

CSC 362 Programming Assignment #4

This assignment will involve everything you have learned up until this point. There are multiple ways to do it, and applying you knowledge of pointers and structures will certainly be the simplest.

Your program will first read from two files. The first is a comma-separated list of names and addresses (called person.txt), the second is a comma-separated list of birthdates (called dob.txt). There should be two structs that correspond to these two types of values: a Person struct and a Date_of_Birth struct. You will populate these structs with the values from the files and add them to an array as you create them. Please note that corresponding lines from both files belong to belong to the same record. E.g., a person's information on line 1 in the persons file corresponds to DOB information in DOB file line 1.

Then, sort the array according to the name field using any sorting algorithm you wish. Once the array is sorted, write it out to an output file called output.txt. This program is not required to have console output, only a populated and sorted output file. You can optionally print out the data to the console as it is written to the output file since it should not be much extra work.

Hint: This program may be simpler if you choose to do it using multiple files. A header file could contain all of your function prototypes, structures and constants, a driver/main file could simply open your input and output files and call your functions, and an implementation file could contain all of the functions.

Submit your code file(s) on Canvas

Here are some (big) hints :

- You may want to consider using these structs based on the input you will see in the files:

```
struct DATE {  
    int month;  
    int day;  
    int year;  
};
```

```
struct PERSON {  
    char* last_name;  
    DATE* date_of_birth;  
    int ssn;  
    char* street_address;  
    char* city;  
    int age;  
};
```

- It may also be helpful to think about dividing this up into 4 separate functions (does not include your main function)
 - Fill your array
 - Sort your array
 - Write to output file
 - Free your memory
- Make friends with the strtok(...) function!

- This program is very involved and there are some advanced topics which you will need to understand to finish it. For example, it might be helpful to use “*double pointers*” which are simply pointers to pointers (array of structures is a good example...)
 - Use print statements throughout your program so you know exactly what is happening every step of the way since there is a lot of file IO going on.

Sample output (Note the sorted order and the organization of the data including DOB):

Blake 191553 112 Bay Street Lancaster 23
Date of Birth: 8/14/1979

Brown 750977 21 Central Street Ballard Hill 21
Date of Birth: 5/29/1955

Bush 411499 34 Kennedy Street Littleton 44
Date of Birth: 12/23/1957

Checchi 452759 10 Young Street Sterling 56
Date of Birth: 1/23/1983