EGR 226: Microcontroller Programming and Applications

Winter 2021

Instructor: Prof. Trevor Ekin

Lab 2: C-Programming Refesher 2

Jake Carlson

February 3, 2021

Contents

1. Objectives

The objective of this lab was to go more in depth into c programming with another refresher. In this lab a new program was created from building off of the previous resistor color-band calculator which now also calculates the resistance value after a user inputs each color of the four bands. In the second part of this lab a new program needed to use structures to make a searchable database of books using a .csv file.

2. Introduction

2.1. Part 1: Resistance Value Calculator

In the first part of Lab 2 a new program was created from part 1 of Lab 1 and adding on to it. New functions were needed in order to do the opposite of the previous program which is calculating resistance values from the users input of colors for each color band. These two programs need to work in unison in order for the user to be able to choose if they want to convert from resistance value to color bands, or color bands to resistance value.

2.2. Part 2: Book Database

In the second part of Lab 2 a program was created that searches an entire database of books using structures. This program allows a user to input a title, author, or ISBN, as search criteria in order to locate the correct book.

3. Procedure

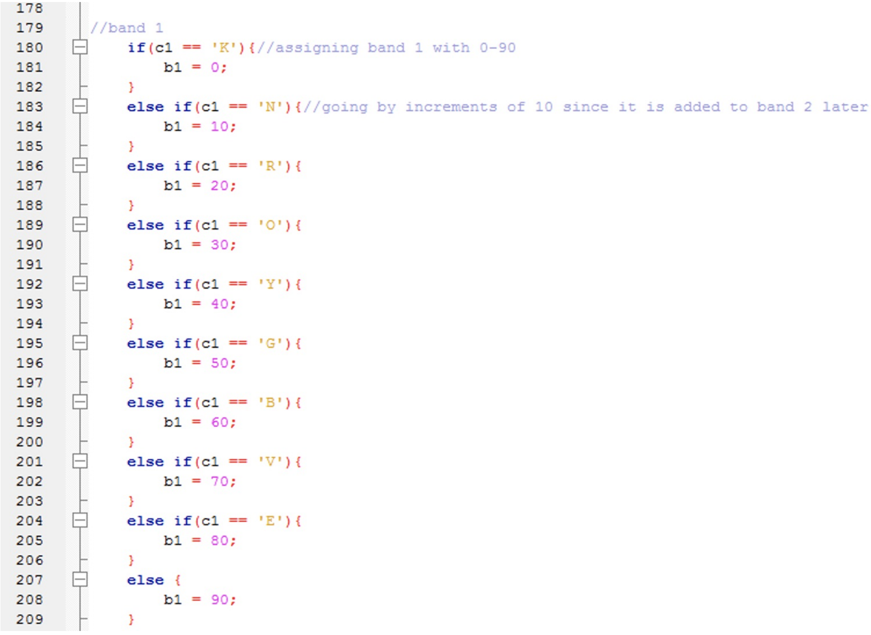
3.1. Part 1: Resistance Value calculator

3.1.1. Discussing the Steps

This program was created by using part 1 from the previous lab and adding new functions such as getColorBands, and calcResistance. The function getColorbands collects the users input of 4 colors for each of the 4 bands. The function calcResistance is used to assign each band with the correct resistance value associated with the color. This function then prints the resistance value and tolerance. Finally, this program needs to work with the first program, so the original prompt had to be altered for the user to decide which calculator is necessary.

3.1.2. Band 1 Assignment in calcReistance

This was a more challenging part of the program because the first 2 bands need to be added together in order to hold the correct value. When simply adding the 2 bands together the incorrect value will be calculated. To find the correct value band 1 needs to increment by 10 for each color since Brown + Brown should be 11. If both brown bands were assigned to 1 then the value would be 2 which is incorrect. an example is shown below.



```

//band 1
    if(c1 == 'K'){//assigning band 1 with 0-90
        b1 = 0;
    }
    else if(c1 == 'N'){//going by increments of 10 since it is added to band 2 later
        b1 = 10;
    }
    else if(c1 == 'R'){
        b1 = 20;
    }
    else if(c1 == 'O'){
        b1 = 30;
    }
    else if(c1 == 'Y'){
        b1 = 40;
    }
    else if(c1 == 'G'){
        b1 = 50;
    }
    else if(c1 == 'B'){
        b1 = 60;
    }
    else if(c1 == 'V'){
        b1 = 70;
    }
    else if(c1 == 'E'){
        b1 = 80;
    }
    else {
        b1 = 90;
    }
```

