

# CBCS SCHEME

18CS32

## Third Semester B.E. Degree Examination, Aug./Sept.2020 Data Structures and Applications

Time: 3 hrs.

Max. Marks: 100

Note: Answer any FIVE full questions, choosing ONE full question from each module.

### Module-1

- 1 a. Define data structures. List and explain the different operations that can be carried on arrays. (10 Marks)
- b. Define pointers. List the advantages of pointers over arrays. (04 Marks)
- c. Define dynamic memory allocation. List and write with explanation the syntax of dynamic memory allocating functions. (06 Marks)

OR

- 2 a. Define strings. List and explain any 5 operations with example. (12 Marks)
- b. Is it possible to store the contents of an array into a points? Justify your opinion and with suitable C-statements. (08 Marks)

### Module-2

- 3 a. Define a stack. Explain the different operation that can be performed on stack using C-functions and show them using diagrammatic representations. (10 Marks)
- b. Write an algorithm to convert a parenthesized infix expression to postfix. Apply the algorithm and show the contents of stack during conversion for the expression :  
 $(A + B * C) * ((D + E - F)/J)$ . (07 Marks)
- c. Differentiate recursion and iteration process. (03 Marks)

OR

- 4 a. Write a C-recursive function for
  - i) Adding n-odd natural numbers
  - ii) Adding n-even natural numbers. (08 Marks)
- b. Define a queue. List the different types of queues. State the limitation of ordinary queue. Explain how do you overcome the limitation by specifying the required C-statements and diagrammatic representation using an example. (12 Marks)

### Module-3

- 5 a. With the C-statements, explain how do you create a node, add and delete on Singly Linked List (SLL) with proper message where each node is containing the details of employee in the form of EmpId, EmpName, Empaddr and Empsalary as data fields. (10 Marks)
- b. Write and explain how do you implement the operations of stack using Singly Linked List (SLL) with the help of C-statements. (10 Marks)

OR

- 6 a. Differentiate Single (SLL) and Doubly (DLL) linked lists. (04 Marks)
- b. State the advantage of Doubly Linked List over Singly Linked List. (02 Marks)
- c. Implement addition and deletion of a NODE on a Doubly Linked List (DLL) with required C-statements. (14 Marks)

Important Note : 1. On completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages.  
2. Any revealing of identification, appeal to evaluator and/or equations written eg. 42+8=50, will be treated as malpractice.

**Module-4**

- 7 a. Define a binary tree. Explain how do you construct and add a NODE to binary tree using C-statements. Also explain how do you represent a binary tree using arrays. (09 Marks)
- b. Define binary tree traversal method. List and explain the different binary tree traversal methods along with C-functions. (08 Marks)
- c. Find the INORDER, PREORDER and POSTORDER for the following :

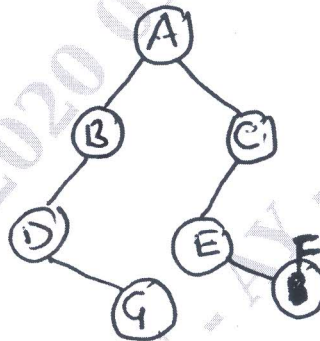


Fig.Q7(c)

(03 Marks)

OR

- 8 a. Define expression tree. Using a C-function, explain how do you construct a expression tree. Construct an expression tree for :  $a + b * c / f^g - h$ . (10 Marks)
- b. With diagrammatic explanation, explain how do you create and construct a BST. Also write C-functions for the same. (10 Marks)

**Module-5**

- 9 a. Define a graph and its traversal methods. List and explain the different graph traversal methods. Find the resultants of the types of graph traversal methods on the following graph : (consider 'a' as starting vertex).

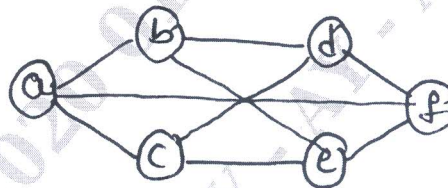


Fig.Q9(a)

(12 Marks)

- b. Write address calculation sort algorithm. Sort Z, A, P, B, Q, I, J, K using the address calculation sort algorithm. (08 Marks)

OR

- 10 a. Define file. List basic file operations. Explain any four operations with syntax and example. (10 Marks)
- b. Define Hashing. Explain the method of sorting data using a Hash function in a Hash table. Identify the problem that occurs during the value storage. Explain how do you resolve the problem using Hashing technique. (10 Marks)

\*\*\*\*\*