

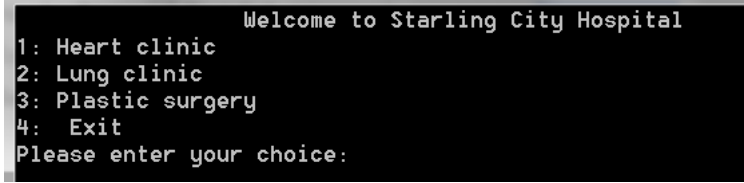
- In order to receive full credit, your program must be running correctly, implements all required functions/data and organized.
- Maximum possible points 100

Starling City Hospital

Assignment learning objectives: You are supposed to correctly define and write neat code that uses queues.

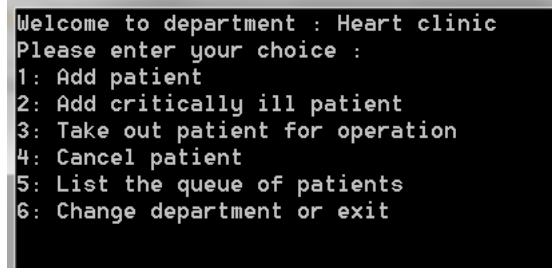
The assignment problem:

For the last assignment of the semester, you need to develop a simple patient's management system for the hospital Starling City Hospital. The hospital has 3 departments: Heart clinic, Lung clinic, and Plastic surgery. The first screen of your application will ask the user to select one of the departments (see figure 1). You need to create a queue for each department that includes patients of the current department only. Users of your software can do the following functions within each department (see figure 2):



```
                Welcome to Starling City Hospital
1: Heart clinic
2: Lung clinic
3: Plastic surgery
4: Exit
Please enter your choice:
```

Figure 1. The application's main screen



```
Welcome to department : Heart clinic
Please enter your choice :
1: Add patient
2: Add critically ill patient
3: Take out patient for operation
4: Cancel patient
5: List the queue of patients
6: Change department or exit
```

Figure 2. The screen of each department

- **Add new patient**

The “*Add patient*” is responsible to add a new patient entry into the queue. Each patient must have a first name, last name, and a social security number (cannot be empty, but can be of any length for simplicity, e.g. 123 is accepted). If the social security number is empty (return character), the operation should be canceled with cancelation message. The maximum number of patients in any department is 100. Note that after adding a new patient, your application should confirm the entered data of the patient on a *new screen*. After confirming the entered data, the user can press any key to return to the department's menu. You do not need to ask the user if the entered data is correct or not. Just confirm it by re-printing it.

- **Add critically ill patient (emergency cases)**

The “*Add critically ill patient*” will add a new patient (first name, last name, social security number) at the beginning of the current queue –passing all current patients in the queue. This means that patients entered the queue as critically ill will have operation before other patients, even if they entered the queue after the regular patients. Your application can add any number of patients as critically ill, but not more than 100 patients in total within each department, both regular and critically ill patients. Note that, if you have several critically ill patients then they have operations in the same order they entered the queue (First in, first out). Once all critically ill patients have been taken out to operation (removed from the queue), the application continues with the regular patients –also in a first in, first out manner. Note that after adding a new patient, your application should confirm the entered data of the patient on a *new screen*. After confirming the entered data, the user can press any key to return to the department’s menu. You do not need to ask the user if the entered data is correct or not. Just confirm it by re-printing it.

- **Take out patient for operation**

This feature is responsible to move patients to the operation room (remove them from the queue). Of course, this should be a first-come first-served operation. Note that having critically ill patients will give them higher priority to enter the operations room (to be removed from the queue first, even if they have been added after the regular patients). After moving a patient to the operation room, your application should print the data of the moved patient (name and social security number) in a new screen. If no patients are left, your application should print a notification message to the screen “No more patients.”

- **Cancel patient**

This feature will ask the user to input the data of a patient (first name, last name, and social security number) and then remove that patient from the queue. If such a patient does not exist, a notification message is printed to the screen “Patient Does Not Exist.”

- **List all current patients**

This feature lists all patients currently in the queue (first name, last name, social security number) including regular and critically ill patients.

- **Exit the current department**

This function exists the current department, clear the screen, and print the main application menu (Figure 1).

You need to define a struct called *patient* to define patients using first name, last name, and Social Security number. In addition, your application must have a class called *queue* that includes the aforementioned functions (methods) as member functions of the class such as, `AddPatient()`; `AddPatientatEnd()`; `AddPatientatBeginning()`; `CancelPatient()`; etc. Feel free to create your own methods –these are just examples.

Extra credit (5 possible points):

You can improve the application to prevent data duplication. For example, your application should not accept two patients with the same social security number.

Remember:

- **Build your code incrementally.**
- **Zip your project and upload it to Canvas before December 12th.**

~BestOfLuck()