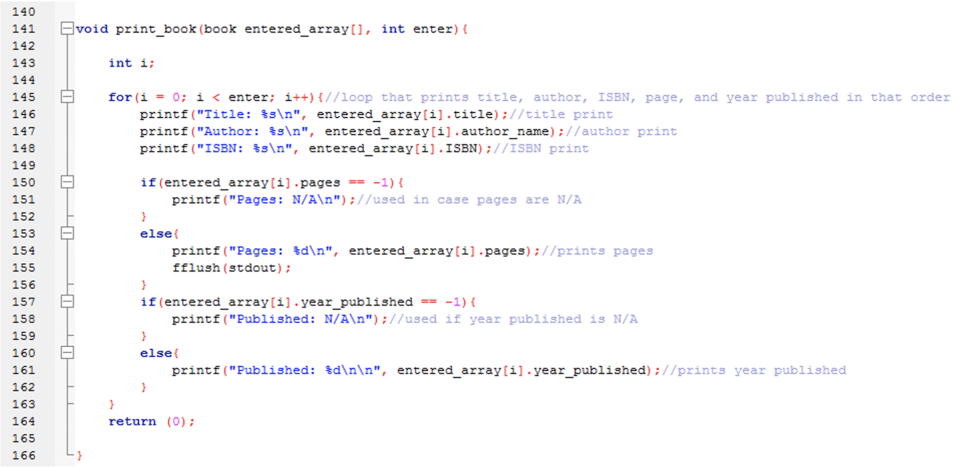
3.2. Part 2: Book Database

3.2.1. Discussing the Steps

This program was created by defining a book structure that includes the title, author, ISBN, pages, and year published. Then a parse file function was created in order to open the file and scan in the book information into the book array. After this a print book function was created to print the book information in a clean manor so the user can read the output. Finally, three functions for searching the title, author, and ISBN were created so that the user can choose what to search by and these functions will search the database. After searching these functions pass back the correct strings to be printed.

3.2.2. Print Book Function

This is a challenging function because not only does the book need to be printed neatly, but it also needs to print the correct book that the user is looking for. Another challenge was that some information in the book database is not given such as pages or the year published, so there needs to be a solution for that issue as well. Print book function can be seen below.



```c
void print_book(book entered_array[], int enter){

    int i;

    for(i = 0; i < enter; i++){//loop that prints title, author, ISBN, page, and year published in that order
        printf("Title: %s\n", entered_array[i].title);//title print
        printf("Author: %s\n", entered_array[i].author_name);//author print
        printf("ISBN: %s\n", entered_array[i].ISBN);//ISBN print

        if(entered_array[i].pages == -1){
            printf("Pages: N/A\n");//used in case pages are N/A
        }
        else{
            printf("Pages: %d\n", entered_array[i].pages);//prints pages
            fflush(stdout);
        }
        if(entered_array[i].year_published == -1){
            printf("Published: N/A\n");//used if year published is N/A
        }
        else{
            printf("Published: %d\n\n", entered_array[i].year_published);//prints year published
        }
    }
    return (0);

}
```

4. Results/Discussion

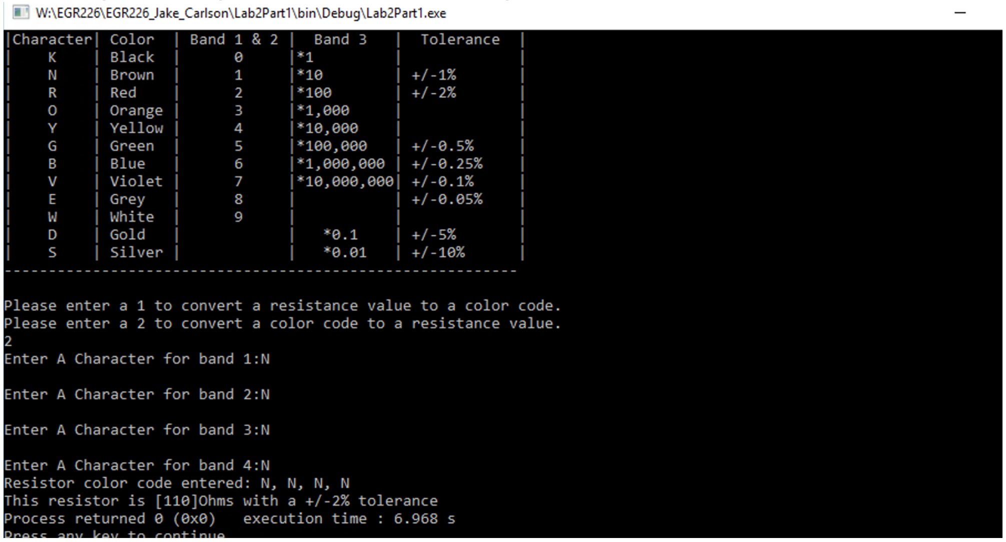
4.1. Conclusion

In conclusion there were multiple challenges in this c programming refresher lab. Many of them were overcome by spending extra time to think through the process. One of the challenges was searching through the book data base, but an example was given in the lab manual to help overcome the challenge. Overall this lab was another great refresher and will ensure that each student is ready to continue.

