C ASSIGNMENT - CPSC 2310

DUE: Sunday, March 14, 2021 Midnight

Learning Objectives:

This assignment is designed to provide practice with the following:

- **Linked List**
- **Function Pointers/Call back function**
- **makefile**
- **Reading comma delimited files using scanset**
- **Structs**
- **enums**

Overview:

Read the entire document to ensure you are aware of all requirements. I will not accept the excuse you did not know you were supposed to do something.

This is an individual assignment, you are not allowed to receive help from anyone other than, myself, or the 2311 lab TA's. There are several times in this document I suggest you perform an internet search. This is perfectly fine. Any other information from the internet or any other source must be documented in your README file. A violation of this or working with another student could result in a violation of academic integrity charge.

There are many ways to complete this assignment, but to force you to use several new functions/concepts in "C", I am going to be very specific on much of the instructions. I am aware that some of what I am going to require you to do could be done easier, and maybe even more efficiently. However, as stated above it is my intention to introduce you to several concepts that some of you may not be familiar with. I strongly suggest you read the entire document and make sure you understand completely what you are required to do. **Points will be deducted for not following directions.**

The program you are going to write will read data from a comma delimited file. Store the data in a linked list and write the data to a file in several different formats. Sounds simple right!?!

Now we will discuss the requirements in more detail.

Requirements:

Scanset conversion:

Most of you know how to read and write information using scanf/printf or fscanf/fprintf. For this assignment, you are going to read data from a comma delimited file using the concept of scanset conversion. In class we discussed and saw examples of scanset conversion. Using scanset conversion when reading the data is a requirement not a suggestion. Below are several links that may help you understand scanset:

https://www.geeksforgeeks.org/scansets-in-c/
https://www.knowprogram.com/c-programming/read-and-display-the-string-in-c-programming/

C ASSIGNMENT - CPSC 2310

DUE: Sunday, March 14, 2021 Midnight

http://www.cplusplus.com/reference/cstdio/fscanf/
https://www.tutorialspoint.com/scansets-in-c

So, what is the data that you will read? In an earlier lab I had you complete a google form. The form asked for your first and last name, birthday, and favorite song. This is the data you will read.

Below is an example of several of your TA's data, (I asked them to complete the form as well):
Narula,Devin,4,16,2001,Free Bird
Dominic,James,5,21,1988,Shine on you crazy diamond.
Aiyetigbo,Mary,10,3,1988,I have many

Structs:
Each set of data will be stored in a linked list. You **must** use two structs; one for the birthday and one that will be used as a node for the linked list.

Enum:

"Enumeration is a user defined datatype. It is used to assign names to the integral constants which makes a program easy to read and maintain. The keyword "enum" is used to declare an enumeration." Tutorials point. https://www.tutorialspoint.com/enum-in-c

So how is this going to help you in this project?

You will write a function that will print the birthday of each person in the data file. As shown above, the birthday is represented using integers. However, the birthday month will be printed as a string rather than an integer. This is a perfect opportunity to use an enum in conjunction with a const char* array[] to represent the months in the form of a string.

Below are several example uses of enums:

https://www.programiz.com/c-programming/c-enumeration
https://www.geeksforgeeks.org/enumeration-enum-c/
https://www.cs.utah.edu/~germain/PPS/Topics/C_Language/enumerated_types.html

Linked List:

As mentioned earlier, you will create a struct for the linked list that will have the following data members:
3 strings for the first and last name and favorite song
An instance of the struct that represents the birthday
A self-referential pointer of the struct you created to represent each node in the list

C ASSIGNMENT - CPSC 2310

DUE: Sunday, March 14, 2021 Midnight

Files:
You will create functions.h, functions.c, and driver.c files.

driver.c
Main will be included in driver.c. Driver.c should have minimal amount of code in it. Things you may have in the driver: Create the input and output files. These files will be given on the command line. You will need to open these files and check that they opened correctly. Call createList and then call the function printName, printBDay, and printSong. Also, don't forget to delete the nodes in the list.

Points will be deducted if you have excessive code in driver.c

functions.h
This file contains the declaration of the structs, all #includes, and the function prototypes. You must also include header guards in this file. Before each function prototype**,** you must have a detailed description of what the overall function does. What the parameters were and what is being returned. Here is an example of an overall function description.

```
/* Parameters: img - image_t pointer array holding the image data for
 *                   each of the input files
 * Return:       output - image_t struct containing output image data
 * This function averages every pixels rbg values from each of the
 * input images and puts those averages into a single output image
 */
```

You are required to have this type of comment block before each function.

functions.c
This file is where you will define the enum and any other data type needed. You will implement all functions in this file.

Functions:

Below I will provide a short description of each function:

void add(node_t **node, node_t **head) – This is the function used to add the node to the linked list

node_t* readNodeInfo(FILE* input) – This function will be used to read the data from the input file. This is where you will allocate the memory for the node that will eventually be added to the linked list. Using scanset conversion you will read the data and store it in the node allocated. You must also check the validity of the birthday. If the birthday is invalid, then do not add the node to the linked list. Returns the node created.

node_t* createList(FILE*, node_t)** – This function is called in main and starts the process of creating the list. You will use a loop to read each record in the file then add the record to the list. Returns a pointer to the head of the list.

void PrintList(FILE*, node_t*) – This function prints, to the output file, the data from the list. If the list is empty you are required to print a message, to stderr, indicating the list is empty and exit the program. If the list is not empty you are to print LIST INFO: then print the information for each node in the list. Described below is a function called printBorder which prints a line of 80 asterisk "*". You will call this function before printing the list and after printing the list.
An example of the print format is below:

Last name, first name, birthday in the format of January 1, 2000, favorite song

Example:
****
LIST INFO:
Feaster, Yvon, October 7, 1963, I'm moving on
:
:
****

void PrintName(FILE*,node_t*) – This function prints, to the output file, only the name from each node in the linked list. If the list is empty print, to stderr, a message indicating the list is empty and exit the program. If it is not empty print "NAMES:" then print the last name and first name. You are also required to call printBorder before and after the list of names.

