1. Can linked list be implemented using arrays?  
**A. Yes**B. No

2. On which principle does stack work?  
**A. FILO**B. FIFO  
C. LILO  
D. Both a and c above

3. An empty list is the one which has no  
A. nodes  
B. data  
**C. both a and b above**D. address

4. In a heap, element with the greatest key is always in the \_\_\_\_\_\_\_\_\_\_\_\_\_\_ node  
A. leaf  
**B. root**C. first node of left sub tree  
D. first node of right sub tree

5. A \_\_\_\_\_\_\_\_\_\_\_\_\_ tree is tree where for each parent node, there is only one associated child node  
A. balanced binary tree  
B. rooted complete binary tree  
C. complete binary tree  
**D. degenerate tree**

6. Items in a priority queue are entered in a \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ order  
**A. random**B. order of priority  
C. as and when they come  
D. none of the above

7. A tree cannot contain cycles  
A. False  
**B. True**

8. In graphs, A hyperedge is an edge that is allowed to take on any number of \_\_\_\_\_\_\_\_\_\_\_\_\_  
A. Vertices  
**B. Edges**C. Both a and b above  
D. Labels

9. Key value pairs is usually seen in  
**A. Hash tables**B. Heaps  
C. Both a and b  
D. Skip list

10. Breadth First Search is used in  
A. Binary trees  
B. Stacks  
**C. Graphs**D. Both a and c above

11. Which of the following ways below is a pre order traversal?  
**A. Root->left sub tree->right sub tree**B. Root-> right sub tree ->left sub tree  
C. right sub tree->left sub tree->Root  
D. left sub tree->right sub tree->Root

12. What is the pecularity of red blac trees?  
A. In red-black trees, the root do not contain data.  
**B. In red-black trees, the leaf nodes are not relevant and do not contain data.**C. In red-black trees, the leaf node are relevant but do not contain data.  
D. Both a and c above

13. AVL trees have a faster \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
A. Insertion  
B. Deletion  
C. Updation  
**D. Retrival**

14. Which of the following statements hold true for binary trees?  
A. The left subtree of a node contains only nodes with keys less than the node’s key  
B. The right subtree of a node contains only one nodes with key greater than the node’s key.  
**C. Both a and b above**D. Noth left and right subtree nodes contains only nodes with keys less than the node’s key

15. Which of the following linked list below have last node of the list pointing to the first node?  
A. circular doubly linked list  
B. circular linked list  
**C. circular singly linked list**D. doubly linked list

16. Which of the following ways below is a In order traversal?  
A. Root->left sub tree->right sub tree  
B. Root-> right sub tree ->left sub tree  
C. right sub tree->left sub tree->Root  
**D. left sub tree->right sub tree->Root**

17. Can stack be describe as a pointer?  
**A. Yes**B. No

18. The time required in best case for search operation in binary tree is  
A. O(n)  
**B. O(log n)**C. O(2n)  
D. O(log 2n)

19. In \_\_\_\_\_\_\_\_\_\_\_\_\_\_ tree, the heights of two child subtree of any node differ by at most one  
A. Binary tree  
B. Red black tree  
C. Splay tree  
**D. AVL tree**

20. 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

21.Which of the following points is/are true about Linked List data structure when it is compared with array

A. Arrays have better cache locality that can make them better in terms of performance.

B. It is easy to insert and delete elements in Linked List

C.Random access is not allowed in a typical implementation of Linked Lists

D.The size of array has to be pre-decided, linked lists can change their size any time.

**E. All of the above**

22. Consider the following function that takes reference to head of a Doubly Linked List as parameter. Assume that a node of doubly linked list has previous pointer as prev and next pointer as next.

void fun(struct node \*\*head\_ref)

{

struct node \*temp = NULL;

struct node \*current = \*head\_ref;

while (current != NULL)

{

temp = current->prev;

current->prev = current->next;

current->next = temp;

current = current->prev;

}

if(temp != NULL )

\*head\_ref = temp->prev;

}

Assume that reference of head of following doubly linked list is passed to above function 1 <--> 2 <--> 3 <--> 4 <--> 5 <-->6. What should be the modified linked list after the function call?