4.2. Example Results

4.3. Part 1 example

Running the program in Part 1 entering Brown, Brown, Brown, Brown.

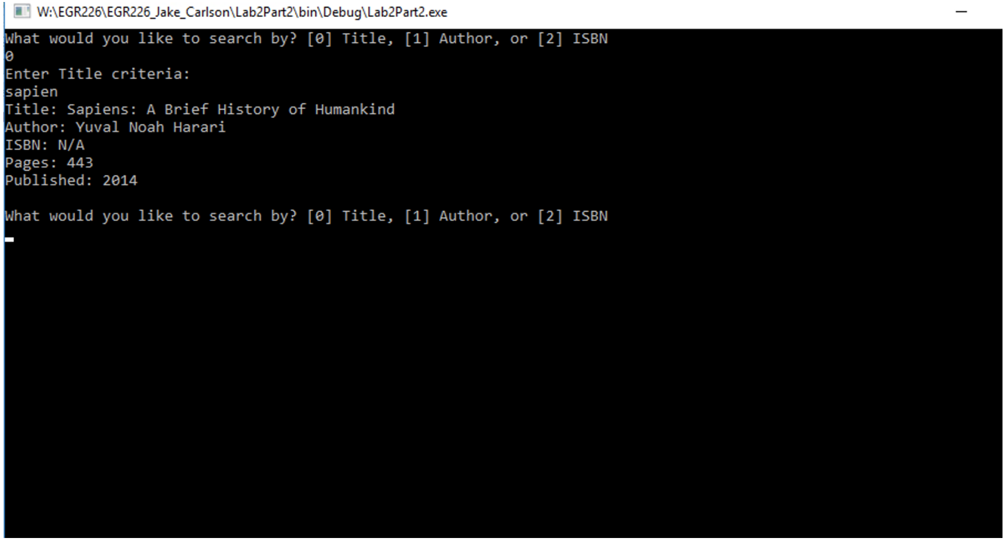


W:\EGR226\EGR226_Jake_Carlson\Lab2Part1\bin\Debug\Lab2Part1.exe

```
|Character| Color  | Band 1 & 2 |  Band 3    |  Tolerance  |
|    K    | Black  |      0     |*1          |             |
|    N    | Brown  |      1     |*10         | +/-1%       |
|    R    | Red    |      2     |*100        | +/-2%       |
|    O    | Orange |      3     |*1,000      |             |
|    Y    | Yellow |      4     |*10,000     |             |
|    G    | Green  |      5     |*100,000    | +/-0.5%     |
|    B    | Blue   |      6     |*1,000,000  | +/-0.25%    |
|    V    | Violet |      7     |*10,000,000 | +/-0.1%     |
|    E    | Grey   |      8     |            | +/-0.05%    |
|    W    | White  |      9     |            |             |
|    D    | Gold   |            |    *0.1    | +/-5%       |
|    S    | Silver |            |    *0.01   | +/-10%      |
----------------------------------------------------------

Please enter a 1 to convert a resistance value to a color code.
Please enter a 2 to convert a color code to a resistance value.
2
Enter A Character for band 1:N

Enter A Character for band 2:N

Enter A Character for band 3:N

Enter A Character for band 4:N
Resistor color code entered: N, N, N, N
This resistor is [110]Ohms with a +/-2% tolerance
Process returned 0 (0x0)   execution time : 6.968 s
Press any key to continue.
```

4.4. Part 2 Title example

Running the program in Part 2 entering the title "sapien".

