 # Data Structures:
 # Section-B: 7 Questions

Q. Why there is a need of data structure?

Q. What is computer?
it is a machine/hardware/digital device which does different tasks for users efficiently and effectively.

+ basic functions of computer:
    1. data storage
    2. data processing
    3. data movement
    4. control

+ there is a need of data structure to acheive following 3 things in programming:
    1. efficiency
    2. abstraction
    3. reusability

Q. What is data structure?

- it is a way to store data into the memory (i.e. into the main memory) in an organized manner so that operations (like insertion, deletion, searching, sorting, traversal etc....) can be performed on it efficiently.

- there are two types of data structures:

1. linear/basic data structures: data ele's gets stored in a linear manner and hence can be accessed in a linear manner.
- array
- structure & union
- class
- linked list
- stack
- queue

2. non-linear/advanced data structures: data ele's gets stored in a non-linear manner and hence can be accessed in a non-linear manner.
- tree (heirachical manner)
- graph (network)
- hash table (associative manner)
- binary heap

etc...

"array": it is a collection/list of logically related similar type of elements gets stored into the memory contiguosly.

"structure": it is a collection/list of logically related similar and disimilar type of elements (gets stored into the memory collectively).

"union": same as structure execept memory allocation

"class": inside class we can combine/collect "data members" (variables) as well as "member functions" (function can be used to perform operations on,
data), wheras in a structure we can combine/collect only data members.

- to learn data structure is nothing to learn an algorithms

Q. What is an algorithm?

- it is a finite set of instructions written in any human understandable language (like english), if followed, acomplishesh the given task.

OR

- it is a finite set of instructions written in any human understandable language (like english) with some programming constraints, if followed, acomplishesh the given task, such kind of algo is reffered as "pseudocode".
- an algorithm is solution of a given problem
- algorithm = solution

e.g. Problem: **"Sorting":** it is a process to arrange data ele's in a collection/list of elements either in an ascending order or in a descending order.

- Sorting Algorithms/Solutions
    - A1: Selection Sort
    - A2: Bubble Sort
    - A3: Insertion Sort
    - A4: Quick Sort
    - A5: Merge Sort
    - A6: Radix Sort
    - A7: Bucket Sort
    - A8: Shell Sort

- One problem has many solutions, and hence when we have many solutions for a single problem we need to select an efficient solution.
- to decide efficiency of an algorithms we need to do their analysis
- analysis of an algo is a work to calculating how much "time" i.e. computer time and "space" i.e. computer memory an algo needs to run to completion.

- there are two measures of an analysis of an algo:

1. time complexity

2. space complexity

1. "time complexity": of an algo is the amount computer time it needs to run to completion.

2. "space complexity": of an algo is the amount space i.e. computer memory it needs to run to completion.

"Best Case Time Complexity": When an algo takes min amount of time then it is considered as best case time complexity.

"Worst Case Time Complexity": When an algo takes max amount of time then it is considered as worst case time complexity.

"Average Case Time Complexity": When an algo takes neither min nor max amount of time then it is considered as an average case time complexity.

 # Array:
Searching Algorithms:
"Searching": it is a process to find key in a given collection/list of elements.
- there are two searching algo's we can apply on an array data structure
1. Linear Search: In this algorithm key elements gets compared with each element in a list/collection of elements from the first element till the match is not found or maximum till last element sequentially, and if match is found it returns true and if match not found it returns false.

- Linear search is also called as "sequential search".

Algorithm/Pseudocode:

```
Algorithm LinearSearch(A, n, key)//A is an array of size "n" & key to be search
{
        for( index = 1 ; index <= n ; index++ )
        {
                //if key is found at any pos then return true
                if( key == A[index] )
                        return true;
        }

        //if key is not found at any position
        return false;
}
```

- In Linear search, best case occures when key is found at very first position, and only 1 no. of comparison takes place in this case and hence
best case time complexity: O(1).

- In Linear search, worst case occures when either key is exists at last position or key does not exists and max "n" (whereas n = size of an array) no. of comparisons takes place, and hence
worst case time complexity : O(n).

- In Linear search, average case occures when key is exists at in between position i.e key does not exists neither first nor last position, and hence
an average case time complexity: O(n/2).