0073048

15221

Reg. No. | | | | | | | | | |

II Semester B.C.A. Degree Examination, September - 2021

COMPUTER SCIENCE

Data Structures

(CBCS Scheme Freshers)

Paper : BCA-203

Time : 3 Hours **Maximum Marks : 70**

Instructions to Candidates:

Answer **ALL** Sections.

SECTION - A

I. Answer any **TEN** questions. Each question carries **2** Marks. **(10×2=20)**

1. Differentiate between Data and Information.
2. What is Big O Notation?
3. Define Sorting.
4. List the basic operations carried out in a Linked List.
5. Define a Stack.
6. What is Circular Queue?
7. What is Graph?
8. State any two Properties of a Binary Tree.
9. Convert the infix expression (A-B/C)*(D/E-F) into a Postfix.
10. Give the node structure of Circular Linked List.
11. Mention any two disadvantages of an array.
12. Define Binary Search Tree.

[P.T.O.

SECTION - B

II. Answer any **FIVE** questions. Each question carries **10** Marks. **(5×10=50)**

13. a) Explain the Classification of Data Structures. **(6)**

    b) Briefly Explain any four String Handling Functions. **(4)**

14. a) Write an algorithm for Binary Search. **(5)**

    b) Explain bubble sort technique with an example. **(5)**

15. a) Explain the Steps involved in Creating a Linked List. **(5)**

    b) Write a C function to insert a node into a sorted Linked List. **(5)**

16. a) What are Stack Operations? Explain. **(5)**

    b) Evaluate the following Postfix Expression assuming A=1, B=2 and C=3
    ABC+*CBA-+* **(5)**

17. a) Write an algorithm for deletion of an element from queue. **(5)**

    b) Write a C Program for Towers of Hanoi Problem **(5)**

18. a) Explain Preorder, Postorder and inorder traversal of a Binary Tree with examples. **(6)**

    b) Explain malloc() and calloc() with Suitable Illustrations. **(4)**

19. a) Write depth first search algorithm to traverse a Graph. **(5)**

    b) Compare Linear Linked List and Doubly Linked Lists. **(5)**

20. a) Write recursive functions for tree traversals. **(5)**

    b) Write a C Program to Sort a List of N elements using quick sort algorithm. **(5)**