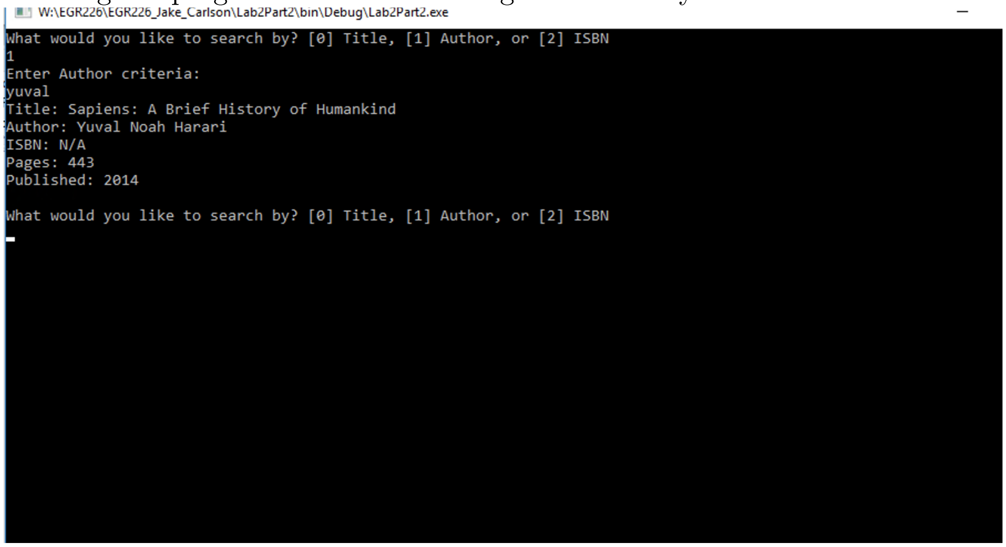


W:\EGR226\EGR226_Jake_Carlson\Lab2Part2\bin\Debug\Lab2Part2.exe

```
What would you like to search by? [0] Title, [1] Author, or [2] ISBN
0
Enter Title criteria:
sapien
Title: Sapiens: A Brief History of Humankind
Author: Yuval Noah Harari
ISBN: N/A
Pages: 443
Published: 2014

What would you like to search by? [0] Title, [1] Author, or [2] ISBN
```

4.5. Part 2 Author example

Running the program in Part 2 entering the author "yuval".

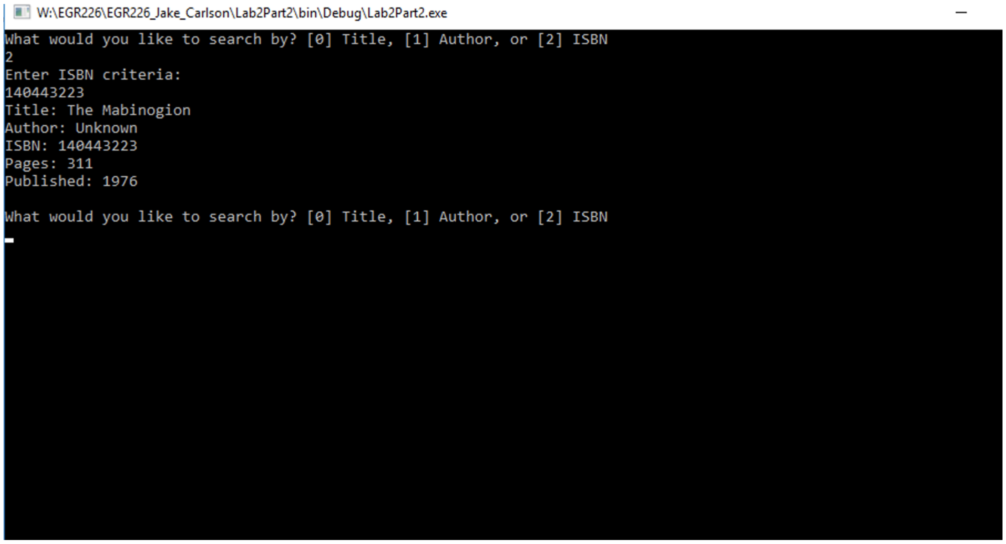


W:\EGR226\EGR226_Jake_Carlson\Lab2Part2\bin\Debug\Lab2Part2.exe

```
What would you like to search by? [0] Title, [1] Author, or [2] ISBN
1
Enter Author criteria:
yuval
Title: Sapiens: A Brief History of Humankind
Author: Yuval Noah Harari
ISBN: N/A
Pages: 443
Published: 2014

What would you like to search by? [0] Title, [1] Author, or [2] ISBN
```

4.6. Part 2 ISBN example

Running the program in Part 2 entering the ISBN "140443223".



W:\EGR226\EGR226_Jake_Carlson\Lab2Part2\bin\Debug\Lab2Part2.exe

```
What would you like to search by? [0] Title, [1] Author, or [2] ISBN
2
Enter ISBN criteria:
140443223
Title: The Mabinogion
Author: Unknown
ISBN: 140443223
Pages: 311
Published: 1976

What would you like to search by? [0] Title, [1] Author, or [2] ISBN
```

