

EGR: 226 Microcontroller Programming and Applications

Winter 2021

Instructor Prof. Trevor Ekin

Lab 2: Programming Refresher Part 2

Gabriel Gasbarre

February 3rd, 2021

1. Objectives	3
2. Equipment	3
3. Introduction.....	4
3.1 Part 1: Resistor Analysis Tool Part 2.....	4
3.2 Part 2: Book Database.....	4
4. Procedure	4
4.1 Part 1: Resistor analysis tool.....	4
4.2 Part 2: Book Database.....	6
5. Results.....	8
5.1: Part 1	8
5.2: Part 2.....	9
6 Conclusions and Future Work	9

1. Objectives

Lab 2 is split up into two parts. Part 1 builds upon the resistor analysis tool created in lab one with the added objective of practicing console C programming by creating a function that accepts color bands and outputs the respective resistance. Part 2 is designed with the objective to develop a searchable database of books using C-structures

2. Equipment

Part	Description	Model
Code::Blocks	cross-platform IDE that supports multiple compilers	20.03
EGR:226 Structured Laboratory Activity	C programming refresher guide Part2	N/A

3. Introduction

3.1 Part 1: Resistor Analysis Tool Part 2

For Lab 1, students were asked to create a C program with function prototypes to prompt the user to input a desired Ohm reading for a resistor. Lab 2 Part 2 asks students to update this analysis tool to allow the user the choice between entering resistance or a color code, outputting the conversion in either direction.

3.2 Part 2: Book Database

For part 2, students were asked to create a C program that uses the concepts of structures to create a database of books from an external input file. This text file will be read into the program, parsed into an array of book structures, and then made available for the user to navigate using user-input of integers.

4. Procedure

4.1 Part 1: Resistor analysis tool

After copying in the information from part 1 and updating the prompt(void) display table, two functions were added to the program.

First, getColorBands() was added to collect the user input of 4 characters, all of which are error checked to ensure they align with the color scheme shown in Table 1 of the lab manual.

```
c X res_library.h X res_library.c X main.c X
1  #include "res_library.h"
2  #include <string.h>
3  #include <stdio.h>
4
5  void getColorBands(char* c1, char* c2, char* c3, char* c4){
6
7      int i=0;
8      char errorCheck[24] = {'K', 'k', 'N', 'n', 'R', 'r', 'O', 'o', 'Y', 'y', 'G', 'g', 'B', 'b', 'V', 'v', 'E', 'e', 'W', 'w', 'D', 'd', 'S', 's'};
9
10     printf("Input the four characters for the corresponding band colors\n");
11     printf("1.)");
12     scanf(" %c", c1);
13     while (strchr(errorCheck, *c1) == NULL ) {
14         printf("invalid character, please try again.\n");
15         scanf(" %c", c1);
16     }
17     printf("\n2.)");
18     scanf(" %c", c2);
19     while (strchr(errorCheck, *c2) == NULL ) {
20         printf("invalid character, please try again.\n");
21         scanf(" %c", c2);
22     }
23     printf("\n3.)");
24     scanf(" %c", c3);
25     while (strchr(errorCheck, *c3) == NULL ) {
26         printf("invalid character, please try again.\n");
27         scanf(" %c", c3);
28     }
29     printf("\n4.)");
30     scanf(" %c", c4);
31     while (strchr(errorCheck, *c4) == NULL ) {
32         printf("invalid character, please try again.\n");
33         scanf(" %c", c4);
34     }
35
36 }
37
38 void calcResistance(char c1, char c2, char c3, char c4) {
39
```

Next, calcResistance() was added to receive the 4 characters from the previous function to determine and describe the color code of a resistor that corresponds to the information entered. This is a simple if/else tree that assigns each character value to its respective ohm value.

```

35
36
37 }
38 void calcResistance(char c1, char c2, char c3, char c4) {
39
40     int res=0;
41
42
43     if (c1 == 'K' || c1 == 'k')
44         res = 0;
45     else if (c1 == 'N' || c1 == 'n')
46         res = 10;
47     else if (c1 == 'R' || c1 == 'r')
48         res = 20;
49     else if (c1 == 'O' || c1 == 'o')
50         res = 30;
51     else if (c1 == 'Y' || c1 == 'y')
52         res = 40;
53     else if (c1 == 'G' || c1 == 'g')
54         res = 50;
55     else if (c1 == 'B' || c1 == 'b')
56         res = 60;
57     else if (c1 == 'V' || c1 == 'v')
58         res = 70;
59     else if (c1 == 'E' || c1 == 'e')
60         res = 80;
61     else if (c1 == 'W' || c1 == 'w')
62         res = 90;
63
64
65     if (c2 == 'K' || c2 == 'k')

```

Above is a snippet of the first section of the function that assigns the first value, a multiple of 10, to the resistance value.

