

## UNIT – III

### 1. Define Linked Lists? Or What is Automatic list management?

Linked list consists of a series of structures, which are not necessarily adjacent in memory. Each structure contains the element and a pointer to a structure containing its successor. We call this the Next Pointer. The last cell's->next pointer points to NULL.



### 2. How can you overcome the limitations of arrays

Limitations of arrays can be solved by using the linked list.

### 3. Define HEAD pointer and NULL pointer?

HEAD-It contains the address of the first node in that list.

NULL - It indicates the end of the list structure.

### 4. What is the need for the header?

Header of the linked list is the first element in the list and it points to the first data element of the list, i.e. it contains the address of the first element in the list. If needed, it may store the number of elements in the list.

### 5. What is a node?

The data element of a linked list is called a node. Node consists of two fields: data field to store the element and link field to store the address of the next node.

### 6. List three examples that uses linked list?

- a. Polynomial ADT
- b. Radix sort
- c. Multi lists

### 7. State the difference between arrays and linked lists?

Arrays	Linked Lists
Size of an array is fixed	Size of a list is variable
It is necessary to specify the number of elements during declaration.	It is not necessary to specify the number of element during declaration
Insertions and deletions are somewhat difficult	Insertions and deletions are carried out easily
It occupies less memory than a linked list for the same number of elements	It occupies more memory

### **8. List the basic operations carried out in a linked list?**

The basic operations carried out in a linked list include:

- a. Creation of a list
- b. Insertion of a node
- c. Deletion of a node
- d. Modification of a node
- e. Traversal of the list

### **9. List out the different ways to implement the list?**

1. Array Based Implementation
2. Linked list Implementation
  - i. Singly linked list
  - ii. Doubly linked list
  - iii. Cursor based linked list

### **10. List out the advantages of using a linked list?**

- a. It is not necessary to specify the number of elements in a linked list during its declaration
- b. Linked list can grow and shrink in size depending upon the insertion and deletion that occurs in the list
- c. Insertions and deletions at any place in a list can be handled easily and efficiently
- d. A linked list does not waste any memory space

### **11. List out the disadvantages of using a linked list?**

- e. Searching a particular element in a list is difficult and time consuming
- f. A linked list will use more storage space than an array to store the same number of elements

### **12. State the different types of linked lists?**

The different types of linked list include

- i. Singly linked list
- ii. Doubly linked list
- iii. Cursor based linked list

### **13. What are the advantages of Linked List over arrays?**

1. In Linked List implementation Memory location need not be necessarily contiguous.
2. Insertion and deletions are easier and needs only one pointer assignment.
3. A small amount of memory is been wasted for storing a pointer, which is been associated with each node.

**14. List the operations of single linked list?**

- a. MakeEmpty
- b. IsEmpty
- c. IsLast
- d. Find
- e. Delete
- f. FindPrevious
- g. Insert
- h. Deletelist

**15. List out the advantages and disadvantages of singly linked list**

**Advantages:**

- Easy insertion and deletion.
- Less time consumption.

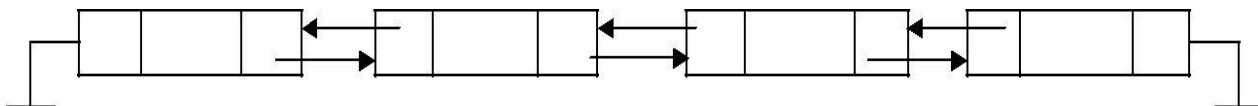
**Disadvantages:**

- Data not present in a linear manner.
- Insertion and deletion from the front of the list is difficult without the use of header node.

**16. What is a doubly linked lists? (Nov 2012)**

Doubly linked list is a collection of nodes where each node is a structure containing the following fields.

1. Pointer to the previous node.
2. Data.
3. Pointer to the next node



**17. Compare Single linked list and double linked list?**

Singly	Doubly
It is a collection of nodes and each node is having one data field and next link field	It is a collection of nodes and each node is having one data field one previous link field and one next link field
The elements can be accessed using next link	The elements can be accessed using both previous link as well as next link
No extra field is required hence, Node takes less memory in SLL.	One field is required to store previous Link Hence, node takes memory in DLL.
Less efficient access to elements	More efficient access to elements.

