

THEORY – 6

PROGRAM -1

AIM- Linked List

THEORY

A linked list is a linear data structure where elements are stored in nodes. Each node contains data and a reference (or pointer) to the next node in the sequence. The last node typically points to null, indicating the end of the list.

Here are some key points about linked lists:

1. ****Nodes****: The basic building blocks of a linked list are nodes. Each node contains two fields: data and a reference (or pointer) to the next node in the sequence.
2. ****Head****: The first node of the linked list is called the head. It acts as the starting point for accessing the elements of the list.
3. ****Traversal****: To traverse a linked list, you start from the head and follow the pointers to visit each node in sequence until you reach the end of the list (i.e., a node with a null pointer).
4. ****Insertion****: Inserting an element into a linked list involves creating a new node with the given data and adjusting the pointers to maintain the logical order of the list. Insertion can be done at the beginning, end, or any specific position in the list.
5. ****Deletion****: Deleting an element from a linked list involves adjusting the pointers to remove the node containing the desired data. Similar to insertion, deletion can be performed at the beginning, end, or any specific position in the list.
6. ****Types of Linked Lists****: There are different types of linked lists, including singly linked lists (each node has only one pointer to the next node), doubly linked lists (each node has pointers to both the next and previous nodes), and circular linked lists (the last node points back to the first node, forming a circular structure).