Advantages of Linked List:

1. Linked List being a **dynamic data structure** can shrink and grow at the runtime by deallocating or allocating memory, so there is no need for an initial size in linked list.
2. As the size of a linked list can grow or shrink at runtime, so there is **no memory wastage**. Only the required memory is allocated.
3. Some very helpful data structures like queues and stacks can be easily implemented using a Linked List.
4. **Easy Insertion and Deletion**: In a Linked List, insertion and deletion operations are quite easy, as there is no need to shift every element after insertion or deletion. Only the address present in the pointers needs to be updated

Disadvantages of Linked List

1. The memory required by a linked list is more than the memory required by an array due to presence of pointer.
2. To access node an at index **x** in a linked list, we have to traverse through all the nodes before it.
3. In a singly linked list, reverse traversal is not possible, as every node stores only the address of the next node.

Array vs Linked List

1. Array is stored in contiguous memory location but linked list not.
2. Array is fixed in size, LL is dynamic in size.
3. In Array, memory is allocated at compile time, but in LL memory is allocated in Runtime.
4. Array uses less memory, LL uses more memory.
5. In array elements can be accessed easily, LL required full traversal.
6. Array has slower insertion and deletion, but L has faster.