```c
/*******************************************************************************
* Author:       Rachel Jacquay
* Course:       EGR 226-902
* Date:         01/27/2021
* Project:      Lab 2 Part 2
* File:         main_part2.c
* Description:  This program takes in a file of type .csv, and also user input to
                create a search engine of books, using their title, author, ISBN,
                page number, and year published. It loops until the user tells it to
                stop by setting the variable 'loop' equal to 0. Error checking was
                used for all inputs by the user as well as the file.
*******************************************************************************/

#include <stdio.h>      // preprocessor directives
#include <stdlib.h>
#include <math.h>
#include <ctype.h>
#include <string.h>

#define MAX 500         // macro

typedef struct {                // creating struct
        char title[225];
        char author_name[50];
        char ISBN[10];
        int pages;
        int year_published;
    } book;

int parse_file(char filename[], book book_array[]);             // function prototypes
void print_book(book my_book);
void search_title(book book_title[], int n, char title[]);
void search_author(book book_author[], int n, char author_name[]);
void search_ISBN(book book_ISBN[], int n, char ISBN[]);

int main() {                // main function
    book my_book;           // declaring variables
    book book_array[360];
    char filename[MAX];
    char userin[255];
    int b, num;
    int status = 1;
    int loop = 1;

    strcpy(filename, "BookList.csv");       // set filename to BookList.csv

    b = parse_file(filename, book_array);       // set i equal to the number of books read in

    do {                                                    // do all of this while the user still
wants to loop
        printf("Which process would you like to search by?\n");     // prompt user
        printf("Please enter one of the following numbers:\n");
        printf("[0] Search by Title\n");
        printf("[1] Search by Author\n");
        printf("[2] Search by ISBN\n");
        printf("[3] Exit code\n\n");

        do {                                // do all of this while the scanf and numbers scanned in are
valid
            fflush(stdin);
            status = scanf("%d", &num);     // checking for scanf to be valid

            if (status == 0 || num < 0 || num > 3) {        // if invalid, redo
                printf("Incorrect value\n");
                printf("Please enter a '0' or '1' or '2' or '3'\n");
                fflush(stdin);
```

```c
            }
        } while (status == 0 || num < 0 || num > 3);

        if (num == 0) {                     // search by title
            printf("\nUser entered 0\n");

                fflush(stdin);
                printf("\nPlease enter a case-sensitive title\n\n");
                scanf("%[^\n]s", userin);                       // get the string
                search_title(book_array, b, userin);        // call search by title function
        }

        else if (num == 1) {                // search by author
            printf("\nUser entered 1\n");

                fflush(stdin);
                printf("\nPlease enter a case-sensitive author\n\n");
                scanf("%[^\n]s", userin);                       // get the string
                search_author(book_array, b, userin);       // call search by author function
        }

        else if (num == 2) {                // search by ISBN
            printf("\nUser entered 2\n");

                fflush(stdin);
                printf("\nPlease enter an ISBN\n\n");
                scanf("%[^\n]s", userin);                       // get the string
                search_ISBN(book_array, b, userin);         // call search by ISBN function
        }

        else if (num == 3) {                // end the program entirely
            printf("\nUser entered 3\n");
            printf("Goodbye!");
            loop = 0;
        }

        printf("\n");

    } while (loop != 0);    // loop until user wants to end program

    return 0;
}

/*  parse_file
    Description: This function opens, stores, and closes the file, while
    also parsing its contents into words and using those for the search
    engine. It returns the total number of books found.
