Questions Linked list

1) In a circular linked list
   a) Components are all linked together in some sequential manner.
   b) There is no beginning and no end.
   c) Components are arranged hierarchically.
   d) Forward and backward traversal within the list is permitted.

2) In linked list each node contain 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

3) What would be the asymptotic time complexity to add a node at the end of singly linked list, if the pointer is initially pointing to the head of the list?
   a) O(1)
   b) O(n)
   c) θ (n)
   d) θ (1)

4) 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) None

5) Which of the following operations is performed more efficiently by doubly linked list than by singly linked list?
   a) Deleting a node whose location in given
   b) Searching of an unsorted list for a given item
   c) Inverting a node after the node with given location
   d) Traversing a list to process each node

6) Consider an implementation of unsorted singly linked list. Suppose it has its representation with a head pointer only. Given the representation, which of the following operation can be implemented in O(1) time?
   i) Insertion at the front of the linked list
   ii) Insertion at the end of the linked list
   iii) Deletion of the front node of the linked list
   iv) Deletion of the last node of the linked list

   a) I and II
   b) I and III
   c) I,II and III
   d) I,II and IV

7) Consider an implementation of unsorted singly linked list. Suppose it has its representation with a head and tail pointer. Given the representation, which of the following operation can be implemented in O(1) time?
   i) Insertion at the front of the linked list
   ii)Insertion at the end of the linked list
   iii) Deletion of the front node of the linked list
   iv) Deletion of the last node of the linked list

   a) I and II
   b) I and III

c) I,II and III
d) I,II and IV

8) Consider an implementation of unsorted doubly linked list. Suppose it has its representation with a head pointer only. Given the representation, which of the following operation can be implemented in O(1) time?
i) Insertion at the front of the linked list
ii) Insertion at the end of the linked list
iii) Deletion of the front node of the linked list
iv) Deletion of the end node of the linked list

a) I and II
b) I and III
c) I,II and III
d) I,II,III and IV

9) Consider an implementation of unsorted doubly linked list. Suppose it has its representation with a head pointer and tail pointer. Given the representation, which of the following operation can be implemented in O(1) time?
i) Insertion at the front of the linked list
ii) Insertion at the end of the linked list
iii) Deletion of the front node of the linked list
iv) Deletion of the end node of the linked list

a) I and II
b) I and III
c) I,II and III
d) I,II,III and IV

10) Consider an implementation of unsorted circular linked list. Suppose it has its representation with a head pointer only. Given the representation, which of the following operation can be implemented in O(1) time?
i) Insertion at the front of the linked list
ii) Insertion at the end of the linked list
iii) Deletion of the front node of the linked list
iv) Deletion of the end node of the linked list
a) I and II
b) I and III
c) I, II, III and IV
d) None

11) Consider an implementation of unsorted circular doubly linked list. Suppose it has its representation with a head pointer only. Given the representation, which of the following operation can be implemented in O(1) time?
i) Insertion at the front of the linked list
ii) insertion at the end of the linked list
iii) Deletion of the front node of the linked list
iv) Deletion of the end node of the linked list
a) I and II
b) I and III
c) I, II and III
d) I,II,III and IV

12) What would be the asymptotic time complexity to add an element in the linked list?
a) O(1)
b) O(n)
c) $O(n^2)$
d) None

13) aWhat would be the asymptotic time complexity to insert an element at the second position in the linked list?
    1) O(1)
    2) O(n)
    3) $O(n^2)$
    4) None

14) The concatenation of two list can performed in O(1) time. Which of the following variation of linked list can be used?
    1) Singly linked list
    2) Doubly linked list
    3) Circular doubly linked list
    4) Array implementation of list

15) In doubly linked lists, traversal can be performed?
    1) Only in forward direction
    2) Only in reverse direction
    3) In both directions
    4) None

16) What kind of data structure is best to answer question like “What is the item at position n?”
    1) Singly linked list
    2) Doubly linked list
    3) Circular linked list
    4) Array implementation of linked list

17) aA variant of the linked list in which none of the node contains NULL pointer is?
    1) Singly linked list
    2) Doubly linked list
    3) Circular linked list
    4) None

18) In circular linked list, insertion of node requires modification of?
    1) One pointer
    2) Two pointer
    3) Three pointer
    4) None

19) Linked lists are not suitable to for the implementation of?
    1) Insertion sort
    2) Radix sort
    3) Polynomial manipulation
    4) Binary search

20) aIn worst case, the number of comparison need to search a singly linked list of length n for a given element is
    1) log n
    2) n/2
    3) $\log_2 n-1$
    4) n

21) aThe following C Function takes a singly- linked list of integers as a parameter and rearranges the elements of the lists. 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

22) The following C function takes a singly 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 left blank.

```
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;
  }
  // Write code here.
  return head;
}
```

Choose the correct alternative to replace the blank line
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;

23) Can we sort linked list using quick sort?
a) True
b) False

24) Can we sort linked list using merge sort?
    a) True
    b) False

25) Best case time complexity to sort linked list?
    a) $O(n^2)$
    b) $O(n)$
    c) $O(n\ log\ n)$
    d) $O(n^3)$

[Programming]

1) Write a program to implement a linked list and give options to user to add element at head, tail or in between at any node given that node's value. Also give options
2) Write a program to implement doubly linked list have all the features of above linked list.
3) Write a program to reverse the singly linked list.
4) Write a program to delete n nodes after m nodes of a linked list.
5) Write a program to delete last occurrence of an item from linked list