

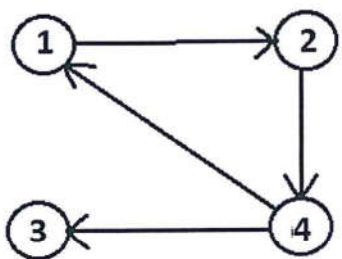


**DECEMBER 2020: END SEMESTER ASSESSMENT (ESA) B TECH 3<sup>rd</sup> SEMESTER**

**UE17/18CS202 – Data Structures**

Time: 3 Hrs	Answer All Questions	Max Marks: 100
-------------	----------------------	----------------

Note: C code should be given for implementation based questions. State any assumptions made.

1	a)	Define Data Structure. List 2 linear and 2 non linear data structures. Write one application of each of them.	4
	b)	Write a Recursive C function to find the sum of n elements in an array.	4
	c)	Given a pointer to the first node of a singly linked list with the node structure as follows: struct node { int data; struct node* next; }; i) Write a function to insert a node with data x in the beginning of the list ii) Write a function to insert a node with data x at the end of the list	7 (3+4)
	d)	Given the address of the first node of a circular doubly linked list, write a function to delete the last node.	5
2	a)	Convert the following infix expression to postfix expression. $A + (((B - C) * (D - E) + F) / G)$	5
	b)	Write a C function to evaluate the given postfix expression and returns the evaluated value. Use stack functions push, pop. Assume that the function with prototype: int isoper(char); returns 1 if the character is an operator and 0 if operand. You need not write stack functions and isoper function. The prototype for the function to be implemented: int postfixEval(char *postfix);	6
	c)	Define Queue data structure. Write the difference between an ordinary queue and a circular queue. Write 2 applications of Queue data structure.	5
	d)	Write a function called AlternateDivide to divide an integer queue into two queues which contain alternate elements of the original queue. Use only queue functions Insert, remove and isempty. For eg. if original queue Q is (10,20,30,40,50) then Q1 is (10,30,50) and Q2 is (20,40). You do not have to write insert, remove and isempty functions. The prototype for the function to be implemented: AlternateDivide(int *Q, int *Q1, int *Q2);	4
3	a)	Write the adjacency matrix for the digraph given below. 	2

	b)	Write a C function to implement Depth First Search for a graph represented using adjacency Matrix.	5
	c)	For the tree given below, write the height and depth of the Tree and also write all the leaf nodes.  <div style="text-align: center;"> <pre> graph TD     A((A)) --- B((B))     A --- C((C))     C --- D((D))     C --- E((E))     D --- F((F))     D --- G((G)) </pre> </div> <p>Define the following terms:  i) Binary Tree (ii) height of a tree (iii) depth of a tree (iv) leaf node.</p>	7
	d)	Write the pre order, in order and post order traversal of the binary tree given in the question 3c	6
4	a)	Write a C function to compute the smallest element in a Binary Search Tree.	2
	b)	Construct a Binary Search Tree with the following numbers inserted into an empty binary search tree in the given order: 10, 2, 1, 3, 5, 15, 12, 16. Reconstruct the tree after deleting the node with info value 10.	6
	c)	Construct an expression tree from the following postfix expression: abc+d*+e-	6
	d)	Define Heap. Write the algorithm for bottom-up heap construction.	6
5	a)	Write a C function to insert words consisting of characters of lower case into a trie. Consider the following structure of the trie node <pre> struct trienode {     struct trienode *child[26];     int endofword; } </pre> Function prototype to be implemented : insert(struct trienode *root, char *key).	6
	b)	Write any 2 applications of Trie.	2
	c)	Suppose a hash table contains HASHSIZE =11 entries indexed from 0 to 10 and that the following keys are to be mapped into the table. 10,100,32,45,59,126,3,29,200 Determine the hash addresses and find how many collisions occur when the hash function <b>key % HASHSIZE</b> is used. Show the contents of the hash table when the collision is resolved using linear probing.	8
	d)	Explain Quadratic Probing and Double Hashing.	4