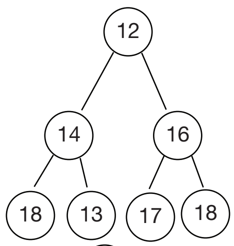
PROGRAMMING AND DATA STRUCTURES TEST 4

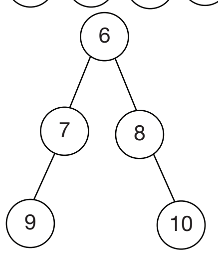
Number of Questions: 25 **Section Marks: 30**

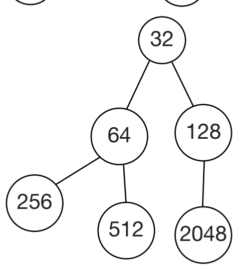
Directions for questions 1 to 25: Select the correct alternative from the given choices.

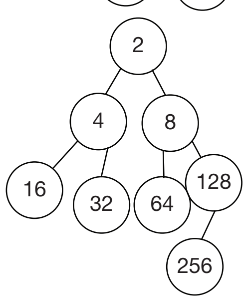
1. Which of the following data structure(s) can be used to represent Stack, Queue, Binary Tree, Hash Table, Heap and Graph?
(A) Array (B) Singly linked list
(C) Doubly linked list (D) Both (A) and (C)

2. Which of the following is a min heap?

(A)


(B)


(C)


(D)


3. Which traversal of a binary search tree will give the result in an ascending order of elements.
(A) Preorder
(B) In-order
(C) Post-order
(D) None of the above (a special code is required to print elements in ascending order)

4. Which of the following cannot be implemented using self-referential structure?
(A) Graphs (B) AVL trees
(C) Splay trees (D) B-Trees

5. Which of the following is TRUE about the AVL tree?
(1) AVL tree is a binary tree.
(2) The search time of AVL tree will be always less than (or) equal to the search time of binary search tree.
(3) AVL tree is binary search tree with maximum balancing factor of 2.
(A) Only (1) (B) Only (1) and (2)
(C) Only (1) and (3) (D) All the above

6. Which of the following is suitable for the data that contains a relationship between pairs of elements, which is not necessarily hierarchical in nature?
(A) Heaps (B) AVL Trees
(C) Graphs (D) B-Trees

7. What is the Number of Null links in a Binary Tree of 'n' nodes?
(A) n (B) $n + 1$
(C) $n - 1$ (D) $n/2$

8. What is the number of stacks that should be used to implement a queue?
(A) 1 (B) 2
(C) 3 (D) 4

9. Consider the recursive function for fibonacci sequence:

```
fib (n)
{
if (n ==0)
return 0;
else if (n==1)
return 1;
else
return fib (n – 1) + fib (n – 2);
}
```

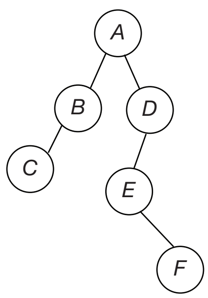
What is the Number of additions required, when *fib*(n) is invoked ?
(A) $fib(n + 1) - 1$ (B) $fib\ (n + 1) + 1$
(C) $fib\ (n + 1) - n$ (D) $fib\ (n + 1) + n$

10. Which of the following tree traversals are necessary to get the unique pattern of a tree structure?
(A) Only In-order
(B) In-order and pre-order
(C) Post-order and pre-order
(D) All the above

Common data for questions 11 and 12:

```
Consider the following routines:
struct Binary
{
struct Binary * Left ;
int data;
struct Binary * Right;
}
```

```
void F1 (struct Binary *t)
{
if (t)
{
printf("%d", t → data);
F2(t → Left);
F2(t → Right);
}
}
void F2(struct binary * t)
{
if (t)
{
F1(t → Left);
printf("%d", t → data);
F1 (t → Right);
}
}
Let us consider a tree
```



11. If $F1(t)$ is called on the above tree with t as root node (A), what is its output?

(A) *ABCDEF* (B) *ACBFED*
(C) *ABDCFE* (D) *ACBEFD*

12. When $F2(t)$ is called on this tree, with root as the parameter that is ($F2$(root)), what is its output?

(A) *BCADEF* (B) *BCDEFA*
(C) *DEFBCA* (D) *DEFABC*

13. What is the prefix expression of the expression $a + b * c/d \wedge e \wedge f - g$

(A) $-/ * + a\ b\ c$ d $\wedge \wedge e\ f\ g$
(B) $- +$ a/$* b\ c \wedge d \wedge e\ f\ g$
(C) $- + a \wedge \wedge / * b\ c\ d\ e\ f\ g$
(D) $\wedge \wedge + a / * b\ c\ d\ e - f\ g$

