# CMSC257 - Computer Systems

## Project 2: C-Strings and Structs

## Objectives:

The primary goal of this project is to enhance your proficiency in using C strings and structs. Additionally, you will learn to utilize format specifiers for formatted printing, pointers, and creating Makefiles. Effective use of the GDB debugger and Valgrind tools to debug your program and fix memory errors is expected.

The project involves building a program for a small employee database, with an emphasis on the use of C-strings, which is an essential aspect of this assignment.

## Description:

This project requires you to develop an employee database application with a menu-driven interface. The system should be capable of sorting the employee database, displaying employee records, searching for records, and deleting records.

## Input:

Menu-driven interactive program. The initial database will be loaded from a file.

## Output:

A sample output is provided. Your program output must exactly match the given sample output. (Both text and formatting)

## Instructions:

### Step 1:

* Copy starter files from **/home/sonmeza/projects/project2**
* Create a Makefile to build your program.

### Step 2:

Define a structure with the tag name Employee (within p2-support.h), with the following members:

* **ID (ID)**: a character string with a size of 6
* **First Name (first\_name): a** character string with a size of 10
* **Last Name (last\_name): a** character string with a size of 10
* **Email address (email)**: a character string with a size of 20
* **Employee Salary (salary)**: a double type.
* **Hire Date (hire\_date)**: a date type. (use the date type given in the header file.)

Then your “DataBase” struct should be defined (within p2-support.h) as:

* + A dynamic array of employees. (Note that you cannot declare a dynamic array within the struct definition. You only need to create a pointer to point to the dynamic array within the definition. The array is already allocated within the initialize function)
* A variable to hold the number of current employees in the database (type of int).

Read all comments in the file carefully. There are hints and instructions.

### Step 3:

* Implement the functions for displaying the main (display\_menu) menu and options in the menu in the **p2-support.c** file and add headers for functions into the **p2-support.h** file.
* Read all comments in the file carefully. There are hints and instructions.

The main menu of your program should print as follows:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\* Employee Database - Total Employees: 010 \*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Choose one of the menu options below:

1. Sort by ID

2. Sort by Hire Date

3. Display all employees

4. Search employees by ID

5. Search employees by Last Name

6. Delete employee by ID

7. Save

8. Exit

Note that there is an integer indicating the number of employees in the title.

Following is the description of each menu option:

1. Sort records with ID numbers from lowest to highest.
2. Sort records with hire date from earliest to most recent.
3. Display all records as ordered in the database.
4. Display the employee with the given ID. Will print a message if an employee with a given ID does not exist.
5. Search the database by last name and display employee(s) partially or completely matching with the given string. This option may result in multiple hits, for example, there might be more than one entry with the same last name, but a different first name.
6. Delete one record with the matching ID from the list. Print a message if no match.
7. Save the database to another file named as database\_updated.
8. Exit the program. Should also deallocate all dynamically allocated memory.

Part of the program is given to simplify your job:

You are given the **cmsc257-p2.c** file. Do not change that file, otherwise, your code may not work on Gradescope. I will use the same code in Gradescope.

You are given a starter **p2-support.c** file. Add all other function definitions within this file. You are to implement at least one function per menu item (see the comments in the display\_menu function) except for menu items 7 and 8. You may need to implement extra supporting functions as needed.

You are given a starter **p2-support.h** file. Include all function headers and structs within this file. Do not forget to add comments for each function in the header file.

### Step 4:

* Add formal comments into the p2-support.h file. An example is given for the initialize function.
* Add your name and date information.
* You can add other comments in p2-support.c file. But those comments are only for the purpose of reminding yourself what you were doing when you look at the code later.

### Step 5:

* Test your program with Valgrind. Make sure there are no memory leaks or errors.
* Once you are finished check the “to turn in” section to prepare all the files to be submitted.

## Print Formatting

**Displaying Menu:**

Menu title: 62 charters long

Total employees should always be printed as right aligned and 3 characters.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\* Employee Database - Total Employees: 010 \*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Displaying employees(Menu item 3):**

No: 3 characters left aligned

ID: 5 characters, right aligned and zero filled.

Name: Limited to 13 characters, first name, and last name is combined when printing. Truncate to 13 characters if needed. But, do not change the original database.

Salary: 10 characters. The first character is always a $ sign. The salary amount is right aligned with two digits after the decimal point.

Date: in MM/DD/YYYY format. 10 Characters in total.

