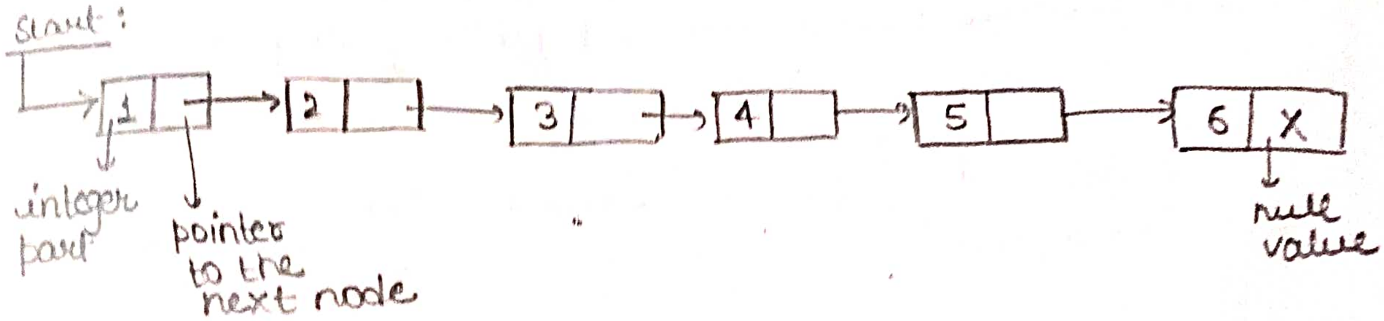
Assignment 5

What is the linked list & Explain Singly Linked List, Doubly linked list & Circular linked list with an example.

→ The linked lists are linear data structures where the data is not stored at contiguous memory locations so we can only access the elements of the linked list in a sequential manner. Linked lists are used to overcome the shortcoming of arrays in operations such as deletion, insertion, etc. The elements in a linked list are linked using pointers.
Unlike an array, a linked list doesn't allow random access of data.

Structure of a linked list:-



linked list implementation:

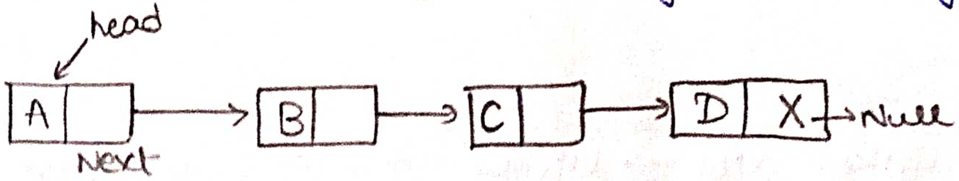
```
Struct node
{ int data ;
  Struct node *next;
};
```

* Types of linked list:-

Singly linked list

- It is the simplest type of linked list in which every node contains some data & a pointer to the next node of the same data type
- It allows the traversal of data only in one way.

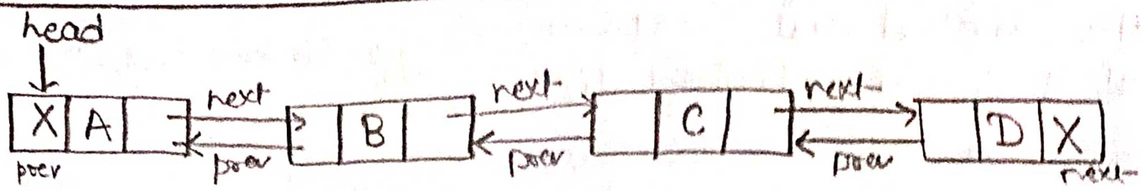
eg.



Doubly linked list

A doubly linked list or a two-way linked list is a more complex type of linked list that contains a pointer to the next as well as the previous node in sequence.
It contains 3 parts of data, a pointer to the next node, & a pointer to the previous node.
This would enable us to traverse the list in the backward direction as well.

eg,



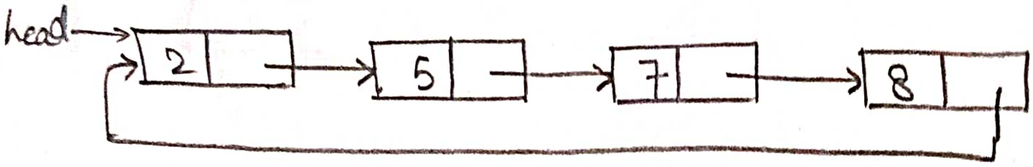
© Circular linked list.

→ A circular linked list is that in which the last node contains the pointer to the first node of the list.

→ while traversing a circular linked list, we can begin at any node & traverse the list in any direction forward & backwar until we reach the same node we started.

→ Thus, a circular linked list has no beginning and no end.

eg,



Q.2 Explain Insertion, Deletion, & Modification of the nodes of Singly linked list, Doubly linked list & Circular linked lis

Ⓐ Insertion

i) In Singly linked list → To insert a node, adjust the pointers of the preceding & succeeding nodes accordingly.

ii) In Doubly linked list → Insertion involves updating pointers of the preceding & succeding nodes to include the new node.

iii) In Circular linked list → Insertion involves updating pointers to maintain the circular structure.

Ⓑ Deletion

i) In Singly linked list → Update pointers to bypass the node to be deleted.

ii) In Doubly linked list → Adjust pointers of the preceding & Succeding nodes to remove the node.

iii) In Circular linked list → update pointers to remove the node & maintain the circular structure.

(c) Modification:

i) Singly linked list → Traverse to the desired node & update its data.

ii) Doubly linked list → Similar to singly linked list, but traversal can be in both direction.

iii) Circular linked list → Modify the data of the desired node, Similar to singly linked list, while considering the circular Structure.