**Lab Session**

Name - Aadarsh Deshmukh

Roll No - 150096724058

## **Questions**

1. Types of Data Structures
   * What are the main types of data structures, and can you provide an example of each?

Answer->

There are Two main types of data structures :

1. Primitive Data Structure
   1. This data structure only holds a single value
   2. The int,char,float,double are the primitive data types
   3. It is directly operated upon by the machine instructions.
2. Non-Primitive Data Structure
   1. It is a more sophisticated data structure.
   2. It is derived from primitive data structure.
   3. emphasize on structuring of a group of homogeneous (same type) or heterogeneous (different type) data items.
   4. Lists, Stack, Queue, Tree, Graph are example of non-primitive data structures.
   5. In this type of structure , there are two more sub type Linear and Non Linear
      1. Linear :
         1. In linear data structure, the data elements are organized in sequential manner
         2. The data structures used for this purpose are Arrays, Linked list, Stacks, and Queues.
         3. In these data structures, one element is connected to another element in a linear form.
      2. Non Linear:
         1. Data structure in which elements are organized in any arbitrary order.
         2. When one element is connected to the 'n' number of elements, it is known as a non-linear data structure.
         3. Example: trees and graphs.
3. Importance of Data Structures
   * Why are data structures important in programming? Discuss how they affect algorithm efficiency.

Answer->

Data structures play a vital role in programming for several reasons:

1. They organize and store data in ways that enhance performance.
2. The right data structures can lower time complexity, leading to faster execution.
3. They facilitate effective memory usage.
4. Data structures are used in various applications, such as GPS routing (graphs) and database indexing (trees).

Influence on Algorithm Performance:

* The selection of a data structure significantly impacts algorithm performance.
* For instance, using a hash table for search operations can achieve a time complexity of O(1), whereas searching in an array typically has a time complexity of O(n).

1. Characteristics of Arrays
   * What are the key characteristics of an array in c++? How does it differ from other data structures like linked lists?

Answer->

Characteristics of Arrays in C++:

1. The size of an array is established at the time of declaration and cannot be changed , It means that the array is fixed size.
2. Elements are stored in consecutive memory locations, allowing for efficient access.
3. Elements can be accessed directly using their index, enabling quick retrieval.
4. All elements within an array must be of the same data type.

Differ from other data structures:

* **Memory Allocation**: Arrays utilize contiguous memory allocation, whereas linked lists consist of scattered memory locations linked by pointers.
* **Insertion/Deletion**: Insertion and deletion operations are generally faster in linked lists, while arrays may require shifting elements, making these operations slower.
* **Access Method**: Arrays allow for random access to elements, while linked lists necessitate sequential traversal to access elements.

1. Fundamental Concepts of Algorithms
   * Define an algorithm and explain its fundamental concepts. How do data structures play a role in algorithm design?

Answer->  
An algorithm is a finite sequence of well-defined instructions designed to solve a specific problem.

In addition, all algorithms must satisfy the following criteria:

1. Input. There are zero or more quantities that are externally supplied.
2. Output. At least one quantity is produced.
3. Definiteness. Each instruction is clear and unambiguous.
4. Finiteness. If we trace out the instructions of an algorithm, then for all cases,the algorithm terminates after a finite number of steps.
5. Effectiveness. Every instruction must be basic enough to be carried out, inprinciple, by a person using only pencil and paper.

* Role of Data Structures in Algorithm Design:
  + Data structures determine how data is organized and accessed, which in turn affects the design and efficiency of algorithms.
  + Example: Sorting algorithms, such as QuickSort, utilize arrays, while Dijkstra's algorithm is based on graph structures.

1. Array Implementation
   * How do you declare and initialize an array in c++? Provide an example of creating an array of integers.

Answer->

I have written the code in cpp file