    Inputs: filename[], book_array[]
    Outputs: i
*/
int parse_file(char filename[], book book_array[]) {
    char buffer[512];                                   // Create temporary string buffer variable
    int i = 0;
    char* ptr;
    FILE *infile;                                       // Attempt to open file

    infile = fopen(filename, "r");

    if (infile == NULL)                                 // Return 0 (failure) if file could not open
        return -1;

    while (fgets(buffer, 512, infile)) {                // Loop collecting each line from the file
        ptr = strtok(buffer, ",");                      // Parse string by commas and newline
            strcpy(book_array[i].title, ptr);           // First parse is title

        ptr = strtok(NULL, ",\n");                      // repeat for author name
```

```
131             strcpy(book_array[i].author_name, ptr);
132
133         ptr = strtok(NULL, ",\n");                          // repeat for ISBN
134             strcpy(book_array[i].ISBN, ptr);
135
136         ptr = strtok(NULL, ",\n");                          // repeat for pages
137         if (strcmp(ptr, "N/A"))                             // if N/A, set ptr to an int and store the
value in ptr
138             book_array[i].pages = atoi(ptr);
139         else if (strcmp(ptr, "N/A") == 0)                   // if it is 0 to start, output 0
140             book_array[i].pages = 0;
141
142         ptr = strtok(NULL,",\n");                          // repeat for year published
143         if (strcmp(ptr,"N/A"))                             // same as pages
144             book_array[i].year_published = atoi(ptr);
145         else if (strcmp(ptr, "N/A") == 0)
146             book_array[i].year_published = 0;
147
148         i++;        // increment i to see how many books are found
149     }
150
151     fclose(infile);     // close file
152
153     return i;       // return how many books are found total
154 }
155
156 /*  print_book
157     Description: This function prints out the book info whenever it is called
158     in any of other functions.
159     Inputs: my_book
160     Outputs: none
161 */
162 void print_book(book my_book) {                         // printing book function
163     printf("\nTitle:       %s\n", my_book.title);       // print title
164     printf("Author:      %s\n", my_book.author_name);   // print author name
165     printf("ISBN:        %s\n", my_book.ISBN);          // print ISBN
166
167     if (my_book.pages == 0)                             // if struct is 0
168         printf("Pages:       N/A\n");                   // print N/A
169
170     else if (my_book.pages != 0)                        // if struct is not 0
171         printf("Pages:       %d\n", my_book.pages);     // print value
172
173     if (my_book.year_published == 0)                    //if struct is 0
174         printf("Year Published:       N/A\n");          // print N/A
175
176     else if (my_book.year_published != 0)                               // if struct is not 0
177         printf("Year Published:       %d\n", my_book.year_published);   // print value
178 }
179
180 /*  search_title
181     Description: This function determines if the title given by the user
182     matches any title in the .csv file.
183     Inputs: book_title, n, title
184     Outputs: none
185 */
186 void search_title(book book_title[], int n, char title[]) {     // search title function definition
187     int i;
188     int var;
189     char outcome;
190
191     for (i = 0; i <= n; i++) {                              // until index equals n number of books
192         outcome = (strstr(book_title[i].title, title));     // set outcome equal to strstr of book_title[i]
and title
193
194         if (outcome) {                                      // if outcome == 1
```

```
195             print_book(book_title[i]);                      // print title
196             var++;                                          // var tells user how many books were found
197             }
198         }
199
200     if (var == 0) {                         // if var == 0
201         printf("\nNo results found\n");     // no books found
202     }
203 }
204
205 /*  search_author
206     Description: This function determines if the author name given by the user
207     matches any author names in the .csv file.
208     Inputs: book_author, n, author_name
209     Outputs: none
210 */
211 void search_author(book book_author[], int n, char author_name[]) {     // search author function
definitions
212     int i;
213     int var;
214     char outcome;
215
216     for (i = 0; i <= n; i++) {                                          // until index is equal to n number
of books
217         outcome = (strstr(book_author[i].author_name, author_name));    // set outcome equal to strstr of
book_author[i] and author
218
219         if (outcome) {                      // if outcome == 1
220             print_book(book_author[i]);     // print author
221             var++;                          // var tells user how many books were found
222         }
223     }
224
225     if (var == 0) {                         // if var == 0
226         printf("\nNo results found\n");     // no books found
227     }
228 }
229
230 /*  search_ISBN
231     Description: This function determines if the ISBN given by the user
232     matches any ISBN in the .csv file.
233     Inputs: book_ISBN, n, ISBN
234     Outputs: none
235 */
236 void search_ISBN(book book_ISBN[], int n, char ISBN[]) {        // search ISBN function definition
237     int i;
238     int var;
239     char outcome;
240
241     for (i = 0; i <= n; i++) {                          // until index is equal to n number of books
242         outcome = (strstr(book_ISBN[i].ISBN, ISBN));    // set outcome equal to strstr of book_ISBN[i] and
ISBN
243
244         if (outcome) {                  // if outcome == 1
245             print_book(book_ISBN[i]);   // print ISBN
246             var++;                      // var tells user how many books were found
247         }
248     }
249
250     if (var == 0) {                         // if var == 0
251         printf("\nNo results found\n");     // no books found
252     }
253 }
```