Appendices

A. Source Code: Lab 2 Part 1 main.c

```c
/* ****************************************************************************************
* Author: Jake Carlson
* Course: EGR 226 - 902
* Date: 02/1/2021
* Project: lab21part1
* File: lab2part1main.c
* Description: This program takes an input from the user in ohms and solves to find the
* resistors color-bands or takes color-bands and converts to resistance.
*
****************************************************************************************  */


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

void getColorBands(char*, char*, char*, char*);
void calcResistance(char, char, char, char);
void prompt(void);//Declaring function protoypes
int calcResistorColors(int x);
int getIntBetween ();
int getinput();

int main(){

int enteredVal;
int userChoice;
int goodEnter = 0;
char c1, c2, c3, c4;


prompt();
do{
    printf("Please enter a 1 to convert a resistance value to a color code.\n");// new user prompt
    printf("Please enter a 2 to convert a color code to a resistance value.\n");
    scanf("%d", &userChoice);

    if(userChoice == 1){//if user chooses 1 then call functions from previous lab
        enteredVal = getIntBetween(); //function calls
        calcResistorColors(enteredVal);
        goodEnter = 1;
    }
    else if(userChoice == 2){//if 2 call new functions and new prompt
    printf("Enter A Character for band 1:");
    scanf(" %c", &c1);
    printf("\nEnter A Character for band 2:");
    scanf(" %c", &c2);
    printf("\nEnter A Character for band 3:");
    scanf(" %c", &c3);
    printf("\nEnter A Character for band 4:");
    scanf(" %c", &c4);

        getColorBands(c1, c2, c3, c4);
        calcResistance(c1, c2, c3, c4);
        goodEnter = 1;
    }

    else{
        printf("Error\n");//error check
        goodEnter = 0;
    }
} while(goodEnter != 1);

 return (0);}


void prompt(void){ //printf statements for prompt

 printf("----------------Resistor Codes-------------------------------\n");
 printf("|Character| Color  | Band 1 & 2 |  Band 3   |  Tolerance  |\n");
 printf("|    K    | Black  |     0      |*1         | +/-1%%       |\n");
 printf("|    N    | Brown  |     1      |*10        | +/-2%%       |\n");
 printf("|    R    | Red    |     2      |*100       |             |\n");
 printf("|    O    | Orange |     3      |*1,000     |             |\n");
 printf("|    Y    | Yellow |     4      |*10,000    |             |\n");
 printf("|    G    | Green  |     5      |*100,000   | +/-0.5%%     |\n");
 printf("|    B    | Blue   |     6      |*1,000,000 | +/-0.25%%    |\n");
 printf("|    V    | Violet |     7      |*10,000,000| +/-0.1%%     |\n");
 printf("|    E    | Grey   |     8      |           | +/-0.05%%    |\n");
 printf("|    W    | White  |     9      |           |             |\n");
 printf("|    D    | Gold   |            |   *0.1    | +/-5%%       |\n");
 printf("|    S    | Silver |            |   *0.01   | +/-10%%      |\n");
 printf("-------------------------------------------------------------\n\n");

}
int getIntBetween (int val){

    printf("Please enter a value from 1 to 99M ohms\n");
    val = 0; //Value entered
    char wrong[100]; //storage for error when using alphabet
    int pro, i=0;

while(i == 0){ //while loop to make sure value is between 1 and 99

    pro= scanf("%d", &val);

    if (pro == 1){
    if (val > 0 && val <= 99000000){//checking value
        i += 1;
    }
    else {
        printf("Please enter a value from 1 to 99M ohms\n");
    }
    }
    else{
        scanf("%s", &wrong);//saving the incorrect characters to wrong
        printf("Please enter a value from 1 to 99M ohms\n");
    }
}
return (val);

}



