

CSc 3320: Systems Programming

Fall 2021

Midterm 2: Total points = 200

Assigned: 17th Nov 2021

Submission Deadline: 6th Dec 2021, Monday, 11.59 PM (No extensions. If your submission is not received by this time then it will NOT be accepted.)

Submission instructions:

1. Create a Google doc for your submission.
2. Start your responses from page 2 of the document and copy these instructions on page 1.
3. Fill in your name, campus ID and panther # in the fields provided. If this information is missing TWO POINTS WILL BE DEDUCTED.
4. Keep this page 1 intact. If this *submissions instructions* page is missing in your submission TWO POINTS WILL BE DEDUCTED.
5. Start your responses to each QUESTION on a new page.
 6. If you are being asked to write code, copy the code into a separate txt file and submit that as well. The code should be executable. E.g. if asked for a C script then provide myfile.c so that we can execute that script. In your answer to the specific question, provide the steps on how to execute your file (like a ReadMe).
7. Provide the evidence of your outputs through a screenshot and/or screen video-recordings and copy the same into the document.
8. Upon completion, download a .PDF version of the google doc document and submit the same along with all the supplementary files (videos, pictures, scripts etc).

Full Name:

Campus ID:

Panther #:

READ THESE NOTES BEFORE YOU START!

- Questions 1-4 are 30pts each.
- Questions 5 and 6 are 40 pts each.
- All questions **MUST** be **ATTEMPTED**. Your **MIDTERM 2** will **NOT** be evaluated if there is **NO ATTEMPT** for even 1 question.
- All programs have to be well commented. Non-commented programs will receive 0 points. Comments have to be comprehensible and concise.

1. Consider the array given below. Write a C program that must be able to sort the elements in the array. You must use pointers in your code to work with the arrays. The sort functionality must be implemented as a separate function named "sort_numeric()"

Array for your evaluation

[10, 0.25, -2342, 12123, 3.145435, 6, 6, 5.999, -2, -5, -109.56]

If given user input A or a: sort in Ascending order

If given user input D or d: sort in Descending order

2. Consider the list of names given below. Write a C program that will first create a string array that will contain this list and then sort the elements in the array as per alphabetical order. You must use pointers in your code to work with the arrays. The sort functionality must be implemented as a separate function named "sort_alphabetic()". The program can be case insensitive (i.e. capital or small letters are treated the same).

List for your evaluation

Systems

Programming

Deep

Learning

Internet

Things

Robotics

Course

If given user input A or a: sort in alphabetical order (a comes first) If

given user input D or d: sort in reverse alphabetical order(z comes first)

3. Repeat Question 1 or Question 2, considering that the number of elements can potentially increase. That is, the size of the array will be unknown at the start of the program. Note that the requirement of using pointers still holds.

Show proof of evaluation of your program being able to work for more than 10 entries. Show 5 evaluation trials in your submission. You can pick any number of entries between 10 and 30 for your trials.

(Hint: To solve this, use dynamic memory allocation, where you will NOT treat the input array as a known or finite size. Allocate memory space (e.g. malloc()) as and when the number of elements in the list increases).

4. Using C programming and using Structures or Unions in your program, build a COVID vaccine registration form where any user can register by filling in their First Name, Last Name, Date of Birth (mm/dd/yyyy), Dose number (1 or 2), Date of previous dose, Type of vaccine (e.g. Pfizer, Moderna, Johnson & Johnson etc.), Residential zip code.

Upon registration, the system must output a 8 letter alphanumeric code that will be unique to that user. The code is generated as <First letter of First Name><First Letter of Last Name><current age of user -as of registration date><First letter of Vaccine type><last 3 numbers of zipcode>

Add functionality in your program such that it will display all the user's information on the screen (one item in each line).

Show an evaluation trial for registering at least 10 users. For registration, for relevant questions, users must choose values based on the options provided. Use pseudo values instead of actual personal details.

(Hint: Write a program that contains main(), register(), generate_code() and retrieve() functions, at the least).

5. Copy the contents of this document into a text file. Make sure the spacings and indentations are included. Write a C program that READS the TEXT file and then outputs
 - a. the number of characters (space is to be considered a character), b. number of words (a word is any sequence of non-white-space characters)
 - c. number of lines.

-- Each of the functionalities a, b, and c above must be written as FUNCTIONS and passing of arguments MUST be through POINTERS. -- Name the functions problem5char.c, problem5words.c, and problem5lines.c

-- Write a Makefile that will execute the main C program to include all these three scripts.

-- All these outputs from (number of chars, words and lines) must be saved into ANOTHER text file row-wise. Every execution of your script with a new input must APPEND the outputs to a new row in that text file. You can separate each value in a row using any delimiter of your choice (e.g. comma or semi-colon etc)

6. Repeat everything in Problem 5, but create the functionalities a, b and c as HEADER (Library) files and execute using a Make file.