Assignment 3

Instructions (READ CAREFULLY!!!)

1. Follow the directory format provided. Assignments that violate the directory format will be considered as not having been submitted. The directory can be downloaded here (https://tinyurl.com/sl8wqpa). **Remember to change the name of root folder to your student ID and to re-zip into a zip folder named your student ID!**
2. Complete this assignment in one of either Python 3.7, C, C++, or Java. If using Java, do not use packages as they would violate the directory format and complicate command-line compilation. Also, if using Java, ensure that your file paths for input and output files are not specific for your machine; submissions with this problem will be awarded 0. You should name your entry file main.⟨ext⟩ where ⟨ext⟩ is the appropriate file extension for your choosen language; for example, if you used Python, your file should be named main.py
3. **REMEMBER THAT YOU ARE BEING MARKED ON TEST CASES (PER SLIDE# 22 OF LECTURE 1). HENCE CODE THAT DOES NOT COMPILE, PRODUCE THE REQUIRED OUTPUT FILES, CRASHES DURING EXECUTION, ETC... WILL BE AWARDED A MARK OF 0**
4. Ensure that you edit the JSON file in the root of the directory with your name, Student ID, programming language used, and Email address.
5. Sign and attach either a soft copy or clear image of the plagiarism declaration with your submissions. There is a folder in the directory for you to store this declaration. You are allowed to discuss the assignment with your classmates; however, your code and write-up MUST be your own. You are free to use the sample code provided on the course Github repo.
6. After unzipping the directory structure, edit with the confines of that directory structure and then places it back into a .zip file with your

Student ID. Other compression formats such as .rar or .7zip, or archives not named your Student ID will be rejected.

7. After completion of your assignment, fill out **this form (located at https://forms.gle/PAYqarM9M9CKVPoD7)** and email a copy of the assignemnt to irahamancourses@gmail.com. Your email should have the subject COMP2611 A2 1234 where 1234 is your student ID number. YOU MUST DOWNLOAD THIS PDF TO CLICK THE LINK TO THE FORM!

8. You are free to use any in-built data type in Python, C++, or Java (sets, dictionaries, vectors, etc....).

9. - Draft Date: 21st November 2019
   - Final Hard deadline: 28th November 2019

Assignment 3

Part A - Patient Wait-time

A hospital wants to conduct a study on the wait time of patients in a hospital. In a hospital, the wait time is the amount of time a patient must wait to see a doctor, i.e. the time elapsed from the patient's entry into the hospital to when they are seen by a doctor.

Unlike many other systems, hospitals do not always operate on a first come first serve basis; they serve in order of most priority. When a doctor becomes available, they see the patient with the highest priority.

You are given a file named *data.txt.* Each line in this file contains an event. An event is either:

- A patient enters the hospital. Such events have the event code P and also store the patient's time of arrival and their priority score. For example, the line 'P 5 79" says that a patient entered at time $t = 5$, and with a priority of 79.
- A doctor tends to the highest priority patient at a particular time. Such events have the event code D and store the time of the doctor tending to a patient. For example, the line 'D 19' says that a doctor served the most critical patient at time $t = 19$

Events in *data.txt* are listed in chronological order, and you may assume that every patient is seen by a doctor at some point. You are to write a programme that when given the events listed in *data.txt* outputs and stores the average wait-time in a file named *solution.txt.*

Consult the *data.txt* and *solution.txt* provided for further details.

NB:

You may assume that no more than 200 events are stored in the *data.txt.* You may also assume that no more than 100 patients in are in the system at any point in time, and that a patient's priority is between 0 to 100 (inclusive)