Reta Gela
November 2, 2023

ENEB 340 Lab 7 Report

Introduction

In this lab, the objective was to apply knowledge of functions, and multi-filing programming to compile multiple files to complete 3 assignments. The first program was to add 2 matrices and print the resulting sum matrix. The second program had to take a matrix and transpose it and give it back to the user. The last task was to convert a string entered by the user in “MM/DD/YYYY” format into the American format “month day, year.”

Procedure

1. First, the function.h and makefile files were written and stored into the Raspberry Pi directory “lab 7” with the suggested declaration names of the functions.
2. The function.c was edited and all the functions were written beforehand.
3. A separate main.c file was created and the first program was simulated using this main.c file as the driver program.
4. The code was compiled and tested with the results shown in Figure __.
5. After this, the main.c driver code was commented out and the next driver program was written to call the second function. The code was compiled and the results of the test case simulation are in Figure __.
6. Finally, the driver program for the second program was commented out and the last program driver code was written to compile with the third function in the functions.c file.
7. An image of the test case results was taken (see Figure __)

Method

<u>Pseudocode For Program 1</u>

Main loop

1. Prompt user for matrix A and matrix B
2. Pass Matrix A and B to function in “functions.h” file using statement
    a. Addition function takes matrix A and Matrix B
    b. Loop 1 : using var1
        i. Iterate through all rows using variable
            1. Nested loop (using var2)
                a. Iterate through all columns
                    i. Add a[var1][var2] and b[var1][var2] together
                    ii. Set equal to result matrix
            2. Return resulting matrix to printMatrix function
    c. Program terminates

Pseudocode for Program 2

Main loop

1. Prompt user for a matrix
2. Pass the matrix to transpose function in “functions.h” file
    a. Transpose function takes the matrix
    b. Creates an empty matrix of equal dimensions
    c. Loop 1 : using var1
        i. Iterate through all rows using variable
            1. Nested loop (using var2)
                a. Iterate through all columns
                    i. Swap the row and column values using var1 and var2 inside the empty matrix and equate it to the original matrix
            2. Return resulting matrix to printMatrix function
    d. Program terminates

Pseudocode for Program 3

Main loop

1. Take string from User
2. Scan the string of “MM/DD/YYYY” format separately into 3 integer variables ‘month’ ‘day’ and ‘year’
3. Pass the three variables to the date converter function in function.h file
    a. Date function takes the integer variables
        i. Loop 1
            1. Iterate through each possible month (1-12)
                a. Print corresponding month
        ii. Print out the day and year variables separated by a comma
    b. Program terminates

Pseudocode for Print Matrix function (used for first 2 programs)

    c. Loop 1
        i. Iterate through each row
            1. Loop 2: Iterate through each column
                a. Print each variable with space
            2. Indent before next iteration
        ii. Program terminates

Results

```
r374@raspberrypi:~/lab7 $ make
gcc -c functions.c
gcc -o main main.o functions.o
r374@raspberrypi:~/lab7 $ ./main

Enter number of rows: 2

Enter number of columns: 2

Enter Matrix A: 1 2 3 4

Enter Matrix B: 5 6 7 8

Result:
6 8
10 12

r374@raspberrypi:~/lab7 $
```

Figure 1: Matrix Adder Program Test Case Results

```
r374@raspberrypi:~/lab7 $ make
gcc -c main.c
gcc -o main main.o functions.o
r374@raspberrypi:~/lab7 $ ./main

Enter number of rows: 1
Enter number of columns: 2

Enter Matrix: 1 2

Result:
1
2

r374@raspberrypi:~/lab7 $ ./main

Enter number of rows: 3
Enter number of columns: 2

Enter Matrix: 1 2 3 4 5 6

Result:
1 3 5
2 4 6
```

Figure 2: Test Case Results of Transpose Program

```
{

    int month = 0, day = 0;

    if(m == 1) {
        printf("January ");
    }
    else if(m == 2) {
        printf("February ");
    }
    else if(m == 3) {
        printf("March ");
    }
    else if(m == 4) {
        printf("April ");
```

```
Enter a Date: 2/3/2013
February 3, 2023

...Program finished with exit code 0
Press ENTER to exit console.
```

Figure 3: Test Case 1 Result of Date Converter

```
    }
    else if(m == 4) {
        printf("April ");
    }
    else if(m == 5) {
        printf("May ");
    }
    else if(m == 6) {
        printf("June ");
    }
    else if(m == 7) {
        printf("July ");
    }
    else if(m == 8) {
        printf("August ");
```

```
Enter a Date: 29/3/2013
Invalid Date

...Program finished with exit code 0
Press ENTER to exit console.
```

Figure 4: Test Case Result 2 for Date Converter

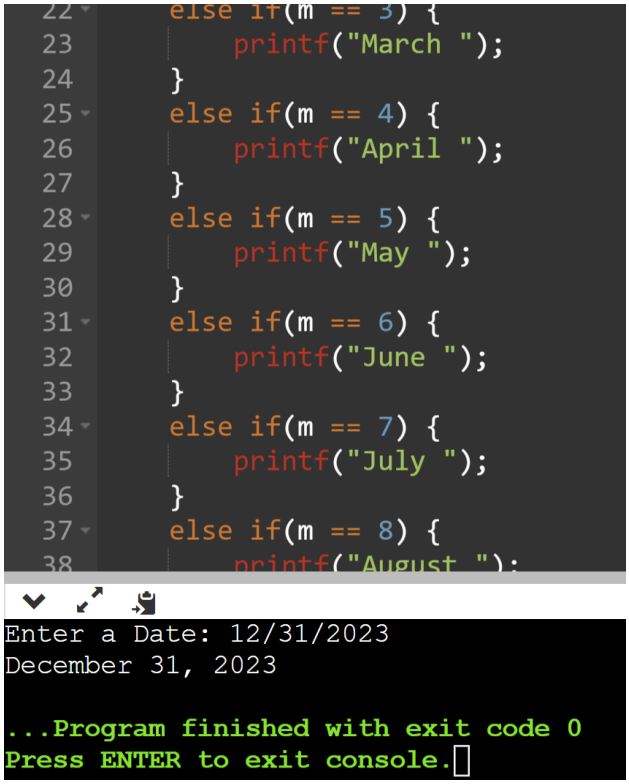

Figure 5: Test Case Result 3 for Date Converter

NOTE: I was unable to finish the third lab investigation in class so I used an online compiler at home to compile the results. I have demonstrated how to compile multiple files using a makefile for the first 2 assignments.

Conclusion

After completing this assignment I learned a lot about multi-file programming, constructing makefiles, matrix manipulation as well as using void function loops. It is clear to me that using functions is extremely helpful even though one cannot really identify errors in the code as well compared to if all of the code was compiled in one file.

In the future, I would like to continue using these concepts but maybe with fewer problems, because I was unable to complete all 3 program functions in the lab session and had to use an online compiler at home to test the program. I am sure that I will become much faster at completing these assignments however.