4.2 Part 2: Book Database

For part 2, we start by creating a book structure, which will be turned into an array to store each value read by our program. In order to read our file, we have a function called

parse_file() that takes a parameter of filename and the array of book structures previously created.

Parse_file() works by reading the text file and tokenizing it, storing each value to its respective position: author to author, title to title and so on. Listed below is a snippet that reads the file and stores the first tokenization to the title portion of the book structure array.

```
9
10 int parse_file(char filename[50], book book_array[]) {
11     /* Brief: Reads file input from user and tokenizes information
12     * param:  Filename, book_array
13     * return: N/A
14     */
15     /*
16     FILE* infile = fopen(filename, "r");
17     if (infile == NULL) {
18         printf("Failed to open file with name '%s'", filename);
19         return 0;
20     }
21     char buffer[512];
22     int i = 0;
23     while (fgets(buffer, 512, infile)) {
24         char * ptr = strtok(buffer, ",");
25         if(strcmp(ptr, "N/A")){
26             strcpy(book_array[i].title, ptr);
27         }
28         else if (strcmp(ptr, "N/A") == 0) {
29             strcpy(book_array[i].title, "N/A");
30         }
31     }
32     book_parse_file::book_array
33 }
34
```

Next, functions were added to sort and display the information read. These functions allow for user input and display information matching to the string entered. For example, our first function search_title() allows a user to input a string like “The” which will then return any book with the string “The” in the title, and how many books were found.

The same functions were added except changed to allow searching for authors, and also ISBN numbers. An example of the function search_title() is shown below.

```

62
63
64
65 void search_title(book book_array[],int n, char title[]){
66     /***| search title | *****/
67     * Brief: receives a book array and title to compare, and sends matches to function print_book
68     * param: book_array, n (number of books in array) and title.
69     * return: printf staement with # of matching books found
70     *
71     *****/
72     int i = 0;
73     int j = 0;
74     for (i ; i < n; i++) {
75         if (strstr( book_array[i].title, title) != 0) {
76             print_book(book_array[i]);
77             j++; }
78     }
79     printf("Found %d books with that title string.\n\n", j);
80 }
81
82
83
84

```

5. Results

5.1: Part 1

Running the resistor analysis tool worked successfully and provided the desired output.

Listed below is an example of how the program responds to the input of “G, G, Y, N” without another desired input.

```

E | Grey | 8 | | +/-0.05%|
W | White | 9 | | |
D | Gold | | | +/-5|
S | Silver | | | +/-10% |
-----
Would you like to
1.) Determine the color code from a resistance value
or
2.)Convert a color code to a resistance?
Enter '1' or '2'

2
Input the four characters for the corresponding band colors
1.)G
2.)G
3.)Y
4.)N
You entered: G, G, Y, N

Resistance: 550000 +/-2%

Would you like to input another resistor?
Y/N?

```


5.2: Part 2

Shown below is an example of the Book database reading in a file called “BookList.csv.” We then decide to search by title with the string “The Re” which returns one result. The program then loops and allows an infinite amount of user requests and inputs.

```
Please enter the filename you would like to read in.
BookList.csv
Select one of the following options to search by:
[0]Search by Title
[1]Search by Author Name
[2]Search by ISBN
50
Enter the title of the book you would like to search
The Re
The Return of Sherlock Holmes (Sherlock Holmes #6), Arthur Conan Doyle, 192123173, 455, 1993
Found 1 books with that title string.

Select one of the following options to search by:
[0]Search by Title
[1]Search by Author Name
[2]Search by ISBN
-
```

6 Conclusions and Future Work

I found this week’s labs to be quite fulfilling, especially part two. It forced me to come up with many of my own function designs and taught me a lot about structures, reading files, and strings that I had not known before. I am also quite happy with the completeness of my function and would say there are even some bonus functions added like displaying how many matching books were found. I look forward to future labs!