```c

 #include <stdio.h>
 #include <stdbool.h>
 #include <string.h>
 #include "lectura.h"
 #include "listaAsignaturas.h"

 #define MAX_RES 3

 /* Definición de prototipos de funciones internas*/

 /*****************************************************************************
 Función: haySitioEnListaAsignaturas
    Comprueba si hay sitio en la lista.
    Parámetro de entrada: la, array con los nombres de las asignaturas de un alumno.
    Precondiciones:  la lista la tiene que estar inicializada.
    Parámetro de salida: psitio, pasado por referencia, puntero a entero con la posición
    del primer hueco libre en la.
    Parámetro de salida (valor devuelto por la función): booleano, false si no hay sitio,
        true si hay sitio.
 *****************************************************************************/
bool haySitioEnlistaAsignaturas (const tListaAsignaturas la, int *psitio);

void inicializarListaAsignaturas (tListaAsignaturas la){
    for (int i=0; i<MAX_LISTA_ASIG; i++)
            la[i].ocupado= false;
}

int rellenarListaAsignaturas (tListaAsignaturas la){
    char respuesta[MAX_RES];
    tNombre asignatura;
    int sitio;
    int contador;
    int pos;

    contador= 0;
    printf("Desea matricular una asignatura? ");
    leerCadena(respuesta, MAX_RES);
    while (!strcmp(respuesta,"si") && haySitioEnlistaAsignaturas(la, &sitio)){
        printf("Asignatura: ");
        leerCadena(asignatura, MAX_NOM);
        if (buscarAsignaturaEnLista(asignatura, la, &pos))
            printf("\nLa asignatura ya ha sido matriculada\n");
        else{
            strcpy(la[sitio].nombreAsignatura, asignatura);
            la[sitio].ocupado= true;
            contador++;
            printf("Desea matricular otra asignatura? ");
            leerCadena(respuesta, MAX_RES);
        }
    }
    if (!haySitioEnlistaAsignaturas(la, &sitio))
        printf("\nLo sentimos, la lista de asignaturas esta llena\n");
    return contador;
}

void escribirListaAsignaturas (const tListaAsignaturas la){
    for (int i=0; i<MAX_LISTA_ASIG; i++){
            if(la[i].ocupado)
                printf("%s\t", la[i].nombreAsignatura);
    }
}

bool buscarAsignaturaEnLista (const tNombre a, tListaAsignaturas la, int *ppos){
    int i;
    bool enc;
```

```
67
68      i= 0;
69      enc= false;
70      while (!enc && i<MAX_LISTA_ASIG){
71          if(la[i].ocupado){
72              if (!strcmp(a, la[i].nombreAsignatura))
73                  enc= true;
74              else
75                  i++;
76          }
77          else
78              i++;
79      }
80      *ppos= i;
81
82      return enc;
83  }
84
85
86  bool borrarAsignaturaDeLista (tListaAsignaturas la, int pos){
87      char respuesta[MAX_RES];
88      bool borrado;
89
90      printf("Desea borrar la asignatura? ");
91      leerCadena(respuesta, MAX_RES);
92      if (!strcmp(respuesta,"si")){
93          la[pos].ocupado= false;
94          borrado= true;
95      }
96      else
97          borrado= false;
98
99      return borrado;
100 }
101
102 bool haySitioEnlistaAsignaturas (const tListaAsignaturas la, int *psitio){
103     bool haySitio;
104     int i;
105
106     haySitio= false;
107     i= 0;
108     while (i<MAX_LISTA_ASIG && !haySitio){
109         if (!la[i].ocupado)
110             haySitio= true;
111         else
112             i++;
113     }
114     *psitio= i;
115
116     return haySitio;
117 }
118
119
```