

Total No. of Questions – []

Total No. of Printed Pages 02

G.R. No.	
----------	--

Paper Code - U228-135 (ESE)

MAY 2019/ENDSEM
S. Y. B. TECH. (E&TC) (SEMESTER -II)

COURSE NAME: Data Structures

COURSE CODE: ETUA22175

(PATTERN 2017)

Time: [2 Hours]

[Max. Marks: 50]

(*) Instructions to candidates:

- 1) Answer Q.1, Q.2, Q.3, Q.4, Q.5 OR Q.6, Q.7 OR Q.8
- 2) Figures to the right indicate full marks.
- 3) Use of scientific calculator is allowed
- 4) Use suitable data where ever required

Q. 1 a) Define recursion with example in detail. [6]

OR

b) Define data structures. Write types of data structures with example [6]

Q. 2 a) Write a C function to implement linear search [6]

OR

b) Sort the following data using merge sort [38, 27, 43, 03, 09, 82, 10] [6]

Q. 3 a) Explain circular linked list. Differentiate between SLL and CLL. [6]

OR

b) Write a C function to create linked list. [6]

Q. 4 a) Define queue and write a function for addqueue operation in queue using array. [4]

OR

b) Convert following infix expression to postfix form [4]
(A+(B*C/D)-E)

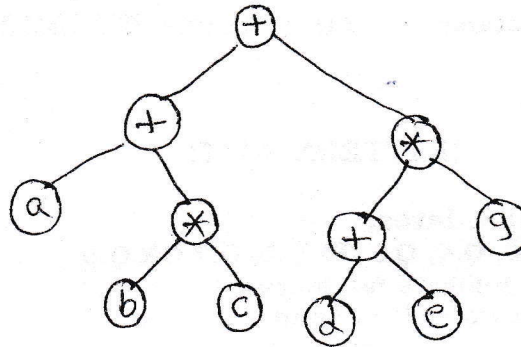
Q. 5 a) Explain memory representation of binary tree with suitable example. [6]

- b) Construct a binary tree using following tree traversals [4]
 Post-order: D,F,E,B,G,L,J,K,H,C,A
 In-order: D,B,F,E,A,G,C,L,J,H,K

- c) Write C function to perform search operation in BST [4]

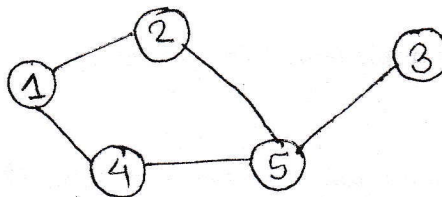
OR

- Q. 6 a) Construct an infix expression from given expression tree [6]



- b) Write a recursive function for post-order traversal of BST. [4]
 c) Create BST from given data [20,10,5,1,7,15,30,25,35,32,40] [4]

- Q. 7 a) Explain following for given graph [6]
 1) adjacency matrix representation
 2) adjacency list representation



- b) Explain Breadth first search traversal algorithm. [4]
 c) Write a C function for indegree of a vertex [4]

OR

- Q. 8 a) Write a C function for implementing breadth first search for a graph. [6]
 b) Differentiate between linear and non-linear data structure. [4]
 c) Find minimum spanning tree for the given graph using prims algorithm. [4]

