Data Structures Summary

1. What is a Data Structure?

A data structure is a way of organizing and storing data in memory for efficient access and modification.

2. Types of Data Structures (Mind Map):

- Linear Structures: Array, Linked List, Stack, Queue
- Non-Linear Structures: Tree, Graph
- Hash-based Structures: Hash Table

3. Overview of Data Structures:

- Array: Contiguous memory, same data type. Example: [1,2,3,4]
- Linked List: Nodes with data + pointer. Example: A -> B -> C
- Stack: LIFO structure. Example: Top -> [3,2,1]
- Queue: FIFO structure. Example: Front -> [1,2,3] <- Rear
- Tree: Hierarchical structure. Example: A -> B, C
- Graph: Nodes and edges. Example: A - B - C
- Hash Table: Key-value pairs. Example: {"name": "Ahmed"}

4. What is a Linked List?

A collection of nodes, each containing data and a pointer to the next node.

Types of Linked Lists:

- Singly Linked List: A -> B -> C -> NULL
- Doubly Linked List: NULL <- A <-> B <-> C -> NULL
- Circular Linked List: A -> B -> C -> A

Applications:

- Dynamic memory allocation

- Stack & Queue implementation

- OS scheduling

5. Stack vs Queue - Comparison:

Feature	Stack (LIFO)	Queue (FIFO)
Insertion	Top	Rear
Deletion	Top	Front
Functions	push(), pop()	enqueue(), dequeue()
Examples	Undo, Recursion	Task scheduling, Print queue