List

1. Definition of a list is to be
   a) Finite
   b) Infinite
   c) No-limit
   d) Empty
   Answer: a

2. A linear collection of data elements where the linear node is given by means of pointer is called?
   a) Linked list
   b) Node list
   c) Primitive list
   d) Unordered list
   Answer: a

3. In linked list each node contains a minimum of two fields. One field is data field to store the data second field, is?
   a) Pointer to character
   b) Pointer to integer
   c) Pointer to node
   d) Node
   Answer: c

4. Which of the following is true in case of a null pointer?
   a) Marks the end of a node
   b) Is equal to '\0' in C
   c) Is the address of some node
   d) Is also called the void pointer in C
   Answer: a

5. Consider the following definition in c programming language.

   ```
   struct node
   {
       int data;
       struct node * next;
   }
   typedef struct node NODE;
   NODE *ptr;
   ```

   Which of the following c code is used to create new node?
   a) ptr = (NODE*)malloc(sizeof(NODE));
   b) ptr = (NODE*)malloc(NODE);
   c) ptr = (NODE*)malloc(sizeof(NODE*));

d) ptr = (NODE)malloc(sizeof(NODE));
Answer: a

6. What kind of linked list is best to answer questions like “What is the item at position n?”
a) Singly linked list
b) Doubly linked list
c) Circular Linked list
d) Array implementation of linked list
Answer: d

7. Linked list is considered as an example of ----------- type of memory allocation.
a) Dynamic
b) Static
c) Compile time
d) Heap
Answer: a

8. In Linked List implementation, a node carries information regarding -----------
a) Data
b) Link
c) Data & Link
d) Node
Answer: c

9. Linked list data structure offers considerable saving in -------------
a) Computational Time
b) Space Utilization
c) Space Utilization & Computational Time
d) Speed Time
Answer: c

10. To Maintain a linked list in memory, how many parallel arrays of equal size are used?
a) 1
b) 2
c) 3
d) 4
Answer: b

11. What does create a node mean?
a) Defining its structure
b) Allocating memory to it
c) Initialization
d) All of the above
Answer: d

12. ---------- a list means accessing its elements one by one to process all or some of the elements
    a) Traversing
    b) Creating
    c) Linking
    d) None of the above
    Answer: a

13. 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);
}
```

    a) Prints all nodes of linked lists
    b) Prints all nodes of linked list in reverse order
    c) Prints alternate nodes of Linked List
    d) Prints alternate nodes in reverse order
    Answer: b

14. What is the output of following function for start pointing to first node of following linked list?
    1->2->3->4->5->6

```
void fun(struct node* start)
{
   If ( start == NULL)
      return;
  printf("%d  ", start->data);
  if(start->next != NULL )
   fun(start->next->next);
  printf("%d  ", start->data);
}
```

    a) 1 4 6 6 4 1
    b) 1 3 5 1 3 5
    c) 1 2 3 5
    d) 1 3 5 5 3 1
    Answer: d

15. The following C function takes a simply-linked list as input argument. It modifies the list by moving the last element to the front of the list and returns the modified list. Some part of the code is left blank. Choose the correct alternative to replace the blank line.

```
typedef struct node
{
   int value;
   struct node *next;
}Node;
Node *move_to_front(Node *head)
{
   Node *p, *q;
   if ((head == NULL: || (head->next == NULL))
     return head;
   q = NULL; p = head;
   while (p-> next !=NULL)
   {
     q = p;
     p = p->next;
   }
   ---------------------------------------
   return head;
}
```

a) q = NULL; p->next = head; head = p;
b) q->next = NULL; head = p; p->next = head;
c) head = p; p->next = q; q->next = NULL;
d) q->next = NULL; p->next = head; head = p;
Answer: d

16. The following C function takes a single-linked list of integers as a parameter and rearranges the elements of the list. The function is called with the list containing the integers 1, 2, 3, 4, 5, 6, 7 in the given order. What will be the contents of the list after the function completes execution?