int calcResistorColors(int enteredVal){

    char colors [10][10] = {
    "Black", "Brown", "Red", "Orange", "Yellow", "Green", "Blue", "Violet", "Grey", "White" //array for colors of resistor bands
};

    int b1, b2, b3; //declaring band 1 band 2 and band 3;
    int counter = 0;


 printf("A resistor of %d Ohms would have a color code of: ", enteredVal);

        do{
            enteredVal = enteredVal/10;
            counter = counter +1;
}
    while (enteredVal >= 100);//do while loop to divide the resistor value by 10 and increment a counter variable

    if (counter == 0){//assigning band 3 with 0-7
        b3 =0;
 }
    else if (counter ==1){
        b3 =1;
 }
    else if (counter ==2){
        b3 =2;
 }
    else if (counter ==3){
        b3 =3;
 }
    else if (counter ==4){
        b3 =4;
 }
    else if (counter ==5){
        b3 =5;
 }
    else if (counter ==6){
        b3 =6;
 }
    else if (counter ==7){
        b3 =7;
 }
            b1 = enteredVal / 10;
            b2 = enteredVal % 10;

printf(" %s-%s-%s", colors[b1], colors[b2], colors[b3]); //final printf


 return (enteredVal);}

 void getColorBands(char* c1, char* c2, char* c3, char* c4){//new function that passes with pointers
   char color[4] = {c1, c2, c3, c4};

   printf("Resistor color code entered: %c, %c, %c, %c\n", c1, c2, c3, c4);

   return (0);
 }

 void calcResistance(char c1, char c2, char c3, char c4){


    int b1, b2, b3, total;
    double b4;

//band 1
```

```
    if(c1 == 'K'){//assigning band 1 with 0-90
        b1 = 0;
    }
    else if(c1 == 'N'){//going by increments of 10 since it is added to band 2 later
        b1 = 10;
    }
    else if(c1 == 'R'){
        b1 = 20;
    }
    else if(c1 == 'O'){
        b1 = 30;
    }
    else if(c1 == 'Y'){
        b1 = 40;
    }
    else if(c1 == 'G'){
        b1 = 50;
    }
    else if(c1 == 'B'){
        b1 = 60;
    }
    else if(c1 == 'V'){
        b1 = 70;
    }
    else if(c1 == 'E'){
        b1 = 80;
    }
    else {
        b1 = 90;
    }

//band 2

    if(c2 == 'K'){//assigning band 2 with 0-9
        b2 = 0;
    }
    else if(c2 == 'N'){
        b2 = 1;
    }
    else if(c2 == 'R'){
        b2 = 2;
    }
    else if(c2 == 'O'){
        b2 = 3;
    }
    else if(c2 == 'Y'){
        b2 = 4;
    }
    else if(c2 == 'G'){
        b2 = 5;
    }
    else if(c2 == 'B'){
        b2 = 6;
    }
    else if(c2 == 'V'){
        b2 = 7;
    }
    else if(c2 == 'E'){
        b2 = 8;
    }
    else {
        b2 = 9;
    }

//band 3

     if(c3 == 'K'){//assigning band 3 with appropriate multiplier
        b3 = 1;
    }
    else if(c3 == 'N'){
        b3 = 10;
    }
    else if(c3 == 'R'){
        b3 = 100;
    }
    else if(c3 == 'O'){
        b3 = 1000;
    }
    else if(c3 == 'Y'){
        b3 = 10000;
    }
    else if(c3 == 'G'){
        b3 = 100000;
    }
    else if(c3 == 'B'){
        b3 = 1000000;
    }
    else if(c3 == 'V'){
        b3 = 10000000;
    }
    else if(c3 == 'D'){
        b3 = 0.1;
    }
    else if(c3 == 'S'){
        b3 = 0.01;
    }

    else {
        b3 = 0;
    }

    //band 4
    if(c4 == 'K'){//assigning band 4 with appropriate tolerance
        b4 = 1;
    }
    else if(c4 == 'N'){
        b4 = 2;
    }
    else if (c4 == 'G'){
        b4 = 0.5;
    }
    else if(c4 == 'B'){
        b4 = 0.25;
    }
    else if(c4 == 'V'){
        b4 = 0.1;
    }
    else if(c4 == 'E'){
        b4= 0.05;
    }
    else if(c4 == 'D'){
        b4 = 5;
    }
    else if(c4 == 's'){
        b4 = 10;
    }
    else {
        b4 = 0;
    }

    total = ((b1+b2)*b3);//adding band 1 and 2 then multiplying by the multiplier

    printf("This resistor is [%d]Ohms with a +/-%g%% tolerance.", total, b4);//final print


return (0);
}
```

