

LARGE PROGRAM 4 DATABASE RECORDS

COP3223C Introduction to Programming with C
Dr. Andrew Steinberg

Fall 2022

Background Story of this Assignment

You are about to implement your fourth and final large program in this course! In this assignment, the primary objective is to apply all your skills from everything you learned in the beginning up until now!!

You will write a menu driven program in C that simulates an inventory database system. The theme of the database is up to the student. However, please keep the theme appropriate. In order to do this you will have to use an array of typedef structs.

The program will have the following setup.

- The program will ask the user what kind of action they would like to perform. Actions include:
 - Insert (add a brand new record to the array)
 - Remove (remove an existing record from the array)
 - Search (check to see if a record exists in the array)
 - Display (display all records in the array)
 - Exit (program exits and writes to a text file all records of the array)

You will create a file called `LargeProgram4_lastnamefirstname.c` for this assignment and write out the code. You are allowed to use the same message that is provided in the sample output for your program. You can also make modifications to the text of messages as long as the program follows all directions as stated in this assignment. Check out the sample output of the fully working program on Webcourses. In my sample output, the theme of my database comes from the Harry Potter Series. I've created a Hogwarts School Database System..

Requirements for Large Program 4

Requirements for the Program:

1. Please do not use dynamic memory for this assignment. Only static arrays!
2. The program must have a minimum of 6 user-defined functions. It is up to the student of how the functions are utilized in the program.

3. Program must be able to support an insert, remove, locate, and display functionality.
4. The maximum size of the array is 30 elements.
5. The struct must be typedef. The minimum number of components for the typedef struct is 4. Students are allowed to have more components based on the theme of their respective large program.
6. When the program runs at the beginning you are required to populate the array with 6 records. Hint: I recommend students create a Hardcode Six function that will hard code values into the array.
7. Use good code practice skills you learned throughout the semester. For example, do not use global variables. **Anything that is not good practice will result in point deductions.**
8. Before your program terminates, make sure to print all records stored in the array to a text file called Records.txt.
9. Your program must be able handle error input without crashing!

Due Date

The assignment is due on November 30th at 11:59pm EST via Webcourses. **Do not email the professor or TAs your submissions as they will not be accepted!** This assignment is accepted late up to 24 hours with a penalty. Please see the syllabus for more information on this. Make sure to submit on time to get potential full credit. Make sure to also take into consideration the uploading time. In the past, students who are working last minute on the assignment sometimes run into uploading issues where their Internet may run slow, resulting in late submissions. The timestamp Webcourses uses for your submission will be applied and will be the final say. Please do not email the instructor or TAs saying your Internet was running slow. If the time is off by a second of the due date, then the assignment is considered late. Plan accordingly!

Important! Read Carefully!

You will create a file called LargeProgram4_lastnamefirstname.c for this assignment and write out the code. You are allowed to use the same message that is provided in the sample output for your program. You can also make modifications to the text of messages as long as the program follows all directions as stated in this assignment. Check out the sample output of the fully working program on Webcourses. In my sample output, the theme of my database comes from the Harry Potter Series. I've created a Hogwarts School Database System. **You can add any additional functions, but cannot remove the required ones requested in this assignment.**

Submit the .c file only of your program to Webcourses. Make sure to name your file *LargeProgram4_lastnamefirstname.c*. For example, my file would be named *LargeProgram4_SteinbergAndrew.c*. You do not need to submit the text file output. This will be graded through the compiler on Eustis. If it does not compile, it does not work properly!

Testing on Eustis

It is your responsibility to test your code on Eustis. If you submit your assignment without testing on Eustis, you risk points being deducted for things that may behave differently on your operating system. Remember, you cannot dispute grades if your code didn't work properly on Eustis all because it worked on your machine. The Eustis environment gives the final say. Plan accordingly to test on Eustis!!

Why was I not provided a Python Script File?

You are probably wondering why Dr. Steinberg didn't provide a python script to check your code like in the small programs. The reason is that Dr. Steinberg wants his students to enjoy the assignment without being told of what must be displayed to the terminal and matching exactly. For large programs, Dr. Steinberg allows students to change the text output to the terminal, however your program must perform the specific functionalities that is requested at least to receive potential full credit.

The Rubric

Please see the assignment page for the established rubric on webcourses.

Comment Header

Make sure you place a comment header at the top of your C file. You will use single line comments to write your name, professor, course/section, and assignment. For example, Dr. Steinberg's header would be:

```
//Andrew Steinberg
//Dr. Steinberg
//COP3223C Section 1
//Large Program 4
```

Missing a comment header will result in point deductions!

The Solution Text File

I have provided a sample output of a working program of what is expected on Webcourses. As I stated before, you are welcome to change the text output to the terminal, or you can keep it exact like in the sample.

Recommendations for Completing the Assignment

Dr. Steinberg believes in you all! However, the TAs/ULAs and Dr. Steinberg are here to help still!

Basic Rules for Completing The Assignment

Please make sure to follow these rules when completing the assignment.

- Please use all items that were presented to you in class and labs. Everything that was presented to you can help you complete the assignment.
- DO NOT COPY ANYONE'S CODE! This is cheating as stated in syllabus. This assignment is individual work.
- DO NOT SHOW YOUR CODE to classmates. The only group of people that can see your code are the TAs, ULAs, and Course Instructor. Follow the rules from the syllabus if you talk to other students. Otherwise you may be in cheating violations!
- Make sure to use good practice of writing code. Anything that is considered bad practice will result point deductions. Example GOTO statements should not be seen anyone's code. There is a reason I don't go cover them. They are extremely bad in practice to use and just awful.

Some Advice

Here are some tips and tricks that will help you with this assignment and make the experience enjoyable.

- Do not try to write out all the code and build it at the end to find syntax errors. For each new line of code written (my rule of thumb is 2-3 lines), build it to see if it compiles successfully.
It will go a long way!
- After any successful build, run the code to see what happens and what current state you are at with the program writing so you know what to do next! If the program performs what you expected, you can then move onto the next step of the code writing. If you try to write everything at once and build it successfully to find out it doesn't work properly, you will get frustrated trying find out the logical error in your code! **Remember, logical errors are the hardest to fix and identify in a program!**

- Start the assignment early! Do not wait last minute (the day of) to begin the assignment.
- Ask questions! It's ok to ask questions. If there are any clarifications needed, please ask TAs and the Instructor! We are here to help!!! Do not wait last minute to seek help as generally you could be waiting in a long line of fellow classmates who may need help. As stated previously start early and ask your questions right away!