14. What does the function DO( ) does on a queue?

```
void DO( )
{
if (!Q is empty)
{
int temp = dequeue ( );
DO ( );
enqueue (temp);
}
}
```

(A) queue remains same
(B) elements will get reversed in a queue
(C) elements are removed and reinserted in the same queue
(D) queue remains empty

15. Fill in the blanks for the routine A( ).

```
A( ) will delete the elements in circular queue.
Initially Front = Rear = -1
A(Q, N, Front, Rear, Y) // N is size of a queue, Q is an array to store circular Queue.
{
if (Front == -1)
{
printf("Queue is underflow");
exit (1);
}
Y = Q(Front);
if(I)
Front = Rear = -1;
else
{
if (Front == N)
(II);
else
Front = Front + 1;
}
return Y;
```

(A) I. Front == Rear +1 II. Front = –1
(B) I. Front == Rear II. Front = –1
(C) I. Front == (Rear +1) II. Front = 0
(D) I. Front == Rear II. Front = 0

16. Consider the following routine:

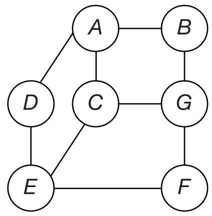
```
struct node
{
int data;
struct node → link;
};
void DO (struct node → p, struct node → q)
{
if (q)
{
DO (q, q → link);
q → link = p;
}
else
head = p;
}
```

When DO(head, head → link) is implemented on singly linked list, the resultant list,

(A) is a reversal of given list
(B) remains the same

(C) is a circular linked list
(D) None of these

17. Consider the graph below;



Which of the following orderings are not possible using depth first search graph traversal:

(A) *A C G F E D B* (B) *A D E F C G B*
(C) *A C G B F E D* (D) *C G B A D E F*

18. What is the time complexity for attaining maximum element from Max heap?

(A) $O(n \log n)$ (B) $O(n)$
(C) $O(\log n)$ (D) $O(1)$

19. Minimum number of nodes in AVL tree of height 5 is (with root at height = 1)

(A) 11 (B) 12
(C) 13 (D) 14

20. Consider a Binary search Tree '*T*' in which every node will be having only right child, there are '*n*' nodes \in tree '*T*'. Time complexity for searching an element will be:

(A) $O(\log n)$ (B) $\left(\log\left(\frac{n}{2}\right)\right)$
(C) $O(n)$ (D) $O\left(\frac{n}{2}\right)$

21. Consider post-order and in-order traversal of a tree:
POSTORDER :- DCBGFEA
INORDER :- BDCAFGE
When tree is constructed from above tree traversals then what are the leaf nodes?

(A) *D*, *G* (B) *C*, *F*
(C) *D*, *E* (D) *C*, *G*

Linked Answer for Questions 22 and 23:
Construct an AVL tree with the elements
21, 26, 30, 9, 4, 14, 28, 18

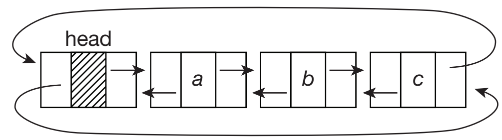
22. What will be the root node for the AVL tree?

(A) 26 (B) 21
(C) 28 (D) 18

23. What will be the post order predecessor of the root node, in the above resultant tree?

(A) 26 (B) 21
(C) 28 (D) 9

Linked Answer for Questions 24 and 25:
Consider the circular doubly linked list given below.



Consider the routine below:

```
struct node
{
sturct node *Lptr ;
int data;
struct node * Rptr;
};
void A ( )
{
struct node → t = head → Rptr;
while (t! = head)
{
swap (t → Lptr, t → Rptr)
t = t → Lptr;
}
swap (head → Lptr, head → Rptr);
}.
```

24. When *A*( ) routine is applied on the given list, the head Right pointer maps to:

(A) Node containing data field as '*a*'
(B) Node containing data field as '*b*'
(C) Node containing data field as '*c*'
(D) The head pointer points to itself

25. To which node *N* (pointer) points when the following operation is performed on the new list formed in the previous question.
(struct node $*$ *N*)
N = Head → Rptr → Rptr → Lptr → Lptr → Rptr;

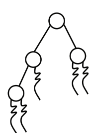
(A) Head node
(B) Node containing data field as '*a*'
(C) Node containing data field as '*b*'
(D) Node containing data field as '*c*'

Answer Keys

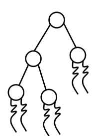
1. A	**2.** C	**3.** B	**4.** A	**5.** B	**6.** C	**7.** B	**8.** B	**9.** A	**10.** B
11. D	**12.** A	**13.** B	**14.** B	**15.** D	**16.** A	**17.** B	**18.** D	**19.** B	**20.** C
21. A	**22.** B	**23.** C	**24.** C	**25.** D					

Hints and Explanations

1. Array can be used to represent all the above data structures. Choice (A)
2. Min Heap is a complete Binary tree, with parent node value less than both the child nodes if they exist. Choice (C)
3. Inorder is the traversal of tree which prints data elements in an ascending order. Choice (B)
4. Graphs uses adjacency matrix (or) linked representation. Choice (A)
5. AVL tree is binary search tree with balancing factor as 0, +1, –1 and search time as $O(\log n)$. Choice (B)
7.



