CSE 240 Spring 2019 Homework 5: Array of Structs and Enum Types (50 points)

Due Saturday, February 16, 2019 at 11:59PM, plus a 24-Hour grace period

Introduction

The aim of this assignment is to make sure that you understand and are familiar with the concepts covered in the lectures, including basic C syntax, using structures and enumeration types. By the end of the assignment, you should have

- understood the concepts and operations of enumeration type, arrays, structures, and array of structures, and files.
- written a program using enumeration type and an array of structures.

Reading: Textbook Chapter 2, sections 2.5, and 2.6, and lecture slides covered.

Exercising: Complete the multiple choice questions in Textbook Section 2.10. The answers of the questions are available in course Web page.

You are expected to do the majority of the assignment outside the class meetings. Should you need assistance, or have questions about the assignment, please contact the instructor or the TA during their office hours.

You are encouraged to ask and answer questions on the course discussion board. However, do not share your answers or code in the course discussion board. **Do not cooperate with your peers in doing the individual assignments.**

Programming Assignment (50 points)

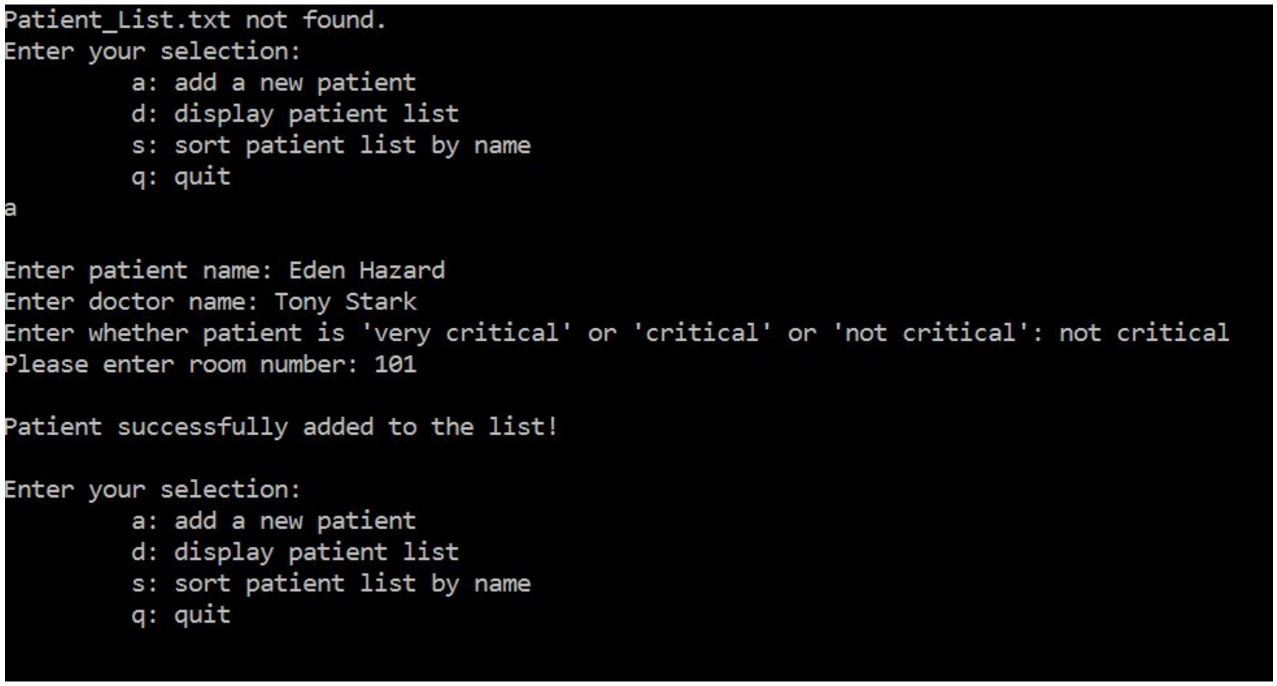
1. You are given a partially completed program hw05q1.c. You should follow the instructions given in the program to complete the functions so that the program executes as instructed. The program declares a struct 'patientRecord' with elements for patient's name, doctor's name, critical level of patient, room number. You will be completing a program that creates a list of patients (array of structures). It is a menu-driven program where the user is given the following options:
   a) Add a new patient to the list. When adding a new patient to the list, the user is prompted for patient's name, doctor's name, critical level of patient and room number of the patient. The patient should be added at the end of the list. If the patient already exists in the list, then you should not add to the list. The critical level is enum type.
   b) Display the list of patients. This should display each patient's details one after the other.

c) Sort the list of patients alphabetically by patient name. The sorting should happen within the list. You should not create a new list (array of structs) of patients having sorted patients.