B. Source Code: Lab 2 Part 2 main.c

```
/* **************************************************************************************
```

```c
* Author: Jake Carlson
* Course: EGR 226 - 902
* Date: 02/1/2021
* Project: lab2part2
* File: main.c
* Description: This program uses structures to create a database of books from
* an external file.
*
***********************************************************************************
    */

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#ifndef NULL
#define NULL 0
#endif // NULL

typedef struct{//Defining the book structure
    char title[255];
    char author_name[50];
    char ISBN[10];
    int pages;
    int year_published;
}book;
int input(int low, int high); //Functions used in program
int parse_file(char name[], book my_book_array[]);
void print_book(book entered_array[], int enter);
void search_title(book my_book_array[]);
void search_author(book my_book_array[]);
void search_ISBN(book my_book_array[]);



int main()
{
   int i , j =1, low = 0, high = 2;
   char name[25] = "BookList.csv";
   book my_book_array[360];


  parse_file(name, my_book_array);//calling function to fill array

  while(j){
        i= input(low, high);//calling input function to use what user has entered
  if(i == 0){
    search_title(my_book_array);// if 0 the program will use title function
  }
  else if(i == 1){
    search_author(my_book_array);//if 1 the program will use author function
  }
  else{
    search_ISBN(my_book_array);// if 2 the program will use ISBN function
  }}

    return 0;

}
int input(int low, int high){

    int test = 1;
    int user, loop;
    char che[10];

    printf("What would you like to search by? [0] Title, [1] Author, or [2] ISBN\n");//user prompt

    while(test){//error checking loop
        loop = scanf("%d", &user);//sets whatever the user enters to loop
        if(loop != 1){
            printf("Try again.\n");
        }
        else if (user > high){
            printf("Try again.\n");//else if's to make sure the user enters between 0 and 2
        }
        else if (user < low){
            printf("Try again.\n");
        }
        else{
        test = 0;
        }
        fflush(stdin);
    }

    che[0] = user;
    che[strlen(che) - 2] = '\0';//using array to set the entry to the first value

    user = che[0];

    return user;

}

int parse_file(char name[], book my_book_array[]){
    int i = 0;
    char chec[512];

    FILE* infile = fopen(name, "r");//opening file and returning 0 in case the file does not exist
    if (infile == NULL){
        return 0;
    }

    while(fgets(chec, 512, infile)){//loop to collect each line from given csv file

        char* point = strtok(chec, ",");//adding commas and newline

        if(strcmp(point, "N/A"))//validate string
            strcpy(my_book_array[i].title, point);//title parse

        point = strtok(NULL, ",\n");
        if(strcmp(point, "N/A"))//validate string
            strcpy(my_book_array[i].author_name, point);//author parse

        point = strtok(NULL, ",\n");
        if(strcmp(point, "N/A"))//validate string
            strcpy(my_book_array[i].ISBN, point);//ISBN parse

        else
            strcpy(my_book_array[i].ISBN, "N/A");

        point = strtok(NULL, ",\n");
        if (strcmp(point, "N/A")){//validate string
            my_book_array[i].pages = atoi (point);//page count parse
        }
        else {
            my_book_array[i].pages = -1;
        }
        point = strtok(NULL, ",\n");
        if(strcmp(point, "N/A"))//validate string
            my_book_array[i].year_published = atoi(point);//year published parse

        else {
            my_book_array[i].year_published = -1;
        }
        i++;
    }
    fclose(infile);//closing file
    return (1);
}

void print_book(book entered_array[], int enter){

    int i;

    for(i = 0; i < enter; i++){//loop that prints title, author, ISBN, page, and year published in that order
        printf("Title: %s\n", entered_array[i].title);//title print
        printf("Author: %s\n", entered_array[i].author_name);//author print
        printf("ISBN: %s\n", entered_array[i].ISBN);//ISBN print

        if(entered_array[i].pages == -1){
            printf("Pages: N/A\n");//used in case pages are N/A
        }
        else{
            printf("Pages: %d\n", entered_array[i].pages);//prints pages
            fflush(stdout);
        }
        if(entered_array[i].year_published == -1){
            printf("Published: N/A\n");//used if year published is N/A
        }
        else{
            printf("Published: %d\n\n", entered_array[i].year_published);//prints year published
        }
    }
    return (0);

}

void search_title(book my_book_array[]){

    char cha1[50], cha2[50], cha3[50];
    int i, j = 0, num = 360;
    char* posis = NULL; //pointer in case no title is found
    book my_entered[512];

    printf("Enter Title criteria:\n");//user prompt


    fgets(cha1, 50, stdin);
    fflush(stdin);
    cha1[strlen(cha1)-1] = '\0';//fgets for the users search input

    strcpy(cha2, cha1);//copying the entered characters and changing the first to an uppercase character
    cha2[0] = toupper(cha2[0]);

    strcpy(cha3, cha1);//copying again to make all of the characters uppercase