Example:
****
Feaster, Yvon
:
:
****

void PrintBDay(FILE*,node_t *) – This function prints, to the output file, the first and last name and the birthday from each node in the list. The format you must use is shown in the example below.

Example:
****
Yvon Feaster's date of birth is October 7, 1963
:
:
****

void Song(FILE*,node_t *) – This function prints, to the output file, the first and last name and favorite song from each node in the list. The format you must use is shown in the example below.

C ASSIGNMENT - CPSC 2310

DUE: Sunday, March 14, 2021 Midnight

Example:

****

Yvon Feaster's favorite song is I'm moving on

:

:

****

void printBorder(FILE*) – This function prints, to the output file, 80 asterisk "*".

void print(void (*fp)(FILE*,node_t*),FILE*, node_t*) – this is a function that you will use to call the print functions listed above. Notice this has a function pointer as a parameter. We have covered several examples of this concept in class.

void checkArgs(int) – This function checks the number of command line arguments. If the incorrect number of arguments are passed to the command line you are to print a message, to **stderr**, indicating this and exit the program.

void checkFile(FILE*, char*) - - This function determines if the file opened successfully. If the file does not open successfully you must print to **stderr** a statement indicating this and exit the program. In your statement you must indicate which file did not open successfully.

void deleteList(node_t)** – After you are finished with the nodes in the list you need to give the memory back to the system. That is what this function does.

bool checkDate(bday_t) – This function is used to check the validity of the date. So, what does this mean? You must check that the month and days are within the appropriate bounds. The year should be within 1900 – 2020. In order to check the day of month, for February 29, you need to know if the year is Leap Year. You will need to call the function isLeapYear. If any of the test fail you must print, to stderr, which test failed. YOU ARE NOT TO EXIT THE PROGRAM AT THIS POINT. Remember when you are reading in the data I stated you are to check the validity of the date. This is the function you call to do that.

bool isLeapYear(int) – return a bool that indicates rather the year passed to the function is leap year or not.

There will be 3 opportunities for EC on this assignment. Two of the EC are worth up to 5 points each, with the third worth up-to 10 points. You are only allowed to complete one of the ECs. Do EC does not change any other requirements of this assignment.

Extra Credit 1 (5 points)

Sort the linked list by last name as you are adding the node to the list.

C ASSIGNMENT - CPSC 2310

DUE: Sunday, March 14, 2021 Midnight

Extra Credit 2 (5 points)

Add a menu to the code that ask the user what they want to print. One of your menu options must be quit. You should continue to give the user the option to print different forms of the data until the user chooses the quit option. You must make sure the user enters a valid menu option.

Extra Credit 3 (10 points)

Before reading the data, display a menu asking the user how they want to sort the data; by first name, last name, birthday, or not at all. When creating the list, sort the data based on the users choice. You must make sure the user enters a valid menu option.

All other requirements are the same.

Formatting, Compiling, and Hand-in Requirements:

Tar your file **PA1.tar.gz** and submit the file through **hand-in to the folder PA1.**

If you are doing the EC you must submit a tarred file based on which EC you chose to complete. Submit your tarred file **PA1EC<number of EC completed>.tar.gz**, through **hand-in to one of the three folders PA1EC1, PA1EC2, or PA1EC3. PLEASE SUBMIT TO ONLY ONE HAND-IN FOLDER.**

DO NOT TAR.GZ THE FOLDER YOUR FILES ARE IN, ONLY TAR.GZ THE FILES. POINTS WILL BE DEDUCTED IF OUR TAR THE FOLDER.

You should submit the following files.

- driver.c
- functions.c
- functions.h
- makefile - your makefile should include a make run and make clean. When I type make run your program should compile then run. The input and output files will be defined on the command line. Please do not submit the input nor output files. I will add these to your submission folders. The names of the files I will add will be input.txt and output.txt.
- A README file that has the following information:
  - problems you encountered with this assignment
  - how you solved the problems
  - your thoughts about the assignment.
- With the exception of the makefile, you should have a header in each of your files that contains the following information. If you neglect to place a header in a file, you will receive a 5-point deduction.

C ASSIGNMENT - CPSC 2310

DUE: Sunday, March 14, 2021 Midnight

```
/*************************/
 *Your name                *
 *CPSC 2310 Spring 21*
 *UserName:                *
 *Instructor:  Dr. Yvon Feaster  *
/*************************/
```

- Your program should compile with no warnings and no errors on the School of Computing servers. I will use the School of Computing servers to test your program. If your program fails to compile you will receive a 0 on the assignment. If the program compiles but has warnings, there will be a minimal of a 10-point reduction. (The more warnings you have the higher the deductions.)
- Your code should be well documented. (comments – see example above) If you do not provide the appropriate documentation a minimal of 5 points will be deducted for each function that is not documented.
- There should be no line of code longer than 80 characters. There will be a minimal of a 5 point deduction for this violation.
- You must use proper and consistent indention. There will be a minimal of a 5 point deduction if your code is not neat with proper and consistent indention.

Failure to do any of the above items may result in a deduction of points for each offense.