No. ID Name Email Salary Hire Date

--- ----- ------------- ------------ ---------- --/--/----

0 00003 Wei Martin wm@vcu.edu $ 98999.99 01/10/2023

1 00007 Mitch Martine mitch@vcu.ed $ 78999.88 12/15/2022

2 00008 David Boyle db@vcu.edu $100000.01 11/10/2022

**Displaying Search Results and Deleted employees (Menu items 4-6):**

Use the same formatting as described in displaying employees.

**Sorting employees(Menu items 1-2)**

No output. Just updated the array containing the employees.

**Saving the database**

No output on the screen, but a new file will be created and saved.

**Exit (Menu item 8)**

Just print “Bye!”

## Restrictions and Suggestions:

* + You are not allowed to modify anything in the cmsc257-p2.c file
  + Your output must match the sample execution (including spacing and formatting).
  + Following files are needed for the project:

1. Makefile: you will create it.
2. cmsc257-p2.c : main and initialize functions. Do not modify!
3. p2-support.h : function declarations, structs and formal comments for each supporting function.
4. p2-support.c : supporting function definitions
5. database: a text file containing the database
   * The Makefile should contain commands to make your program from the source code. To compile your program, you will just need to type make. To delete the files created by make you will need to type make clean. You can read the following page to get more information about make: <https://www.gnu.org/software/make/>. You can also refer to the lecture notes (Module 5, lecture 2) or modify Makefile of project1. The executable name of your program generated after running Makefile must be p2. Otherwise it may not run on Gradescope.
   * Pay attention to the warnings when your program is compiled. You can use the -Wall option to see all warnings.
   * Make sure there are no memory leaks using Valgrind. Tool. You should at least free db struct allocated in the main at the end of your program, in addition to any other dynamic memory you allocated later.

## To turn in:

Submit the following to the Gradescope:

* A pdf file containing:
  + Sample run screenshots which should look very similar to my sample run as below with same user inputs. Inputs should be in the following order: [3 1 3 6 6 4 14 3 5 ma 7 8]
  + Known limitations of your program: this can also be a pdf file (Specifications that you could not complete, situations your program may crash)
  + Valgrind output for sample run.
* And the following files (do not zip them, upload them separately)
  + Makefile,
  + p2-support.h
  + p2-support.c
* Do not submit cmsc257- p2.c file

## Sample Output:

User inputs are highlighted. The first line runs the program with Valgrind to check memory issues.

Gradescope testing will not be limited with the functionality in the following sample output. Any of the specifications can be tested.

valgrind ./p2

==41712== Memcheck, a memory error detector

==41712== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.

==41712== Using Valgrind-3.17.0 and LibVEX; rerun with -h for copyright info

==41712== Command: ./p2

==41712==

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\* Employee Database - Total Employees: 010 \*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Choose one of the menu options below:

1. Sort by ID

2. Sort by Hire Date

3. Display all employees

4. Search employees by ID

5. Search employees by Last Name

6. Delete employee by ID

7. Save

8. Exit

**3**

No. ID Name Email Salary Hire Date

--- ----- ------------- ------------ ---------- --/--/----

0 00003 Wei Martin wm@vcu.edu $ 98999.99 01/10/2023

1 00007 Mitch Martine mitch@vcu.ed $ 78999.88 12/15/2022

2 00008 David Boyle db@vcu.edu $100000.01 11/10/2022

3 00014 Christine Blu cb@vcu.edu $ 40009.99 12/20/2022

4 00105 Stephen Black sb@vcu.edu $ 8500.01 01/09/2023

5 00001 Chris Boyle cb@vcu.edu $200000.01 01/08/2023

6 00015 Paula Brown pb@vcu.edu $ 1000.37 10/25/2022

7 00002 Paul Green pg@vcu.edu $ 2700.45 08/05/2022

8 00006 Chris Reddy cr@vcu.edu $ 23040.67 07/08/2022

9 00009 Chris Magenta cj@vcu.edu $ 23040.67 01/01/2020

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\* Employee Database - Total Employees: 010 \*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Choose one of the menu options below:

1. Sort by ID

2. Sort by Hire Date

3. Display all employees

4. Search employees by ID

5. Search employees by Last Name

6. Delete employee by ID

7. Save

8. Exit

**1**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\* Employee Database - Total Employees: 010 \*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Choose one of the menu options below:

