

1. **Abstract Data Type** is a set of data values and associated operations that are specified accurately, independent of any particular implementation.
2. A **Data Structure** is a specialized format for organizing, processing, retrieving and storing data.
3. Identify the operation that is performed more efficiently by a doubly linked list than a singly linked list. **Deleting a node whose location is given .**
4. A variant of the linked list in which none of the node contains NULL pointer is? **Circular linked list .**
5. Every node of a linked list consist of ? **data and address of the next element in the list.**
6. Linked list is considered as an example of **Dynamic** type of memory allocation.
7. Which of the following points is not true about Linked List data structure when it is compared with array? **Access of elements in linked list takes less time than compared to arrays .**
8. Which of the following statements about linked list data structure is TRUE? **The linked list pointers do not provide an efficient way to search an item in the linked list .**
9. The minimum number of fields with each node of singly linked list is **2** .
10. Under which circumstance the condition head ==NULL will be true on Linked list? **List is empty.**
11. Which is true about a circular linked list? **There is no beginning and no end.**

12. How can a traversal in a doubly linked list achieved? **Can travel in both the directions.**
13. Which type of linked list will not have NULL pointer at all? **Circular Linked List.**
14. In a circular linked list organization, insertion of a record involves the modification of **2 pointers.**
15. The best data structure used for representing and implementing Polynomials is **Linked List.**
16. In the real world scenario, We can access previous and next URL searched in web browser by pressing back and next button. Which type of linked list is used for implementing this type of applications? Doubly **Linked List.**
17. Since a doubly-linked list allows traversing in both the forward and backward directions, it is also referred to as a **Two way** list.
18. How many NULL pointers a doubly linked list can have at minimum? **2.**
19. Which of the following is false about a doubly linked list? **Implementing a doubly linked list is easier than singly linked list.**
20. Which of the following is not a disadvantage to the usage of array? **Accessing elements at specified positions.**
21. Which is the best answer to the question, “What is the item at position n?” ? **Array based implementation of list.**
22. Suppose p is a pointer variable that contains NULL pointer. What happens if your program tries to read or write \*p? **A runtime error always occurs when \*p is evaluated.**

23. Linked list is not suitable to implement **Binary Search**.
24. The value of start=NULL in a linked list indicates **Underflow** situation.
25. To implement Sparse matrix dynamically, the following data structure is used. **Linked List**
26. Linked list is generally considered as an example of **Dynamic** type of memory allocation.
27. A linear collection of data elements where the linear node is given by means of pointer is called? **Linked List**.
28. A collection of data items of similar type arranged in a sequence is termed as? **List**.
29. Defining a structure, allocating memory and initializing it is the operation of **creating a node**.
30. **Traversing** a list means accessing its elements one by one to process all or some of the elements.
31. What does the following function do for a given Linked List with first node as head? `void fun1(struct node* head) { if(head == NULL) return; fun1(head->next); printf("%d ", head->data); }` **Prints all nodes of linked list in reverse order.**
32. Consider the following function to traverse a linked list.  
`void traverse(struct Node *head) { while (head->next != NULL) { printf("%d ", head->data); head = head->next; } }` **The function is implemented incorrectly because it changes head .**

33. In the given answers identify the applications of doubly linked list? **The browser cache which allows you to hit the BACK buttons.**
34. A variation of linked list is circular linked list, in which the last node in the list points to first node of the list. One problem with this type of list is? **It is difficult to traverse the list as the pointer of the last node is now not NULL .**
35. Which of the following operations is performed more efficiently by doubly linked list than by singly linked list? **Deleting a node whose location is given.**
36. Choose the best answer to implement a memory efficient doubly Linked List. **Each node has only one pointer to traverse the list back and forth.**
37. How a circular linked list differs from a normal linked list? **We cannot have the 'next' pointer point to null in a circular linked list.**
38. Identify the application that makes use of a circular linked list. **Allocating CPU to resources.**
39. Depending on  
....., a linked list be classified into various other types.
- The number of pointers in a node.**
  - The purpose for which the pointers are maintained.**
40. `struct node { int data; struct node *next; }`  
`*head = NULL;` Consider the above representation and predict what will be printed on the screen by following statement ? `head->next->data`. **Does not access any field.**

41. Consider the following linked list representation- struct  
node { int data; struct node \*next; }\*head =  
NULL; struct node \*new\_node; Which among the  
following statement will create a new node?  
**new\_node=(struct node \*)malloc(sizeof(struct node));**
42. Given, \*current = start->next; What "current" will contain  
if it is variable of type struct node ? **Address of 2nd Node.**
43. When new element is added in the middle of singly linked  
list then. **No need to move element.**
44. In a singly linked list which operation depends on length of  
the list? **Adding element to the front of the list.**
45. Consider the following linked list representation  
1-2-5-7-3-6 struct node { int data; struct  
node \*next; }\*head = NULL; What will be the value of  
following statement ? head->next->next->next->data.  
**7.**