case 2: printf("\nEsta usted en la OPCION 2. Escribir lista de clase\n");
                        escribirLista(lista);
                        break;
```

```
65                 case 3: printf("\nEsta usted en la OPCION 3: Buscar alumno de clase\n");
66                         printf("\nAlumno a buscar: ");
67                         /*fgets(alumno, MAX_NOM, stdin);*/
68                         /*fscanf(stdin, "%s", alumno);*/
69                         leerCadena(alumno, MAX_NOM);
70                         if (buscarAlumnoEnLista(alumno, lista, &pos))
71                             printf("\nEl alumno %s esta en la posicion %d de la lista", alumno, pos);
72                         else
73                             printf("\nEl alumno %s no esta en la lista", alumno);
74                         break;
75                 case 4: printf("\nEsta usted en la OPCION 4: Modificar alumno de clase\n");
76                         printf("\nAlumno a buscar: ");
77                         /*fgets(alumno, MAX_NOM, stdin);*/
78                         /*fscanf(stdin, "%s", alumno);*/
79                         leerCadena(alumno, MAX_NOM);
80                         if (buscarAlumnoEnLista(alumno, lista, &pos)){
81                             if(modificarAlumnoEnLista(&lista, pos))
82                                printf("\nSe ha modificado el alumno numero %d de la lista", pos);
83                         }
84                         else
85                             printf("\nEl alumno %s no esta en la lista", alumno);
86                         break;
87                 case 5: printf("\nEsta usted en la OPCION 5: Borrar alumno de clase\n");
88                         printf("\nAlumno a buscar: ");
89                         /*fgets(alumno, MAX_NOM, stdin);*/
90                         /*fscanf(stdin, "%s", alumno);*/
91                         leerCadena(alumno, MAX_NOM);
92                         if (buscarAlumnoEnLista(alumno, lista, &pos)){
93                             if(borrarAlumnoDeLista(&lista, pos))
94                                printf("\nSe ha borrado el alumno %s de la lista", alumno);
95                         }
96                         else
97                             printf("\nEl alumno %s no esta en la lista", alumno);
98                         break;
99                 case 6: printf ("\nAdios. Gracias por utilizar este programa\n");
100                        if(guardarInformacion (argv[1], lista))
101                            printf("No se ha podido guardar la informacion en el fichero %s\n", argv[1]);
102                        break;
103                default: printf ("\nNo es una opcion correcta. Por favor, introduzca una opcion valida\n");
104            }
105        }while (opcion!=6);
106    }
107
108    return 0;
109 }
110
111 int menu(){
112    int op;
113        printf ("\n\n  **********Lista de clse**********");
114        printf ("\n  *   1. Rellenar lista de clase     *");
115        printf ("\n  *   2. Escribir lista de clase     *");
116        printf ("\n  *   3. Buscar alumno de clase      *");
117        printf ("\n  *   4. Modificar alumno de clase   *");
118        printf ("\n  *   5. Borrar alumno de clase      *");
119        printf ("\n  *   6. Salir                       *");
120        printf ("\n  ***********************************");
121
122
123        printf("\n\nTeclee opcion: ");
124        scanf("%d", &op);
125        fflush(stdin);
126    return op;
127 }
128
129 bool recuperarInformacion (char *nomFichero, tListaClase *pl){
130    FILE *pfListaClase;
```

```
131     bool error= false;
132
133     if ((pfListaClase= fopen (nomFichero, "rb")) != NULL) {  // Apertura del fichero
134        fread (&(pl->lista[pl->elementosLista]), sizeof(tAlumno), 1, pfListaClase);
135        while ((!feof(pfListaClase)) && (!ferror(pfListaClase))) {
136           pl->elementosLista++;
137           fread (&(pl->lista[pl->elementosLista]), sizeof(tAlumno), 1, pfListaClase);
138        }
139        if (ferror(pfListaClase)) // Ha habido error en la lectura del fichero
140            error= true;
141        fclose (pfListaClase);
142      }
143      else //Ha habido error en la apertura de fichero
144         error= true;
145
146      return error;
147  }
148
149  bool guardarInformacion (char *nomFichero, tListaClase l){
150     FILE *pfListaClase;
151     bool error= false;
152     int contador= 0;
153
154     if ((pfListaClase= fopen (nomFichero, "wb")) != NULL){
155         while ((!ferror(pfListaClase)) && (contador < l.elementosLista)) {
156             fwrite(&(l.lista[contador]), sizeof(tAlumno), 1, pfListaClase);
157             contador ++;
158         }
159         //fwrite(&(l.lista[contador]), sizeof(tAlumno), (unsigned)l.elementosLista, pfListaClase);
160         if (ferror(pfListaClase)) // Ha habido error en la escritura del fichero
161            error= true;
162         fclose (pfListaClase);
163      }
164      else //Ha habido error en la apertura de fichero
165         error= true;
166
167     return error;
168  }
169
```