1. Sort by ID

2. Sort by Hire Date

3. Display all employees

4. Search employees by ID

5. Search employees by Last Name

6. Delete employee by ID

7. Save

8. Exit

**3**

No. ID Name Email Salary Hire Date

--- ----- ------------- ------------ ---------- --/--/----

0 00001 Chris Boyle cb@vcu.edu $200000.01 01/08/2023

1 00002 Paul Green pg@vcu.edu $ 2700.45 08/05/2022

2 00003 Wei Martin wm@vcu.edu $ 98999.99 01/10/2023

3 00006 Chris Reddy cr@vcu.edu $ 23040.67 07/08/2022

4 00007 Mitch Martine mitch@vcu.ed $ 78999.88 12/15/2022

5 00008 David Boyle db@vcu.edu $100000.01 11/10/2022

6 00009 Chris Magenta cj@vcu.edu $ 23040.67 01/01/2020

7 00014 Christine Blu cb@vcu.edu $ 40009.99 12/20/2022

8 00015 Paula Brown pb@vcu.edu $ 1000.37 10/25/2022

9 00105 Stephen Black sb@vcu.edu $ 8500.01 01/09/2023

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\* Employee Database - Total Employees: 010 \*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Choose one of the menu options below:

1. Sort by ID

2. Sort by Hire Date

3. Display all employees

4. Search employees by ID

5. Search employees by Last Name

6. Delete employee by ID

7. Save

8. Exit

**6**

Enter the ID you are searching for

**6**

The following employee is deleted:

No. ID Name Email Salary Hire Date

--- ----- ------------- ------------ ---------- --/--/----

3 00006 Chris Reddy cr@vcu.edu $ 23040.67 07/08/2022

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\* Employee Database - Total Employees: 009 \*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Choose one of the menu options below:

1. Sort by ID

2. Sort by Hire Date

3. Display all employees

4. Search employees by ID

5. Search employees by Last Name

6. Delete employee by ID

7. Save

8. Exit

**4**

Enter the ID you are searching for

**14**

No. ID Name Email Salary Hire Date

--- ----- ------------- ------------ ---------- --/--/----

6 00014 Christine Blu cb@vcu.edu $ 40009.99 12/20/2022

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\* Employee Database - Total Employees: 009 \*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Choose one of the menu options below:

1. Sort by ID

2. Sort by Hire Date

3. Display all employees

4. Search employees by ID

5. Search employees by Last Name

6. Delete employee by ID

7. Save

8. Exit

**3**

No. ID Name Email Salary Hire Date

--- ----- ------------- ------------ ---------- --/--/----

0 00001 Chris Boyle cb@vcu.edu $200000.01 01/08/2023

1 00002 Paul Green pg@vcu.edu $ 2700.45 08/05/2022

2 00003 Wei Martin wm@vcu.edu $ 98999.99 01/10/2023

3 00007 Mitch Martine mitch@vcu.ed $ 78999.88 12/15/2022

4 00008 David Boyle db@vcu.edu $100000.01 11/10/2022

5 00009 Chris Magenta cj@vcu.edu $ 23040.67 01/01/2020

6 00014 Christine Blu cb@vcu.edu $ 40009.99 12/20/2022

7 00015 Paula Brown pb@vcu.edu $ 1000.37 10/25/2022

8 00105 Stephen Black sb@vcu.edu $ 8500.01 01/09/2023

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\* Employee Database - Total Employees: 009 \*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Choose one of the menu options below:

1. Sort by ID

2. Sort by Hire Date

3. Display all employees

4. Search employees by ID

5. Search employees by Last Name

6. Delete employee by ID

7. Save

8. Exit

**5**

Enter the Last Name you are searching for

ma

No. ID Name Email Salary Hire Date

--- ----- ------------- ------------ ---------- --/--/----

2 00003 Wei Martin wm@vcu.edu $ 98999.99 01/10/2023

3 00007 Mitch Martine mitch@vcu.ed $ 78999.88 12/15/2022

5 00009 Chris Magenta cj@vcu.edu $ 23040.67 01/01/2020

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\* Employee Database - Total Employees: 009 \*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Choose one of the menu options below:

1. Sort by ID

2. Sort by Hire Date

3. Display all employees

4. Search employees by ID

5. Search employees by Last Name

6. Delete employee by ID

7. Save

8. Exit

**7**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\* Employee Database - Total Employees: 009 \*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Choose one of the menu options below:

1. Sort by ID

2. Sort by Hire Date

3. Display all employees

4. Search employees by ID

5. Search employees by Last Name

6. Delete employee by ID

7. Save

8. Exit

**8**

Bye!

==54820==

==54820== HEAP SUMMARY:

==54820== in use at exit: 0 bytes in 0 blocks

==54820== total heap usage: 4 allocs, 4 frees, 8,352 bytes allocated

==54820==

==54820== All heap blocks were freed -- no leaks are possible

==54820==