There is save() already implemented to write the patients list to a file ‘Patient_List.txt’. save() is executed at the end of the program when the user quits the program. You need to implement load() which is called at the start of the program. This function will read the saved file and fill in the array of structures. You need to read the file in the same order and manner that it is saved in save().

Expected output of each function:

<u>add:</u>

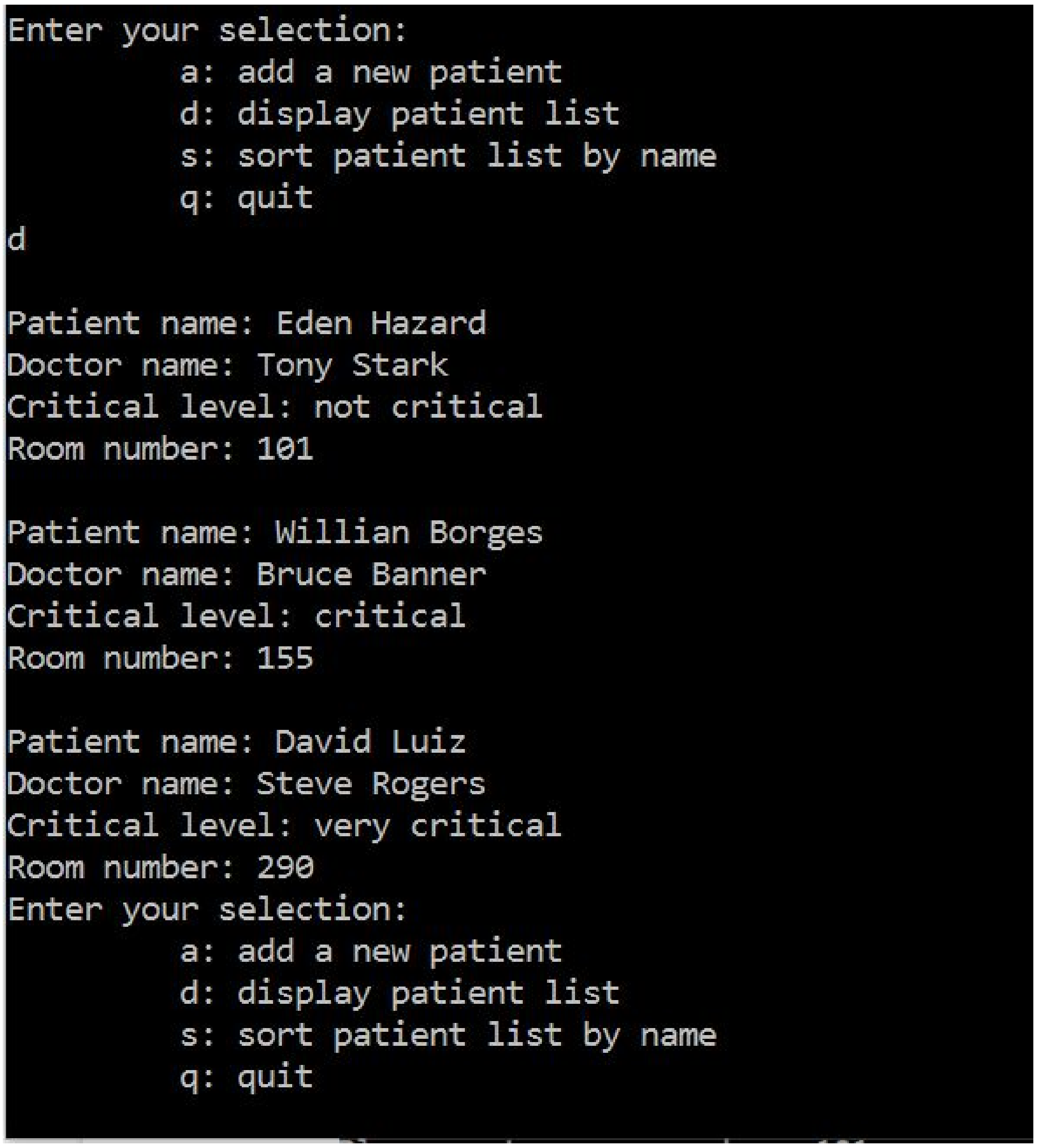


```
Patient_List.txt not found.
Enter your selection:
        a: add a new patient
        d: display patient list
        s: sort patient list by name
        q: quit
a

Enter patient name: Eden Hazard
Enter doctor name: Tony Stark
Enter whether patient is 'very critical' or 'critical' or 'not critical': not critical
Please enter room number: 101

Patient successfully added to the list!

Enter your selection:
        a: add a new patient
        d: display patient list
        s: sort patient list by name
        q: quit
```

