**University of Mumbai**

**Examination 2020 under cluster 4 (Lead College: PCE, New Panvel)**

Program: **Computer Engineering**

Curriculum Scheme: Rev2016

Examination: First/Second/Third/Final Year Semester III

Course Code: CSC 305 and Course Name: Data Structures

Time: 1 hour Max. Marks: 50

=====================================================================

For the students:- All the Questions are compulsory and carry equal marks .

|  |  |
| --- | --- |
| Q1. | An ADT consists of |
| Option A: | Data |
| Option B: | Operations |
| Option C: | Data and Operations |
| Option D: | Definitions of Operations |
|  |  |
| Q2. | The example of non-linear data structure is |
| Option A: | Stack |
| Option B: | Queue |
| Option C: | Array |
| Option D: | Tree |
|  |  |
| Q3. | The double ended queue which allows deletions at both ends of the list but insertion at only one end. |
| Option A: | Input restricted Queue |
| Option B: | Output restricted Queue |
| Option C: | Circular Queue |
| Option D: | Priority Queue |
|  |  |
| Q4. | Insertion and removal of element from stack are called \_\_\_\_\_\_\_\_\_ and \_\_\_\_\_\_\_\_\_\_ respectively. |
| Option A: | Pop , push |
| Option B: | Push, pop |
| Option C: | Insert, remove |
| Option D: | Remove, Insert |
|  |  |
| Q5. | A program for checking sequence of parentheses is balanced or not is written using stack. The content of the stack at the end of reading input string when the input given to the program is : (()(())(()) are: |
| Option A: | ( ) |
| Option B: | ( |
| Option C: | Empty stack |
| Option D: | Stack overflow |
|  |  |
| Q6. | A queue is a \_\_\_\_\_\_\_\_\_\_ data structure |
| Option A: | FIFO(First In First Out) |
| Option B: | LIFO(Last in First Out) |
| Option C: | Ordered Array |
| Option D: | Linear tree |
|  |  |
| Q7. | Give the Postfix form of following expression.  A + (E \* F) |
| Option A: | EF\*A+ |
| Option B: | AEF\*+ |
| Option C: | AE+F\* |
| Option D: | AEF\*+ |
|  |  |
| Q8. | The linked list in which each node of the list points to the previous node and next node of the list is |
| Option A: | Linear Linked list |
| Option B: | Circular Linked List |
| Option C: | Doubly Linked List |
| Option D: | Discrete Linked List |
|  |  |
| Q9. | Select the false statement about Linked List data structure when it is compared with array. |
| Option A: | Arrays have better cache locality that can make them better in terms of performance |
| Option B: | It is easy to insert and delete elements in Linked List |
| Option C: | Random access is not allowed in a typical implementation of Linked Lists |
| Option D: | Linked List is example of static memory allocation |
|  |  |
| Q10. | A pointer to a node N is given in a singly linked list. Head pointer is not given. Is it possible to delete the node N from given linked list? |
| Option A: | Possible if N is not last node |
| Option B: | Possible if size of linked list is even |
| Option C: | Possible if size of linked list is odd |
| Option D: | Possible if N is not first node |
|  |  |
| Q11. | Linked list is an example of \_\_\_\_\_\_\_\_\_ type of memory allocation. |
| Option A: | Static |
| Option B: | Dynamic |
| Option C: | Compile Time |
| Option D: | Static+dynamic |
|  |  |
| Q12. | One field of Linked list mode is called data field and another field is \_\_\_\_\_\_\_\_\_. |
| Option A: | Pointer to Character |
| Option B: | Pointer to Class |
| Option C: | Pointer to Integer |
| Option D: | Pointer to Node |
|  |  |
| Q13. | For any node x, every node in the left subtree of x is less than the value of x and every descendant node’s value in the right subtree is greater than the value x. This type of tree is called \_\_\_\_\_\_\_\_\_\_\_. |
| Option A: | Binary Tree |
| Option B: | Binary Search Tree |
| Option C: | Binary Heap Tree |
| Option D: | AVL tree |
|  |  |
| Q14. | Visiting the root node, then its left subtree and then its right subtree – this traversal in Binary tree is called |
| Option A: | Preorder Traversal |
| Option B: | Postorder Traversal |
| Option C: | In-order traversal |
| Option D: | Linear Traversal |
|  |  |
| Q15. | In ……………………………, for each node the left subtree and right subtree differ in height by at most 1 unit. |
| Option A: | Red and Black Tree |
| Option B: | Binary tree |
| Option C: | Binary Search Tree |
| Option D: | AVL tree |
|  |  |
| Q16. | The \_\_\_\_\_\_ traversal of the Binary Search tree will yield a sorted listing of elements of the tree. |
| Option A: | Preorder |
| Option B: | Postorder |
| Option C: | Inorder |
| Option D: | Preorder and postorder |
|  |  |
| Q17. | The preorder traversal of the following Binary tree is  fig1 |
| Option A: | ABCDEFGHIJK |
| Option B: | ABDHIECFGJK |
| Option C: | HIDEBFJKGCA |
| Option D: | IJKGHDEFABC |
|  |  |
| Q18. | The numbers 10, 11, 12, 7, 5, 1, 8, 3, 6, 0, 9, 4, 2 are inserted in that order into an initially empty binary search tree. The in-order traversal sequence of the resultant tree is |
| Option A: | 0 1 2 3 4 5 6 7 8 9 10 11 12 |
| Option B: | 7 6 5 4 3 2 1 0 8 9 10 11 12 |
| Option C: | 9 8 7 6 5 4 3 2 1 0 11 10 12 |
| Option D: | 0 1 2 3 4 5 7 8 9 6 11 12 10 |
|  |  |
| Q19. | Topological sort starts from a node which has \_\_\_\_\_\_\_\_\_\_ |
| Option A: | Maximum in degree |
| Option B: | Minimum in degree |
| Option C: | Zero in degree |
| Option D: | Maximum Out degree |
|  |  |
| Q20. | In Depth First Search of Graph, which of the following data structure is used? |
| Option A: | Stack |
| Option B: | Queue |
| Option C: | Linked list |
| Option D: | Array |
|  |  |
| Q21. | In the following graph, possible DFS traversals among the following sequences  **I) a b e g h f II) a b f e h g III) a b f h g e IV) a f g h b e** |
| Option A: | I, II and IV only |
| Option B: | I and IV only |
| Option C: | II, III and IV only |
| Option D: | I, III and IV only |
|  |  |
| Q22. | Which of the following is not a limitation of binary search algorithm? |
| Option A: | must use a sorted array |
| Option B: | requirement of sorted array is expensive when a lot of insertion and deletions are needed |
| Option C: | there must be a mechanism to access middle element directly |
| Option D: | binary search algorithm is not efficient when the data elements more than 2000 |
|  |  |
| Q23. | The sorting algorithm which uses of divide and conquer technique is |
| Option A: | Bubble sort |
| Option B: | Insertion Sort |
| Option C: | Merge Sort |
| Option D: | Selection sort |
|  |  |
| Q24. | The O(1) searching technique is |
| Option A: | Binary search |
| Option B: | Linear search |
| Option C: | Hashing |
| Option D: | Tree search |
|  |  |
| Q25. | Give the output of insertion sort after second pass for the following elements:  56, 8, 64, 51, 32, 21 |
| Option A: | 8, 21, 32, 56, 51, 64 |
| Option B: | 8, 56, 64, 51, 32, 21 |
| Option C: | 56, 64, 8, 21, 32,51 |
| Option D: | 8, 56, 21, 32, 51, 64 |