**DATA STRUCTURES**
**DEFINITION, TYPES AND OPERATIONS**

1. The non-homogeneous data elements cannot be stored in which of the following data structures?
   a) Arrays b) Records c) Pointers d) Stacks
   **Ans: a**

2. The array data structure drawback is that
   a) The amount of memory to be allocated should be known beforehand
   b) Elements of array can be accessed in constant time
   c) Elements are stored in contiguous blocks
   d) Multiple other data structures can be implemented using arrays
   **Ans: a**

3. _____ are mathematical models with a set of operations defined on them.
   a) Data Structure
   b) Abstract Data Type
   c) Algorithm
   d) Primitive
   **Ans: b**

4. _____ will be used to hold information about an array that is used in a programme.
   a) Symbol table
   b) Dope vector
   c) Activation record
   d) System table
   **Ans: d**

5. An array of n numbers is given, where n is an even number. The maximum as well as the minimum of these n numbers needs to be determined. Which of the following is TRUE about the number of comparisons needed?
   a) Atleast $n_{+2}$ comparisons
   b) Atleast 2n-c comparisons for some constant c, are needed.
   c) Atleast $n\log_2 n$ comparisons are needed
   d) Atmost 1.5n-2 comparisons are needed
   **Ans: d**

6. Which of the following data structures is suitable for representing the hierarchical relationships between elements?
   a) Stack
   b) Tree
   c) Queue
   d) List
   **Ans: b**

7. A program P reads in 500 integers in the range [0..100] representing the scores of 500 students. It then prints the frequency of each score above 50. What would be the best way for P to store the frequencies?
   a) An array of 100 numbers
   b) An array of 500 numbers
   c) An array of 50 numbers
   d) a array dynamically allocated array of 550 numbers
   **Ans: c**

8. The number of comparisons done by sequential search is
a) (n/2) +1
b) (n+1)/2
c) (n-1)/2
d) (n+2)/2
**Ans: b**

**BTL 2:**

1. **Summarize data structure.**
   A data structure is a method for organizing and storing data which would allow efficient data retrieval and usage.
   A data structure is a way of organizing data that considers not only the items stored, but also their relationships to each other.

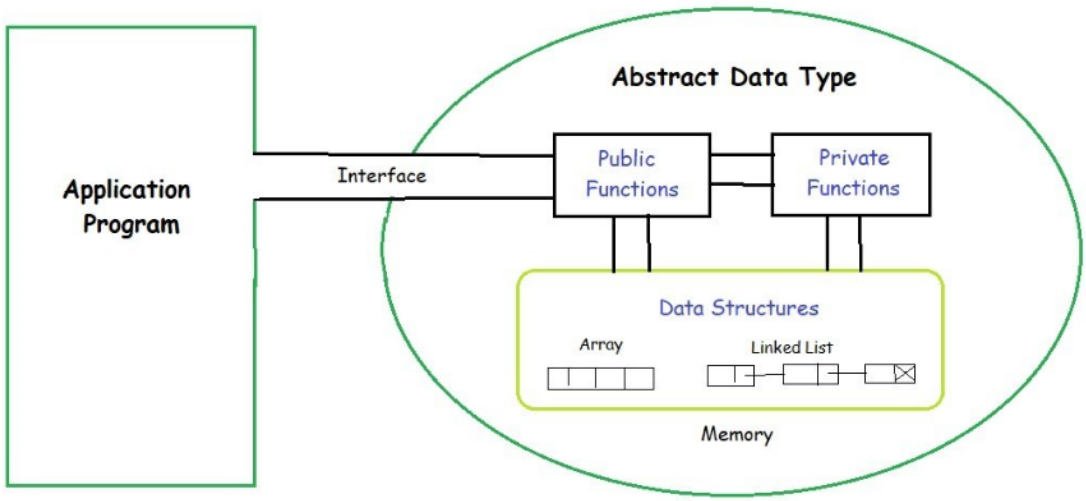
2. **Distinguish linear and non-linear data structure.**

| **Linear Data Structure** | **Non-linear Data Structure** |
|---|---|
| In this structure, the elements are arranged sequentially or linearly and attached to one another. | In this structure, the elements are arranged hierarchically or non-linear manner. |
| Arrays, linked list, stack, queue are the types of a linear data structure. | Trees and graphs are the types of a non-linear data structure. |
| Due to the linear organization, they are easy to implement. | Due to the non-linear organization, they are difficult to implement. |
| As linear data structure is a single level, so it requires a single run to traverse each data item. | The data items in a non-linear data structure cannot be accessed in a single run. It requires multiple runs to be traversed. |
| Each data item is attached to the previous and next items. | Each item is attached to many other items. |
| This data structure does not contain any hierarchy, and all the data elements are organized in a single level. | In this, the data elements are arranged in multiple levels. |
| In this, the memory utilization is not efficient. | In this, memory is utilized in a very efficient manner. |

| | |
|---|---|
| The time complexity of linear data structure increases with the increase in the input size. | The time complexity of non-linear data structure often remains same with the increase in the input |
| | |

3. **Discuss Abstract Data Type.**
   Abstract Data type (ADT) is a type (or class) for objects whose behavior is defined by a set of values and a set of operations. The definition of ADT only mentions what operations are to be performed but not how these operations will be implemented. It does not specify how data will be organized in memory and what algorithms will be used for implementing the operations. It is called "abstract" because it gives an implementation-independent view.

   The process of providing only the essentials and hiding the details is known as abstraction.




4. **Paraphrase the features of ADT.**
   a. Modularity
   i. Divide program into small functions
   ii. Easy to debug and maintain
   iii. Easy to modify
   b. Reuse
   i. Define some operations only once and reuse them in future
   c. Easy to change the implementation
5. **Recognize the operations in data structures.**
   **Searching:** We can search for any element in a data structure.

   **Sorting:** We can sort the elements of a data structure either in an ascending or descending order.

   **Insertion:** We can also insert the new element in a data structure.

**Updation:** We can also update the element, i.e., we can replace the element with another element.

**Deletion:** We can also perform the delete operation to remove the element from the data structure.

**SCENARIO BASED QUESTIONS**

1. **How are the elements of a 2D array stored in memory?**

Row-Major Order: -In row-major ordering, all of the rows of a 2D array are stored in memory in a contiguous manner.

First, the first row of the array is entirely stored in memory, followed by the second row of the array, and so on until the final row.

Column-Major Order: In column-major ordering, all of the columns of a 2D array are stored in memory in the same order. The first column of the array is entirely saved in memory, followed by the second row of the array, and so on until the last column of the array is wholly recorded in memory.

2. How do you reference all of the elements in a one-dimension array?
   Using an indexed loop, we may access all of the elements in a one-dimensional array. The counter counts down from 0 to the maximum array size, n, minus one. The loop counter is used as the array subscript to refer to all items of the one-dimensional array in succession.
3. **Do dynamic memory allocations help in managing data? How?**

Dynamic memory allocation stores simple structured data types at runtime. It has the ability to combine separately allocated structured blocks to form composite structures that expand and contract as needed, thus helping manage data of data blocks of arbitrary size, in arbitrary order.

4. **Can you explain the difference between file structure and storage structure?**

- **File Structure:** Representation of data into secondary or auxiliary memory say any device such as a hard disk or pen drives that stores data which remains intact until manually deleted is known as a file structure representation.
- **Storage Structure**: In this type, data is stored in the main memory i.e RAM, and is deleted once the function that uses this data gets completely executed.

The difference is that the storage structure has data stored in the memory of the computer system, whereas the file structure has the data stored in the auxiliary memory.