Title: Hospital Management System using Priority Queue in C

Introduction:

* Hospital Management System is a computerized system that is designed to manage the hospital's different operations such as managing patient records, managing appointments, managing inventory, managing billing, and managing the overall hospital workflow.
* Priority Queue is a type of abstract data type that allows the elements to be inserted and removed in a specific order based on their priority. The elements with higher priority are dequeued first.

Objective:

* To implement a Hospital Management System using Priority Queue in C language.
* The program should allow the user to add patients to the queue, remove patients from the queue, and display the patient list in the order of their priority.
* The priority of the patients should be determined by their severity of illness or medical condition.

Code Overview:

* The program starts by defining the maximum size of the priority queue.
* Two structures are defined: Patient and PriorityQueue.
* The Patient structure contains two fields: patient\_id and priority, which represent the patient's identification and priority level, respectively.
* The PriorityQueue structure contains an array of Patient type data and a rear field that represents the index of the last element in the array.
* The init function initializes the queue by setting rear to -1.
* The is\_empty function checks if the queue is empty by checking if the rear index is -1.
* The is\_full function checks if the queue is full by checking if the rear index is equal to the maximum queue size - 1.
* The enqueue function adds a patient to the queue based on their priority level. It first checks if the queue is full and then finds the correct position for the patient in the queue based on their priority level.
* The dequeue function removes the patient with the highest priority from the queue and returns it. It first checks if the queue is empty and then moves all the elements in the array one position to the left to fill the gap left by the dequeued element.
* The display function displays the list of patients in the queue along with their patient ID and priority level.

Execution:

* The program starts by initializing an empty priority queue.
* The program then displays a menu to the user with four options: add patient, remove patient, display patient list, and exit.
* If the user chooses the add patient option, the program prompts the user to enter the patient's ID and priority level and adds the patient to the queue.
* If the user chooses the remove patient option, the program removes the patient with the highest priority from the queue and displays a message with the patient's ID and priority level.
* If the user chooses the display patient list option, the program displays the list of patients in the queue along with their patient ID and priority level.
* If the user chooses the exit option, the program terminates.

Conclusion:

* In conclusion, the Hospital Management System using Priority Queue in C language is an efficient way to manage the hospital workflow by prioritizing patients based on their severity of illness or medical condition.
* The program is designed to allow the user to add patients to the queue, remove patients from the queue, and display the patient list in the order of their priority.
* This program can be further extended to include more features such as managing patient records, managing appointments, managing inventory, and managing billing.