TASK4 - REPORT

In this task, we make a hospital system by using previous tasks codes.

Task 1, Task 2 and Task 3 codes are connected together in this task.

Main purpose is show how data structures work together.

Firstly, patients are stored in linked list.

This linked list comes from Task 1.

Linked list is good because patient number is not fixed and adding patient is easy.

Patients are hold as Patient objects inside PatientList.

For treatment requests, queue structure is used.

This queue is implemented in Task 2.

Queue works with FIFO, so patient who comes first treated first.

In Task 4, there is also priority patients.

Because of this, two queues are used.

One is for priority and one is normal queue.

Priority patients always treated before normal ones.

Enqueue and dequeue works in O(1) time.

Discharged patients are stored in stack structure.

Stack is from Task 3.

Stack works with LIFO rule.

Last discharged patient stays on top of stack.

Push and pop operations are also O(1).

In hospital system, HashMap is used for find patient fast with id.

This helps to reach patient in O(1) average time.

So system becomes more faster.

Patients can be sorted by their severity level.

For sorting, Bubble Sort is used.

Bubble sort is easy to understand but slow algorithm.

Time complexity is $O(n^2)$.

For small patient number it is okay but for large system not good.

If heap structure is used, priority handling would be better.

Heap always keeps highest priority element on top.

Insert and delete operations work in O(log n) time.

In conclusion, Task 4 shows connection of different data structures.

Previous tasks are reused and not written again.

This task helps to understand how data structures use in real systems.