**18. Write the difference between doubly and circularly linked list?**

Doubly	Circularly
If the pointer to next node is Null, it specifies the last node.	There is no first and last node.
Last node'S->next field is always Null	Last nodeS->next field points to the address of the first node.
Every node has three fields one is Pointer to the pervious node, next Is data and the third is pointer to the next node.	It can be single linked list and double linked list

**19. State Linked list implementation of stack?**

- i. Push operation is performed by inserting an element at the front of the list.
- j. Pop operation is performed by deleting at the front of the list.
- k. Top operation returns the element at the front of the list.

**20. State Linked List implementation of Queue?**

Enqueue operation is performed at the end of the list. Dequeue operation is performed at the front of the list.

**21. List out the applications of a linked list?**

Some of the important applications of linked lists are

- l. manipulation of polynomials,
- m. sparse matrices,
- n. stacks and queues.

**22. Whether Linked List is linear or Non-linear data structure?**

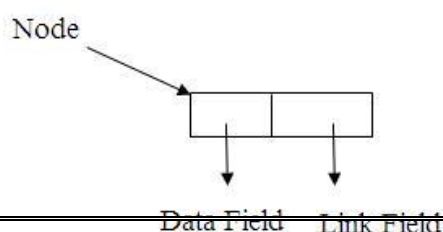
According to Access strategies Linked list is a linear one. According to Storage Linked List is a Non- linear one.

**23. State the disadvantages of linked list implementation of stack**

- 1. Calls to malloc and free functions are expensive.
- 2. Using pointers is expensive.

**24. What is Linked list? (Nov 11, Nov 13, Apr 15)**

- ☐ Linked list is a dynamic and linear data structure.
- ☐ Linked list is an ordered collection of elements in which each element is referred as a node.
- ☐ Each node has two fields namely
  - i. Data field or Information field and
  - ii. Address field or Link field.



## 25.What are various fields in a Linked list?

Each node has two fields namely

- i. Data field or Information field and
- ii. Address field or Link field.

**Data Field:** Data field contains the actual data.

**Address Field:** Address field contains the address of another node.

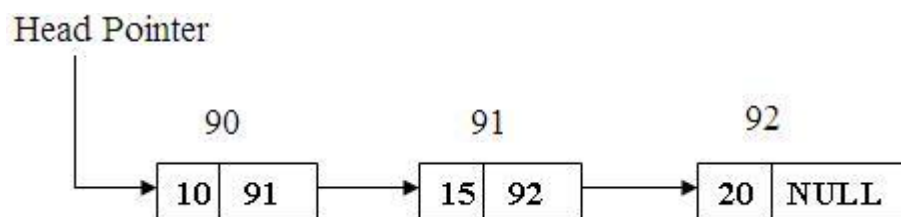
## 26.What are the types of Linked List? (Apr 15)

There are three types of Linked list. They are

- i. Single Linked list,
- ii. Double Linked list, and Circular Linked list.

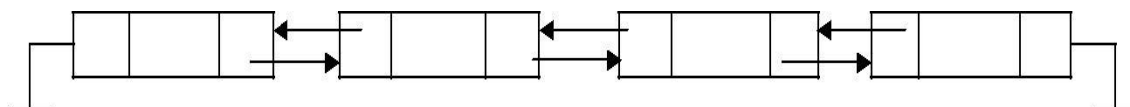
## 27.What is Single linked list?

- In Single linked list, each node has one link to the next node.
- In Single linked list, we can move from only one direction from head pointer to Null pointer.
- Each node has two fields namely
  - i. Data field or Information field and
  - ii. Address field or Link field.
- It is otherwise called as Linear Linked list.
- Consider the following single linked list,



## 28.What is Double Linked list? (Apr 12, Apr 13) (NOV 15)

- Each node consist of three fields namely
  - i. Previous address field or Backward link field,
  - ii. Data field or Information field,
  - iii. External address field or Forward link field.
- The previous address field holds the address of the previous node and the next address field holds the address of next node.
- In Double Linked list, we can move in both the direction from head pointer to Null address or vice versa.
- Consider the following linked list,