```

```c
    for(i = 0; i < strlen(cha1); i++){
        cha3[i] = toupper(cha3[i]);
    }

    for(i = 0; i < num; i++){

        posis = (strstr(my_book_array[i].title, cha1));

        if(posis != NULL){//if statement when not NULL there is a match
            strcpy(my_entered[j].title, my_book_array[i].title);//copying one array to the others
            strcpy(my_entered[j].author_name, my_book_array[i].author_name);
            strcpy(my_entered[j].ISBN, my_book_array[i].ISBN);
            my_entered[j].pages = my_book_array[i].pages;
            my_entered[j].year_published = my_book_array[i].year_published;

            j++;
        }
        posis = (strstr(my_book_array[i].title, cha2));
        if(posis != NULL){//if statement when not NULL there is a match
            strcpy(my_entered[j].title, my_book_array[i].title);//copying one array to the others
            strcpy(my_entered[j].author_name, my_book_array[i].author_name);
            strcpy(my_entered[j].ISBN, my_book_array[i].ISBN);
            my_entered[j].pages = my_book_array[i].pages;
            my_entered[j].year_published = my_book_array[i].year_published;

            j++;
        }
        posis = (strstr(my_book_array[i].title, cha3));
        if(posis != NULL){//if statement when not NULL there is a match
            strcpy(my_entered[j].title, my_book_array[i].title);//copying one array to the others
            strcpy(my_entered[j].author_name, my_book_array[i].author_name);
            strcpy(my_entered[j].ISBN, my_book_array[i].ISBN);
            my_entered[j].pages = my_book_array[i].pages;
            my_entered[j].year_published = my_book_array[i].year_published;

            j++;
        }
    }
    if(strlen(my_entered[0].title) == 0){//check to see if character string exists and the length if it does
        printf("No Title found.\n");
    }
    print_book(my_entered, j);//passing needed array and int back to print book

    return (0);
```

```

}

void search_author(book my_book_array[]){

    char cha1[50], cha2[50], cha3[50];
    int i, j =0, num = 360;
    char* posis = NULL;//pointer in case no title is found
    book my_entered[512];

    printf("Enter Author criteria:\n");//user prompt


    fgets(cha1, 50, stdin);//fgets for user search input
    fflush(stdin);
    cha1[strlen(cha1)-1] = '\0';

    strcpy(cha2, cha1);//copying the entered characters and changing the first to an uppercase character
    cha2[0] = toupper(cha2[0]);

    strcpy(cha3, cha1);//copying again to make all of the characters uppercase

    for(i = 0; i < strlen(cha1); i++){
        cha3[i] = toupper(cha3[i]);
    }


    for(i = 0; i < num; i++){

        posis = strstr(my_book_array[i].author_name, cha1);

        if(posis != NULL){//if statement when not NULL there is a match
            strcpy(my_entered[j].title, my_book_array[i].title);//copying one array to the others
            strcpy(my_entered[j].author_name, my_book_array[i].author_name);
            strcpy(my_entered[j].ISBN, my_book_array[i].ISBN);
            my_entered[j].pages = my_book_array[i].pages;
            my_entered[j].year_published = my_book_array[i].year_published;

            j++;
        }
        posis = (strstr(my_book_array[i].author_name, cha2));
        if(posis != NULL){//if statement when not NULL there is a match
            strcpy(my_entered[j].title, my_book_array[i].title);//copying one array to the others
            strcpy(my_entered[j].author_name, my_book_array[i].author_name);
            strcpy(my_entered[j].ISBN, my_book_array[i].ISBN);
            my_entered[j].pages = my_book_array[i].pages;
            my_entered[j].year_published = my_book_array[i].year_published;

            j++;
        }
        posis = (strstr(my_book_array[i].author_name, cha3));
        if(posis != NULL){//if statement when not NULL there is a match
            strcpy(my_entered[j].title, my_book_array[i].title);//copying one array to the others
            strcpy(my_entered[j].author_name, my_book_array[i].author_name);
            strcpy(my_entered[j].ISBN, my_book_array[i].ISBN);
            my_entered[j].pages = my_book_array[i].pages;
            my_entered[j].year_published = my_book_array[i].year_published;

            j++;
        }
    }

    if(strlen(my_entered[0].title) == 0){//check to see if character string exists and the length if it does
        printf("No Author found.\n");
    }
    print_book(my_entered, j);//passing array and int to print book

    return (0);


}

void search_ISBN(book my_book_array[]){

    book my_entered[512];
    int i, j = 0, num = 360;
    char* posis = NULL;//pointer in case no title is found
    char cha1[50];

    printf("Enter ISBN criteria:\n");//user prompt

    fgets(cha1, 50, stdin);//fgets the users search input
    fflush(stdin);
    cha1[strlen(cha1)-1] = '\0';

    for(i = 0; i < num; i++ ){//going throughout the array to locate users search input
        posis = strstr(my_book_array[i].ISBN, cha1);

        if(posis != NULL){//match found again as long as not NULL
            strcpy(my_entered[j].title, my_book_array[i].title);//copying one array
```

```
to the others
            strcpy(my_entered[j].author_name, my_book_array[i].author_name);
            strcpy(my_entered[j].ISBN, my_book_array[i].ISBN);
            my_entered[j].pages = my_book_array[i].pages;
            my_entered[j].year_published = my_book_array[i].year_published;

            j++;
        }
    }
    if(strlen(my_entered[0].title) == 0){//check to see if character string exists and the length if it does
        printf("ISBN not found.\n");
    }
    print_book(my_entered, j);//passing array and int to print book

    return (0);
}
```