```
struct node
{
    int value;
    struct node *next;
};
void rearrange(struct node *list) {
    struct node *p, * q;
    int temp;
    if ((!list) || !list->next)
      return;
    p = list;
    q = list->next;
    while(q)
    {
         temp = p->value;
         p->value = q->value;
```

```
            q->value = temp;
            p = q->next;
            q = p?p->next:0;
        }
    }
```

a) 1, 2, 3, 4, 5, 6, 7
b) 2, 1, 4, 3, 6, 5, 7
c) 1, 3, 2, 5, 4, 7, 6
d) 2, 3, 4, 5, 6, 7, 1
Answer: b

17. Given pointer to a node X in a singly linked list. Only one pointer is given, pointer to head node is not given, can we delete the node X from given linked list?
a) Possible if X not last node
b) Possible if size of linked list is even
c) Possible if size of linked list is odd
d) Possible if X is not first node
Answer: a

18. In doubly linked list, traversal can be performed?
a) Only in forward direction
b) Only in reverse direction
c) In both direction
d) None of the above
Answer: c

19. What is the functionality of the following code?

```
public void function(Node node)
{
    if(size == 0)
        head = node;
    else
    {
        Node temp,cur;
        for(cur = head; (temp = cur.getNext())!=null; cur = temp);
            cur.setNext(node);
    }
    size++;
}
```

a) Inserting a node at the beginning of the size
b) Deleting a node at the beginning of the size
c) Inserting a node at the end of the size
d) Deleting a node at the end of the size
Answer: c

20. What is the functionality of the following piece of code?

```
public int function()
{
    Node temp = tail.getPrev();
    tail.setPrev(temp.getPrev());
    temp.getPrev().setNext(tail);
    size--;
    return temp.getItem();
}
```

a) Return the element at the tail of the list but do not remove it
b) Return the element at the tail of the list and remove it from the list
c) Return the last but one element from the list but do not remove it
d) Return the last but one element at the tail of the list and remove it from the list
Answer: b

21. The following steps in a linked list

```
p = getnode()
  info (p) = 10
  next (p) = list
  list = p
```

a) Pop operation in stack
b) Removal of a node
c) Inserting a node
d) Modifying an existing node
Answer: c

22. Which of the following operations is performed more efficiently by doubly linked list than by linear linked list?
a) Deleting a node whose location is given
b) Searching an unsorted list for a given item
c) Inserting a node after the node with a give location
d) Traversing the list to process each node
Answer: a

23. The minimum number of fields with each node of doubly linked list is
a) 1
b) 2
c) 3
d) 4
Answer: c

24. Which of these is not an application of a linked list?
a) To implement file systems
b) For separate chaining in hash-table
c) to implement non-binary trees

d) Random Access of elements
Answer: d

25. Consider the below representation and predict what will be printed on the screen by following statement ?

```
start->next->data
struct node {
    int data;
    struct node *next;
}*start = NULL
```

a) Access the “data” field of 3rd node
b) Access the “data” field of 1st node
c) Access the “data” field of 2nd node
d) None of these
Answer: c

26. What is the functionality of the following piece of code?

```
public int function(int data)
{
    Node temp = head;
    int var = 0;
    while(temp != null)
   {
       if(temp.getData() == data)
       {
           return var;
       }
       var = var+1;
       temp = temp.getNext();
      }
   return Integer.MIN_VALUE;
}
```

a) Find and delete a given element in the list
b) Find and return the given element in the list
c) Find and return the position of the given element in the list
d) Find and insert a new element in the list
Answer: c

27. Which of the following is false about a doubly linked list?
a) We can navigate in both the directions
b) It requires more space than a singly linked list
c) The insertion and deletion of a node take a bit longer
d) Implementing a doubly linked list is easier than singly linked list
Answer: d

28. What is a memory efficient double linked list?
   a) Each node has only one pointer to traverse the list back and forth
   b) The list has breakpoints for faster traversal
   c) An auxiliary singly linked list acts as a helper list to traverse through the doubly linked list
   d) A doubly linked list that uses bitwise AND operator for storing addresses
   Answer: a

29. When new element is added in the middle of singly linked list then --------.
   a) Only elements that appear after the new element need to be moved
   b) Only elements that appear before the new element need to be moved
   c) No need to move element
   d) Only elements that appear after the new element and before need to be moved
   Answer: c