|  |  |
| --- | --- |
|  | **NATIONAL INSTITUTE OF TECHNOLOGY, WARANGAL**  (An Institution of National Importance)  DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING  MID SEMESTER EXAMINATIONS, October/November, 2020  II B. Tech (Computer Science & Engineering) I Semester  **CS202 DATA STRUCTURES AND ALGORITHMS**  Date: 27 – 10 - 2020 Max. Marks: 30  N.B.: Answer ALL questions  **All programs/functions/code segments are to be written in C** |
| 1. | Design a Stack structure where a stack element can be a Linked list (data value as a character). Write a complete program which reads few elements(LLs) onto the Stack, then takes out the elements and prints them. |
| 2. | Design a Queue structure where a queue element can be a Linked list (data value as a integer). Write a complete program which enters few elements into the queue , then takes out the elements and prints them. |
| 3. | Write a complete program to implement k queues using a single linked list (data values as integers). In the main() function include logic for entering element on to a particular queue and deleting an element from a particular queue. |
| 4. | A queue Q of integer elements is given as input. Write functions/code segments to construct a Linked list with even elements and a complete binary tree with odd elements of the Q.  Example : Input : Q elements : 2 5 8 4 9 1 6 3 7  Output :  A linked list with even elements L : 2 --- 8 --- 4 --- 6  A complete binary tree with odd elements T :  5  / \  9 1  / \  3 7 |
| 5. | Define the node structure of a Kd tree, where the d key values(int) are stored in a linked list(not as array key[]). Write functions for adding nodes to such tree and printing the tree in inorder. ( No need of full program or main() function ) |
| 6. | Given a linked list L of nodes with data values either integers or characters. The list L is formed is such a way that there will be series of integers and characters as shown below.  Write code segment to create two linked lists N and C as shown below. The N list nodes points to the start of series of integers in L , whereas the C list nodes points to the start of series of characters in L. Write functions for printing N and C. |
| 7. | Write code segment to go through the given constructed MLL ONLY ONCE and print the descending sorted order of data values. You should not use the logic of first sorting values in ascending order in an array and then printing them in reverse order.  Example :  Input :MLL:       3   –   7 –       10   –  36                        |          |             |                        8        12         18                        |           |            |                      15       45       72    Output: 72 45 36 18 15 12 10 8 7 3 |
| 8. | Assume that a binary search tree has to be created in the same shape only as shown below. Write code segment which produces all the possible input sequences for creating such a binary search tree. First mention the logic then write code. You can assume that the binary search tree has already been created using one of the input sequences and is pointed by T.  Binary Search Tree All possible input sequences  9 9, 5, 7, 15, 12  / \ 9, 5, 15, 7, 12  5 15 9, 5, 15, 12, 7  \ / 9, 15, 12, 5, 7  7 12 9, 15, 5, 12, 7  9, 15, 5, 7, 12 |
| 9. | Write a function/code segment to check if two nodes are on the same path from root of a given binary tree. You should traverse the tree only once. |
| 10. | Find least common ancestor ( LCA) of three nodes of a given binary tree ( efficient way) |
| 11. | Front view: Three Binary trees pointers are given as T1 , T2 , T3 . They are placed each other as: T2 is placed on T1 and T3 is placed on T2. Write code segment to print the front view of node values as explained in the example below.  You are free to print Front view output in your own choice of  level order as: 9 4 1 2 8 5 7 7 3 6 18 (or)  preorder as : 9 4 2 7 3 8 6 1 5 7 18 (or any order of your choice. You need not construct a new Tree) |
| 12. | Assume that four linked lists L1, L2, L3 and L4 are given and also L2, L3, L4 are meeting at different nodes of L1 as shown in figure below. The linked lists contain unique data values. Write code segment to print the meeting point information as below:  Input: Given L1 , L2 , L3 , L4 as shown  Output (you can choose your own sequence, but your output should contain the below three lines):  L2 meets L1 L3 L4 at 36  L4 meets L1 L3 at 27  L3 meets L1 at 5 |
| 13. | Assume that a binary tree (data values as integers) has been created and pointed by T.  Write a function (code segment) which prints those levels of the tree where the data values of that level is in the order of increasing, decreasing or decreasing, increasing. Example1: output could be (a level) : 2 6 4 8 3 9 ( increasing , decreasing ie. 2 , 6 increase, 6 , 4 decrease, 4 , 8 increase , 8, 3 decrease , 3 , 9 increase )  Example2 : output could be (a level) : 5 2 7 4 8 3 ( decreasing , increasing ) |
| 14. | Write efficient code segment to find k1 + k2 + k3 + k4 = s , whereas k1 from A1 , k2 from A2, k3 from A3, and k4 from A4 arrays of integers. You should not fully sort the arrays in first instance. There can be n number of arrays where k1+k2+k3+…kn = s. |
| 15. | Write code segment to merge n Binary Search Trees efficiently. |
| 16. | Write code segment/functions to construct Huffman code tree with given frequencies of characters. You **must use only one node structure (only one struct) in the entire code**. You should not use Arrays, Stacks, and Queues. ( no STL) |
| 17. | Write efficient code segment to find k1+k2+k3 = s , whereas k1 , k2, k3 are in a single given Binary Search Tree. You should not use the logic of storing inorder traversal of BST in an array and finding for k1, k2, k3. You can modify the BST and you should not use Arrays, Stacks, and Queues. ( no STL) |
| 18. | Write code to print the data values of nodes in the middle level of a fully complete binary tree without traversing the entire tree. |