BMIF 801 – Mini-Project

Due: 11:59pm Friday, June 10th, 2022

This project will be done in groups of 3. Groups will be assigned during the first lecture. This mini-project has three parts, so your code should be submitted as three Python files. Name your files patient_management_system.py, simulator.py and system_plots.py and upload them to the assignment dropbox on OnQ. As this is a group project, you should also submit a PDF documenting how the work was divided. Each person must contribute to the coding portion in some significant way. You may upload multiple times - we will mark the last submission prior to the deadline. Please check your files after uploading.

Your program should be documented appropriately with a multi-line comment at the top of the file that indicates what the program does and who it was written by. Include the names and student numbers of all group members. In-line comments should appear in the code to explain lines that are not self-explanatory. Use white-space to make your code more readable. Your code for each question in this assignment should be properly structured into functions (including a main() function) to minimize duplicated code where possible.

Patient Management System (10)

A major challenge throughout the Covid-19 pandemic was managing hospital resources, especially ICU capacities. Across Ontario as ICUs fill up in hot zones hospitals began transferring patients to other hospitals that had available ICU beds. For this section, your job is to create a simple patient management system for a 3-hospital system. The names of the hospital and their (fictional) ICU capacities is listed in the table below:

Hospital Name	**Total # ICU Beds**
Kingston	8
Ottawa	17
Toronto	20

In this system, as patients are admitted to the ICU they are assigned a unique ID, a "Severity status" from 0 to 3, and it is noted whether or not they are positive for Covid-19. Each ID should contain 3 numbers and 1 letter and must be unique to a single patient across all hospitals (no duplicates). A list of all patients currently in the hospital system can be found in **initial_hospital_state.csv**. Your program must read in this list of patients upon starting the system. Your system should allow the user to select one of four actions: Add, Transfer, Update, and Discharge. Add allows the user to admit a new patient to the hospital system. Transfer allows the user to move a patient from one hospital to another that has open beds. Update will allow the user to modify the severity status of a patient. Finally, Discharge will remove the patient from the hospital system. After any action, the system should save the new state of the hospital system to a csv file called **final_hospital_state.csv**.

Your system should enforce the following restrictions:

- Patients cannot be transferred to a hospital with no available ICU beds. In this case the system should display the list of hospitals and the number of beds available and prompt the user to select a different hospital.

- Patients with a severity status of 3 are not stable enough to be transferred. The system should suggest the first patient in the current hospital with the lowest severity status. If there are no patients with a status <3, the system should inform the user that no transfers are possible.
- Patients may only be discharged if they have a severity status of 0.

Your system should allow the user to continue making actions until they inform the system that they are finished by typing "Exit".

Specific Coding Guidelines:

Your code may include as many functions as you need but it **must** include the following 4 functions:

1. addPatient(hospitalName, sevStatus, covidPositive):
   - Allows the user to admit a patient to a hospital, assigns them a unique patient ID
   - If the hospital's ICU is full, the system should inform the user that the ICU is full and they must transfer a patient or admit them to a different hospital, then the system should return to the main menu

2. transferPatient(patientID, newHosptialName):
   - Allows the user to transfer a patient from one hospital to another

3. dischargePatient(patientID):
   - Removes a patient from the hospital system

4. updateStatus(patientID, newStatus):
   - From the main menu if the user types 'exit' the program should terminate

Your code must also account for any user errors in inputs, it should not throw any errors based on incorrect user input. The system must also be as user friendly as possible. All outputs should be neatly formatted with whitespace for clarity and all prompts for input should include clear instructions to the user.

Simulate Data (10)
Medical data is often very difficult to come by. In order to meet deadlines, computer scientists will often simulate data to test their programs, before receiving access to real data. For this section, your task is to simulate the use of the patient management system in part 1 in order to observe trends in hospital occupation and case severity. To simplify the simulation, your code should make use of the following assumptions:

- All hospitals begin with an empty ICU (initial_hospital_state.csv should be empty)
- Each hospital only admits one patient to the ICU every 7 days, except Toronto which admits 1 every 4 days
- Patients are automatically discharged when their severity status reaches 0

- It takes 5 days for a patient's severity status to drop from 3 to 2, 3 days for it to drop from 2 to 1, and only 1 day for it to drop to 0. (we're assuming no one dies in this fictional scenario).
    - We also assume that everyone recovers and no one gets worse
- If a patient must be transferred, they are transferred to the hospital with the most available ICU beds.
- The probability that a patient comes in with Covid starts at 10% and increases exponentially by 5% every day
    - E.g. the second day should be 10.5%, tenth day should be 16.3%, etc
- We can also assume that no one contracts Covid while in hospital
- If a patient does not have covid, the probability that they have a severity status upon admission is 3 is 10%, there's a 30% probability that their severity status is 2, and a 60% chance that their severity status is 1. No one is admitted with a severity status < 1.
- If a patient does have covid the probability of each severity status at time of admission is as follows:
    - 30% severity status of 3
    - 30% severity status of 2
    - 40% severity status of 1

Your code should simulate the use of your system for 90 days. For every 10 simulated days, save the state of the hospital to a separate csv file called hospital_state_<Day #>.csv (e.g. 1st day would be hospital_state_01.csv). These CSV files will be used in part 3.

System Plots (5)
The purpose of this section is to create figures that can be used to monitor the Covid situation within the hospital system. Your code for this section must read the state of the hospital system from the series of csv files that you generated in part 2. Your code should produce 2 plots based on the data:

1. Hospital Occupancy by Covid patients:
    - Create a single bar chart that shows the number of patients with Covid in each of the 3 hospitals in each of the states generated in part 2.
2. Severity of cases
    - Create a scatter plot that maps the severity status of all patients in each of the saved states. Your figure should contain 9 time points on the x axis, and severity status of each patient on the y-axis. Each hospital should be plotted in a different colour.

Each of these plots must be properly formatted to include a title, axis titles, and a legend.

What to Submit
Write answers each in a separate Python file. Call your files patient_management_system.py, simulator.py and system_plots.py corresponding to the three tasks described above. Upload each file to the assignment dropbox. Please check your files after uploading.

Extensions
You may hand in assignments up to 12 hours after the deadline with a penalty that is built into the rubric (up to: 4 hours late -20%, 6 hours late -40%, 8 hours -60%, 12 hours -75%). Do not request an extension unless you have a very good reason. Assignments submitted after solutions have been

presented or marks have been returned will not be graded.

Academic Integrity

This project is meant to be completed by individual groups. A reminder of the Policy on Academic Integrity in this course. Findings of departure from Academic Integrity result in an automatic F in the course