Notice that since this is the first execution of the program, load() gave message “Patient_List.txt not found”. This file is created at the end of the program by save().

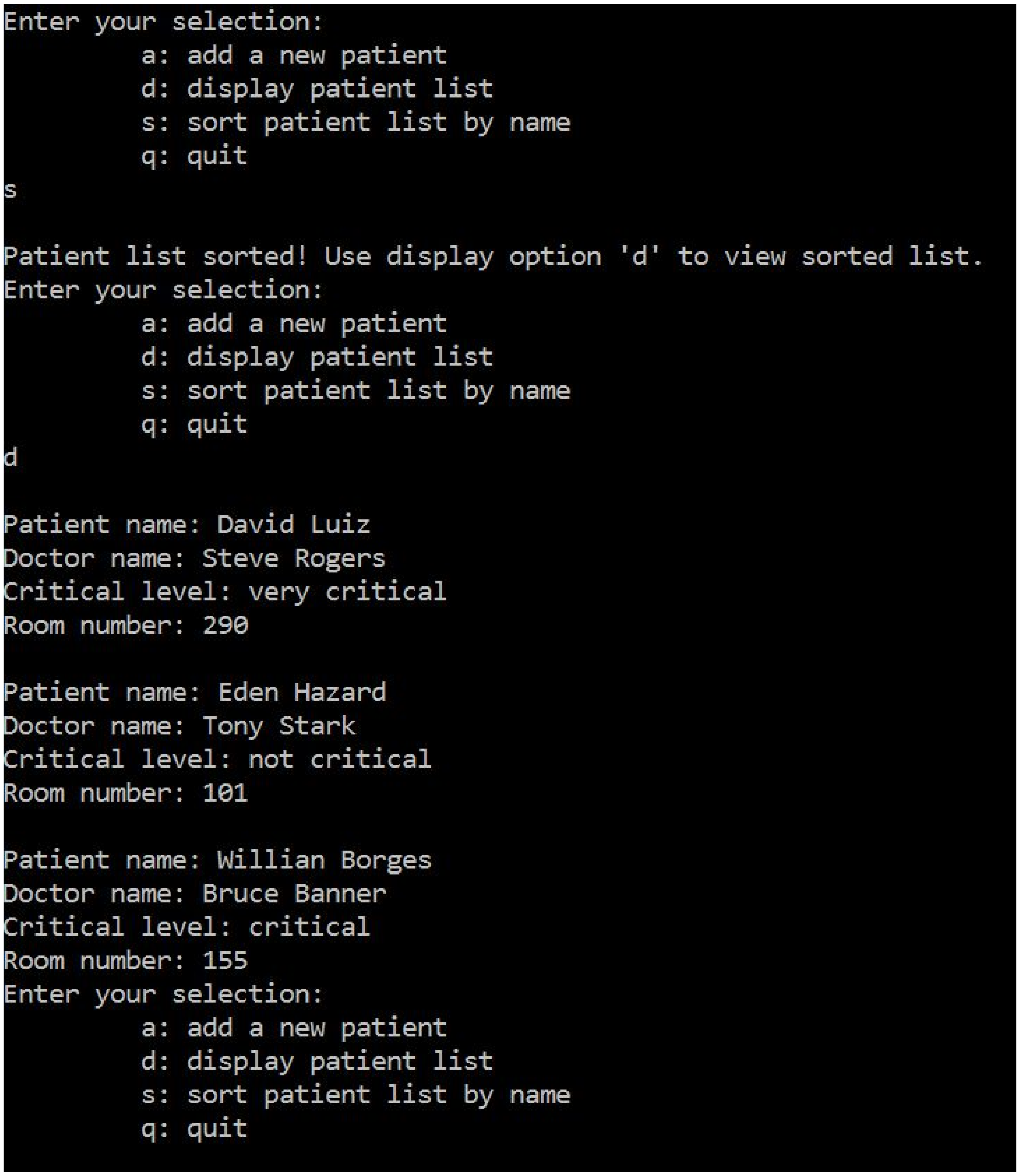
display:

(after adding 3 patient details)



```
Enter your selection:
         a: add a new patient
         d: display patient list
         s: sort patient list by name
         q: quit
d

Patient name: Eden Hazard
Doctor name: Tony Stark
Critical level: not critical
Room number: 101

Patient name: Willian Borges
Doctor name: Bruce Banner
Critical level: critical
Room number: 155

Patient name: David Luiz
Doctor name: Steve Rogers
Critical level: very critical
Room number: 290
Enter your selection:
         a: add a new patient
         d: display patient list
         s: sort patient list by name
         q: quit
```

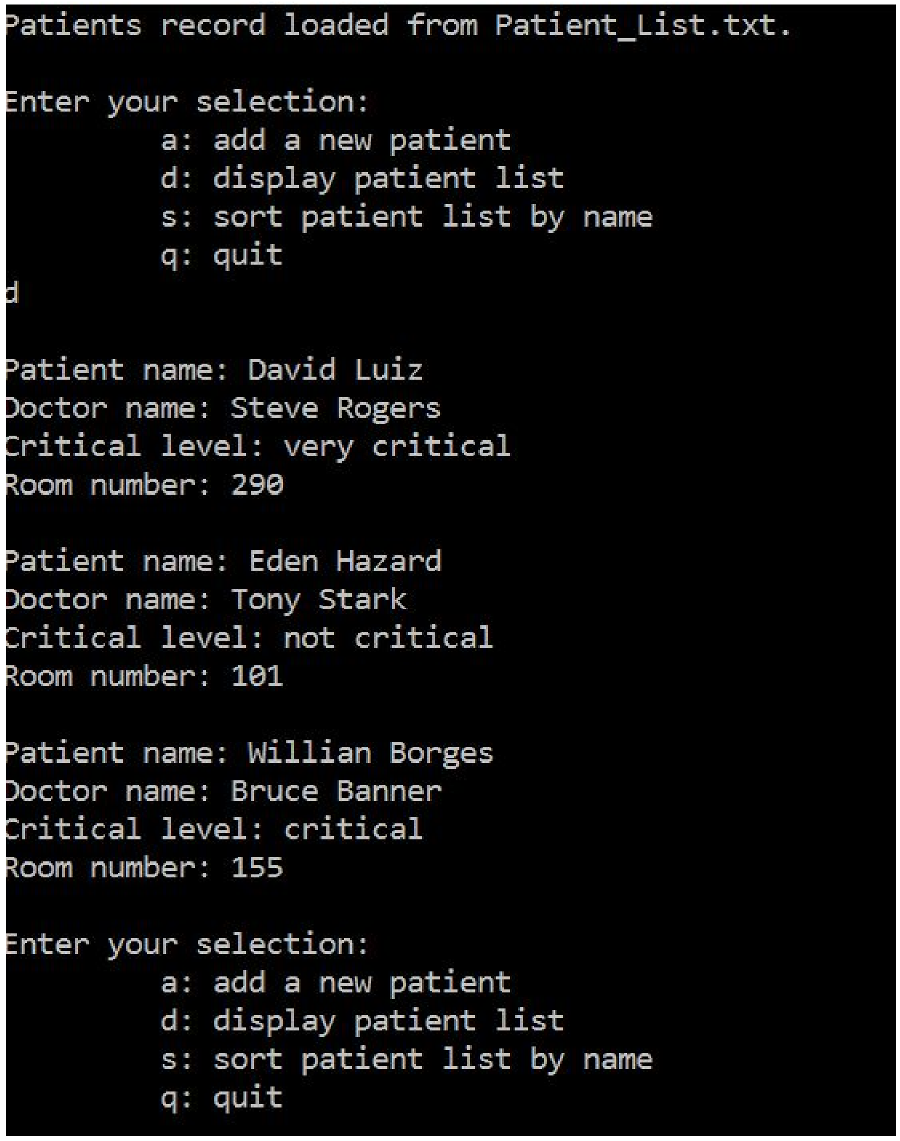
sort:



```
Enter your selection:
         a: add a new patient
         d: display patient list
         s: sort patient list by name
         q: quit
s

Patient list sorted! Use display option 'd' to view sorted list.
Enter your selection:
         a: add a new patient
         d: display patient list
         s: sort patient list by name
         q: quit
d

Patient name: David Luiz
Doctor name: Steve Rogers
Critical level: very critical
Room number: 290

Patient name: Eden Hazard
Doctor name: Tony Stark
Critical level: not critical
Room number: 101

Patient name: Willian Borges
Doctor name: Bruce Banner
Critical level: critical
Room number: 155
Enter your selection:
         a: add a new patient
         d: display patient list
         s: sort patient list by name
         q: quit
```