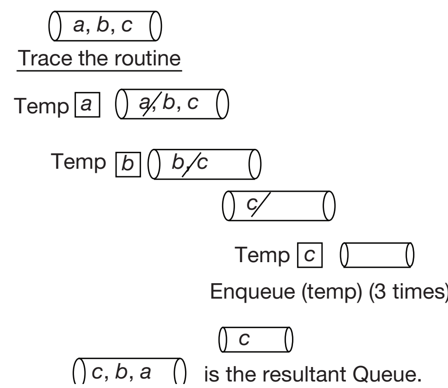
= 5 (for 4 nodes)



= 6 (for 5 nodes)

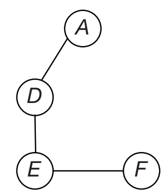
∴ For n nodes it will be $(n + 1)$ Null links. Choice (B)

8. 2 stacks required; use one for the Enqueue and the other for Dequeue. Choice (B)
9. Number of additions in a fibonacci sequence will be fib $(n + 1) - 1$. Choice (A)
10. For unique pattern of tree either (pre order + in order) or (post order + in order) are } necessary. Choice (B)
11. Traverse the routines on the tree. Choice (D)
12. Traverse the routines on the tree. Choice (A)
13. $a + b * c / d \wedge e \wedge f - g$
    $a + b * c / d \wedge \wedge e f - g$
    $a + b * c / \wedge d \wedge e f - g$
    $a + * b c / \wedge d \wedge e f - g$
    $a + / * b c \wedge d \wedge e f - g$
    $+ a / * b c \wedge d \wedge e f - g$
    $- + a / * b c \wedge d \wedge e f g.$
    Choice (B)
14. Let us consider a queue with 3 elements a, b, c.



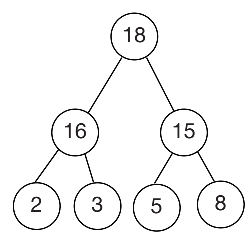
Choice (B)

15. Circular queue deletion routine. Choice (D)
16. It reverses the list. Choice (A)
17. Let us consider option (B)



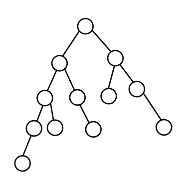
At (F) it can't reach to vertex (C) as it is not adjacent to (F). Choice (B)

18. Let us consider a max heap



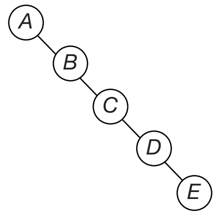
As the maximum element is at the root. To attain root element it takes $O(1)$ time. Choice (D)

19.



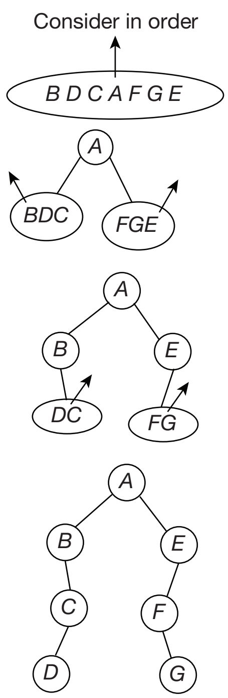
Choice (B)

20. Let us consider 5 nodes



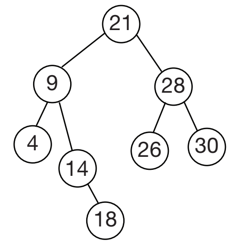
To search an element (E) it has to compare 5 elements
$\therefore$ it is $O(n)$. Choice (C)

21. POST: *DCBGFEA*
IN: *BDCAFGE*



Consider in order Choice (A)

22. From the above elements AVL tree will be:



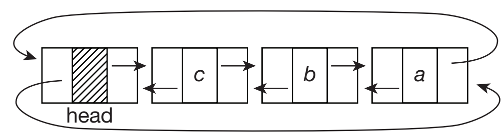
Choice (B)

23. Post order for the above tree is 4 18 14 9 26 30 28 21.
Choice (C)

24. The $A(\,)$ routine will reverse the given list
$\therefore$ Head pointer maps to node containing data field as 'c'.

Choice (C)

25.



Traversing the above list with given operation make to point to node containing c. Choice (D)