# TASK 1 - REPORT

In this task, a singly linked list is used to manage admitted patients in a hospital system.

Each patient is stored as a Patient object, and these objects are connected with nodes.

Each node keeps patient data and a reference to the next node.

The addPatient method adds a new patient to the end of the list.

If the list is empty, the new patient becomes the head of the list.

If the list is not empty, the program moves through the list until the last node.

For this reason, the time complexity of this method is O(n).

The removePatient method removes a patient by using the patient ID.

First, the list is checked from the beginning.

If the patient is at the head, the head is moved to the next node.

If the patient is in the middle or at the end, the previous node is found and links are updated.

In the worst case, the whole list is checked, so the time complexity is O(n).

The findPatient method searches a patient by ID.

The program checks each node one by one until the patient is found.

If the patient is not in the list, all nodes are checked.

Because of this, the time complexity of this method is also O(n).

When we compare Linked List and ArrayList, both have advantages and disadvantages.

Linked List is good for adding and removing elements because there is no shifting.

ArrayList is faster for accessing elements by index, because it supports direct access.

However, removing an element from the middle of an ArrayList is slower.

In this task, Linked List is a good choice because patient data can change often.