Important Questions on Linked list 1. Explain linked list and its structure? 2. List various operations performed on Linked list. 3. Explain what does it mean by self-referential structure? Explain its need? 4.Difference between Array and Linklist 5. List types of Linked List. 6.List real time applications of Linked List. 7.Mention some drawbacks of the linked list. 8. List some advantages of Linked List 9. Which data structures can we implement using linked list? 10.How can we implement stack and queue using linked list? 11.How to write a program to create heterogeneous linked list? 12..Mention a package that is used for Linked List class in Java. 13.Mention some interfaces implemented by Linked List in Java. 14.How will you partition linked list 15. How to write a program to reverse linked list? 16. How can we reverse linked list using single pointer? Write a program 17. How to detect whether linked list is linear or circular?

1. Explain linked list and its structure?

Answer-A linked list is a sequence of data structures, which are connected together via links.

Linked List is a sequence of links which contains items. Each link contains a connection to another link. Linked list is the second most-used data structure after array. Following are the important terms to understand the concept of Linked List.

**Link** − Each link of a linked list can store a data called an element.

**Next** − Each link of a linked list contains a link to the next link called Next.

**LinkedList** − A Linked List contains the connection link to the first link called First.

1. List various operations performed on Linked list.

* [Traversal](https://www.programiz.com/dsa/linked-list-operations" \l "traverse) - access each element of the linked list.
* [Insertion](https://www.programiz.com/dsa/linked-list-operations" \l "add) - adds a new element to the linked list.
* [Deletion](https://www.programiz.com/dsa/linked-list-operations" \l "delete) - removes the existing elements.
* [Search](https://www.programiz.com/dsa/linked-list-operations" \l "search) - find a node in the linked list.
* [Sort](https://www.programiz.com/dsa/linked-list-operations" \l "sort) - sort the nodes of the linked list.

1. Explain what does it mean by self-referential structure? Explain its need?

The self-referential structure is a structure that points to the same type of structure. It contains one or more pointers that ultimately point to the same structure.

**Applications:**   
Self referential structures are very useful in creation of other complex data structures like: 

* [Linked Lists](https://www.geeksforgeeks.org/data-structures/linked-list/)
* [Stacks](https://www.geeksforgeeks.org/stack-data-structure/)
* [Queues](https://www.geeksforgeeks.org/queue-data-structure/)
* [Trees](https://www.geeksforgeeks.org/binary-tree-data-structure/)
* [Graphs](https://www.geeksforgeeks.org/graph-and-its-representations/) etc

4.Difference between Array and Linklist.

****Let's look at the differences between the array and linked list in a tabular form.****

|  |  |
| --- | --- |
| **Array** | **Linked list** |
| An array is a collection of elements of a similar data type. | A linked list is a collection of objects known as a node where node consists of two parts, i.e., data and address. |
| Array elements store in a contiguous memory location. | Linked list elements can be stored anywhere in the memory or randomly stored. |
| Array works with a static memory. Here static memory means that the memory size is fixed and cannot be changed at the run time. | The Linked list works with dynamic memory. Here, dynamic memory means that the memory size can be changed at the run time according to our requirements. |
| Array elements are independent of each other. | Linked list elements are dependent on each other. As each node contains the address of the next node so to access the next node, we need to access its previous node. |
| Array takes more time while performing any operation like insertion, deletion, etc. | Linked list takes less time while performing any operation like insertion, deletion, etc. |
| Accessing any element in an array is faster as the element in an array can be directly accessed through the index. | Accessing an element in a linked list is slower as it starts traversing from the first element of the linked list. |
| In the case of an array, memory is allocated at compile-time. | In the case of a linked list, memory is allocated at run time. |
| Memory utilization is inefficient in the array. For example, if the size of the array is 6, and array consists of 3 elements only then the rest of the space will be unused. | Memory utilization is efficient in the case of a linked list as the memory can be allocated or deallocated at the run time according to our requirement. |

5. List types of Linked List

**There are four key types of linked lists:**

* Singly linked lists.
* Doubly linked lists.
* Circular linked lists.
* Circular doubly linked lists.