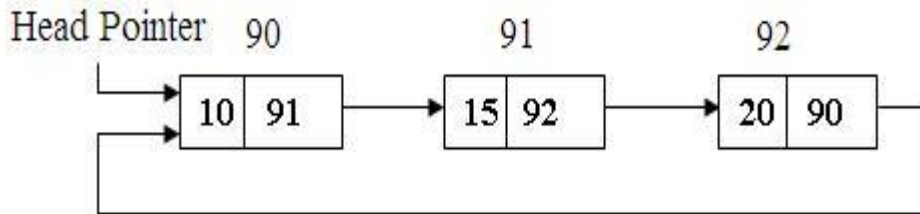


## 29 .What is Circular Linked list? Mention its types.

- In Circular linked list, the last node is connected to the first node.
- In Circular Linked list, we can move from head pointer to Null address.
- There are two types of Circular linked list. They are
  - i. Circular Single Linked list and
  - ii. Circular Double Linked list.

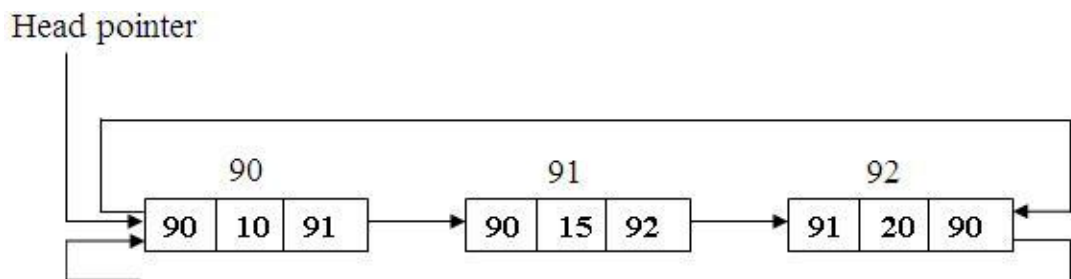
### 30 .What is Circular Single Linked list?

- In Circular Single linked list, the last node is connected to the first node.
- In Circular Single linked list, we can move from only one direction from head pointer to Null pointer
- Consider the following circular single linked list,



### 31.What is Circular Double Linked list?

- In Circular Double linked list, the last node is connected to the first node.
- In Circular Double Linked list, we can move in both the direction from head pointer to Null address or vice versa. Consider the following circular single linked list,



### 32. In how many ways we can insert a node into a single linked list?

We can insert a new node into the list by following positions

- Inserting a node as a first node,
- Inserting a node as a last node and
- Inserting a node as an intermediate node.

### 33.How does a stack-linked list differ from a linked list?

A stack linked list refers to a stack implemented using a linked list. That is to say, a linked list in which you can only add or remove elements to or from the top of the list. A stack-linked list accesses data last in, first out; a linked list accesses data first in, first out.

### 34.List out the application of a linked list .

Some of the important applications of linked lists are

- Manipulation of polynomials,
- Stacks and
- Queues.

**35.State the difference between arrays and linked list.**

Arrays	Linked List
<input type="checkbox"/> Size of any arrays is fixed.	<input type="checkbox"/> Size of a list is variable.
<input type="checkbox"/> It is necessary to specify the number of Elements during the declaration.	<input type="checkbox"/> It is not necessary to specify the number of elements during the Declaration.
<input type="checkbox"/> It occupies less memory space than linked list for the same memory number of elements.	<input type="checkbox"/> It occupies more memory space.

**36. Advantages of circular linked list.**

- Any node can be a starting point. We can traverse the whole list by starting from any point. We just need to stop when the first visited node is visited again.
- Useful for implementation of queue. Unlike this implementation, we don't need to maintain two pointers for front and rear if we use circular linked list. We can maintain a pointer to the last inserted node and front can always be obtained as  $S \rightarrow \text{next of last}$ .
- Circular lists are useful in applications to repeatedly go around the list. For example, when multiple applications are running on a PC, it is common for the operating system to put the running applications on a list and then to cycle through them, giving each of them a slice of time to execute, and then making them wait while the CPU is given to another application. It is convenient for the operating system to use a circular list so that when it reaches the end of the list it can cycle around to the front of the list.