1. 2 <--> 1 <--> 4 <--> 3 <--> 6 <-->5
2. 5 <--> 4 <--> 3 <--> 2 <--> 1 <-->6.
3. **6 <--> 5 <--> 4 <--> 3 <--> 2 <--> 1.**
4. D.6 <--> 5 <--> 4 <--> 3 <--> 1 <--> 2

23. Which of the following sorting algorithms can be used to sort a random linked list with minimum time complexity?

A. Insertion Sort

B. Quick Sort

C. Heap Sort

**D. Merge Sort**

24. The following function reverse() is supposed to reverse a singly linked list. There is one line missing at the end of the function.

/\* Link list node \*/

struct node

{

int data;

struct node\* next;

};

/\* head\_ref is a double pointer which points to head (or start) pointer

of linked list \*/

static void reverse(struct node\*\* head\_ref)

{

struct node\* prev = NULL;

struct node\* current = \*head\_ref;

struct node\* next;

while (current != NULL)

{

next = current->next;

current->next = prev;

prev = current;

current = next;

}

/\*ADD A STATEMENT HERE\*/

}

What should be added in place of "/\*ADD A STATEMENT HERE\*/", so that the function correctly reverses a linked list.

1. **\*head\_ref = prev;**
2. \*head\_ref = current;
3. \*head\_ref = next;
4. \*head\_ref = NULL;

25. 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);

}

1. 1 4 6 6 4 1
2. 1 3 5 1 3 5
3. C1 2 3 5
4. **1 3 5 5 3 1**

26. 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;

}

1. q = NULL; p->next = head; head = p;
2. q->next = NULL; head = p; p->next = head;
3. head = p; p->next = q; q->next = NULL;
4. **q->next = NULL; p->next = head; head = p;**

27. 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;

}

}

1. 1,2,3,4,5,6,7
2. **2,1,4,3,6,5,7**
3. C1,3,2,5,4,7,6
4. D.2,3,4,5,6,7,1

28. In the worst case, the number of comparisons needed to search a singly linked list of length n for a given element is (GATE CS 2002)

A. log 2 n

B. n/2

C. log 2 n – 1

D. **n**

29. Suppose each set is represented as a linked list with elements in arbitrary order. Which of the operations among union, intersection, membership, cardinality will be the slowest?

A. union only

B. intersection, membership

C. membership, cardinality

D. **union, intersection**

30. Consider the function f defined below.

struct item

{

int data;

struct item \* next;

};

int f(struct item \*p)

{

return (

(p == NULL) ||

(p->next == NULL) ||

(( P->data <= p->next->data) && f(p->next))

);

}

For a given linked list p, the function f returns 1 if and only if (GATE CS 2003)

1. the list is empty or has exactly one element
2. **the elements in the list are sorted in non-decreasing order of data value**
3. C. the elements in the list are sorted in non-increasing order of data value
4. not all elements in the list have the same data value.

31. A circularly linked list is used to represent a Queue. A single variable p is used to access the Queue. To which node should p point such that both the operations enQueue and deQueue can be performed in constant time? (GATE 2004) circularLinkedList

A. **rear node**

B. front node

C. not possible with a single pointer

D. node next to front

32. What are the time complexities of finding 8th element from beginning and 8th element from end in a singly linked list? Let n be the number of nodes in linked list, you may assume that n > 8.

A. **O(1) and O(n)**

B. O(1) and O(1)

C. O(n) and O(1)

D. O(n) and O(n)

33.Is it possible to create a doubly linked list using only one pointer with every node.

A. Not Possible

B. **Yes, possible by storing XOR of addresses of previous and next nodes.**

C. Yes, possible by storing XOR of current node and next node

D. Yes, possible by storing XOR of current node and previous node

34. 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 is not last node. Use following two steps (a) Copy the data of next of X to X. (b) Delete next of X.**

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. Use following two steps (a) Copy the data of next of X to X. (b) Delete next of X.

35. You are given pointers to first and last nodes of a singly linked list, which of the following operations are dependent on the length of the linked list?

A. Delete the first element

B. Insert a new element as a first element

C. **Delete the last element of the list**

D. Add a new element at the end of the list

36. 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;

}

}

Which of the following is FALSE about above function?

1. The function may crash when the linked list is empty
2. The function doesn't print the last node when the linked list is not empty
3. **The function is implemented incorrectly because it changes head**

37. In the worst case, the number of comparisons needed to search a singly linked list of length n for a given element is

A log2 n

B n/2

C log2 n - 1

D **n**