Task 3 Report – Discharge Stack

In Task 3, a discharge management system is created using a Stack data structure. The purpose of this task is to understand LIFO (Last In First Out) behavior and how it can be used in hospital discharge records. Each discharged patient is stored as a discharge record in the stack.

The stack is implemented using a linked list. The push operation adds a new discharge record to the top of the stack, and the pop operation removes the most recently added record. Both operations have **O(1)** time complexity because only the top pointer is changed. The peek operation is also **O(1)** since it only reads the top element. Printing the stack takes **O(n)** time.

Stack is suitable for managing discharge records because the most recent discharge is usually checked first. For example, if a mistake happens, the last discharged patient can be reviewed easily. This makes stack a logical choice for this scenario.

If a Queue was used instead of a Stack, the first discharged patient would be processed first. This does not fit the requirement of accessing recent discharge information quickly. Although Queue and Stack have similar time complexities, their usage purpose is different.

In conclusion, Stack is a good choice for discharge records because it allows fast access to the most recent data and matches the LIFO logic required in this task.