The 3 patients seen in display() output above are sorted in sort(). Use ‘d’ option to verify sorted result.

load:



```
Patients record loaded from Patient_List.txt.

Enter your selection:
        a: add a new patient
        d: display patient list
        s: sort patient list by name
        q: quit
d

Patient name: David Luiz
Doctor name: Steve Rogers
Critical level: very critical
Room number: 290

Patient name: Eden Hazard
Doctor name: Tony Stark
Critical level: not critical
Room number: 101

Patient name: Willian Borges
Doctor name: Bruce Banner
Critical level: critical
Room number: 155

Enter your selection:
        a: add a new patient
        d: display patient list
        s: sort patient list by name
        q: quit
```

Notice the message given by load() “Patients record loaded from Patient_List.txt” at the top. To verify that load() worked as expected, use ‘d’ display option to display loaded list.

Grading of Programming Assignment

The TA will grade your program following these steps:

(1) Compile the code. If it does not compile, 20% of the points given will be deducted. For example, if there are 20 points possible, you will earn 16 points if the program fails to compile.

(2) The TA will read your program and give points based on the points allocated to each component, the readability of your code (organization of the code and comments), logic, inclusion of the required functions, and correctness of the implementation of each function.

What to Submit?

You are required to submit your solution in a compressed format (.zip). Make sure your compressed file is label correctly - lastname_firstname5.zip. (All lowercase, do not put anything else in the name like "hw5".)

The compressed file MUST contain the following:
hw05q1.c (completed code)

No other files should be in the compressed folder.

If multiple submissions are made, the most recent submission will be graded. (Even if the assignment is submitted late.)

Where to Submit?

All submissions must be electronically submitted to the respected homework link in the course web page where you downloaded the assignment.

Late submission deduction policy

- No penalty for late submissions that are received within 24 hours after the deadline;
- 10% grade deduction for every day it is late after the grace period;
- No late submission after Tuesday at 11:59PM.

Please read the FAQ file in the Course Information folder:

Q: For some reason, my assignment submission did not go through, but I thought it went through. I can show you on my local disk or in my Dropbox that I completed the assignment before the due date. Can my assignment be graded?

A: You should always download your own submission from the blackboard after submission and test if the submission contains all the required files. We will grade the assignment submitted into the Blackboard only. We cannot grade the assignment sent from email or stored in any other places, regardless its last-modified-time. If you submitted your assignment into the blackboard, it cannot be downloaded from the instructor side, but it can download from your side, we can download from your blackboard and grade the assignment. Please meet the instructor or TA in this case.