Assignment #1

Problem Solving and Programming in C++
Department of Computer Science
Old Dominion University

Objective:

The main objectives of this assignment are:

- train you to apply some of the struct concepts which you learned in the class
- review some of the concepts related to functions in C++

Problem description:

A hospital needs an efficient way to handle Personnel information. Given the large number of Personnel, it has become very difficult to manually track information regarding current and past Personnel. This has led the hospital to hire you to program a solution.

The goal of the program is to read Hospital Personnel records from a file (**hospitalPersonnel.txt** – *which is provided*). Each line in the file is a Personnel record. Each record is best represented by a `struct`, so all records must be read into an array of `structs`.

Each line consists of the following fields separated by spaces:

For Medical Personnel:

> *FirstName LastName ID Role* (***MD*** or ***NP***) *DutyDays* (*3 Initials for the days in a week*)
>
> ***Example:*** **Alexander Desplat 000000000 MD M W F**
>
> ***Example:*** **Craig Armstrong 000000007 NP T TH F**

Please note that the two abbreviations "**MD**" and "**NP**" in the role filed stand for medical doctor and nurse practitioner.

For Patients Personnel:

> *FirstName LastName ID Role ExitFlag* (**Y** or **N**) *AdmitDate* (**MM**:**DD**:**YYYY**) *OptionalExitDate* (MM:DD:YYYY) - If the *ExitFlag* is Y (Yes) this means there exists an *OptionalExitDate*
>
> For example: **Michael Giachino 000000008 PT Y 09:11:2013 10:11:2013**

For example: **Bruno Coulais 000000001 PT N 08:04:2015**

After reading the records from a file, the program should present a menu which handles the following requests:

Press 0 to print all Hospital Personnel

Press 1 to print only Patients

Press 2 to print only Doctors

Press 3 to print only Nurse Practitioners

Press 4 to print only Admitted Patients

Press 5 to print only non-Admitted Patients

Press 6 to exit:

Note:

1. Dates printed on the screen must be in the format: MM:DD:YY
2. Print the full day name instead of the initials. For example, if the ***DutyDays*** for a doctor is "M W F". Do not print "M W F," instead print "Mondays-Wednesdays-Fridays". Here is the complete list of full day names:
    - **M** - Mondays, **T** - Tuesdays, **W** - Wednesdays, **TH** - Thursdays, **F** - Fridays, **SA** - Saturdays, **S** – Sunday.

The `main()` function of your program should be very simple. The main function should be a collection of variables declaration and functions call. You will need to use different functions to read the data from the **hospitalPersonnel.txt**, print the menu, print Hospital Personnel fulfilling a user specified criteria. This means your program must consist of at least 3 functions (excluding the main function).

Hints:

- Do not use global variables; instead all information must be passed as parameters to functions.
- Some parameters may be passed by reference

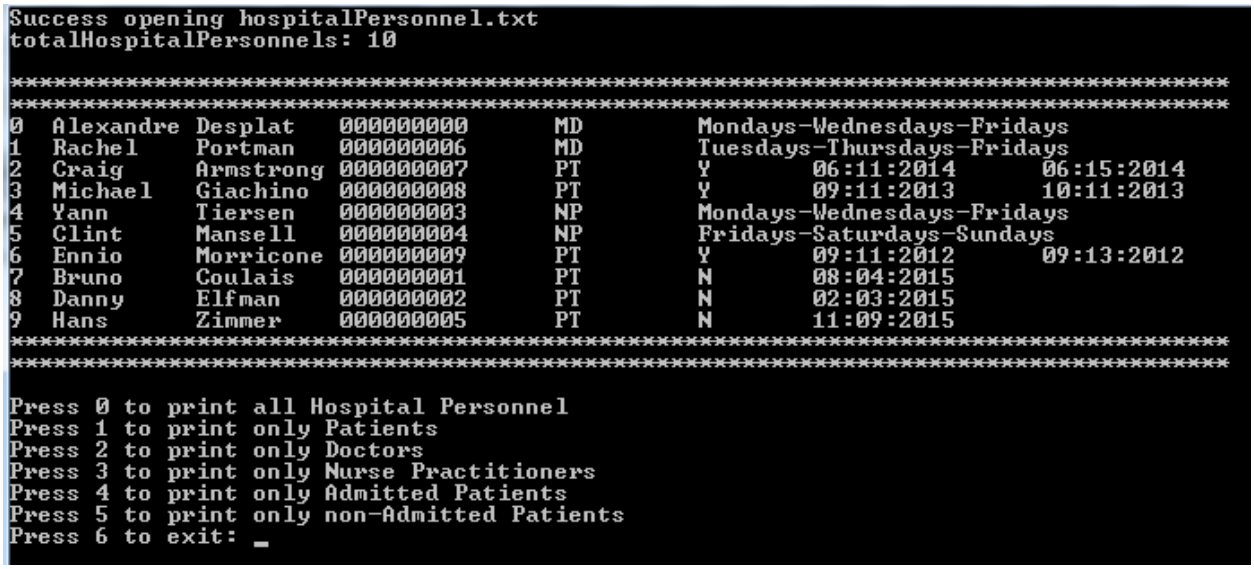
- Please note that some sub-fields like Date item are best represented by struct data type. This means the struct data type representing Hospital Personnel must have a struct representing the **AdmitDate** and **ExitDate** inside it (a struct inside a struct)

- Your program should read the required data from file: **hospitalPersonnel.txt**

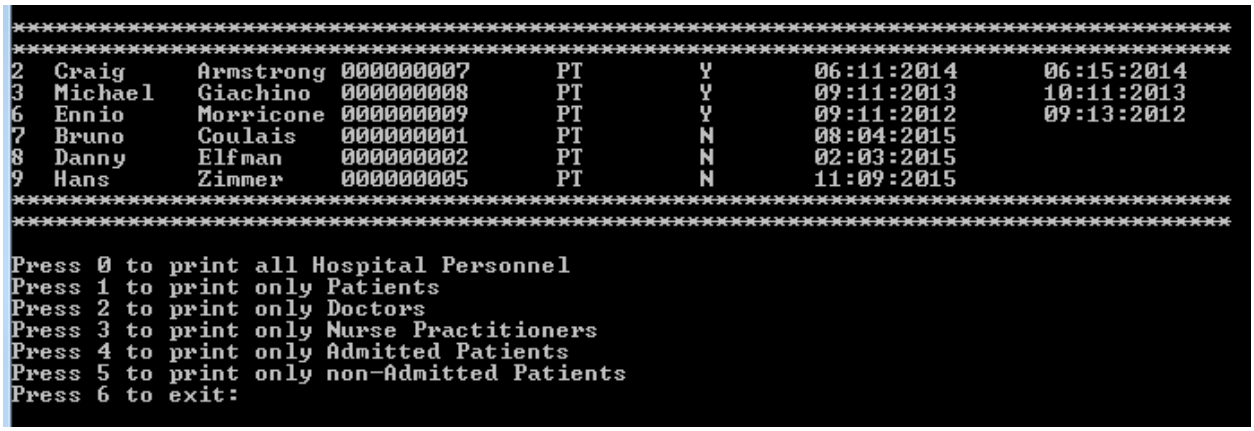
Submission notes:

- Submit all files from your project(s), especially the .cpp, .h, and .cbp file(s)
- Zip the .cpp and name it as "Assg1_cslogin", where the cslogin is your login ID for the computers at the Department of Computer Science at ODU.
- Submit the zipped file to the respective Blackboard link.

Sample output: No selection (initial screen)



Sample output: item 1 section



Sample output: item 2 section

Sample output: item 4 section