

C

(Printed Pages 4)

Roll No. _____

20/1087

B.C.A. Third Semester Examination, 2020

Second Paper

(Data Structure Using C & C++)

Time : Three Hours

Maximum Marks : 75

Note : Attempt any **five** questions. **All** questions carry equal marks. The answers to short questions should not exceed 200 words and the answers to long questions should not exceed 500 words.

1. (a) How is a 2-dimensional array represented in memory? Discuss with example. 7+8
- (b) Write a short note on Tridiagonal matrices and their applications in computer science.

P.T.O.

20/1087

2. (a) Explain how we can represent a stack using : 5+5+5
 - (a) Array
 - (b) Linked list.
- (b) Write the algorithm for converting
 - (i) infix to postfix
 - (ii) Prefix to postfix
- (c) What are the applications of the stack data structure in Computer Science?
3. (a) What do you understand by : 8+7
 - (i) Linear linked list
 - (ii) Doubly linked list? Write the procedure to insert an element in the middle of a doubly linked list.
- (b) Differentiate between queue and Deque. Give applications of each.
4. Differentiate between: 3+4+4+4
 - (a) Linear array and linked list
 - (b) Inorder and Preorder Tree Traversal

20/1087

- (c) Binary Trees and B-Trees
(d) Hashing and Binary search
5. Write the procedure / steps for performing the following tasks: 5+5+5
- (a) Removal of duplicate elements from a sorted linked list.
- (b) Construction of balanced binary search tree from given keys.
- (c) Constructing an expression tree for an infix expression.
6. Write the algorithm for heapsort and trace it for the following list of numbers: 15
8, 15, 18, 3, 16, 0, 35, 4, 7, 63
What is the time complexity for performing merge sort on a list of size N.
7. Perform Insertion sort on the following list:
3, 7, 19, 5, 64, 128, 32, 14, 10, 58
What is the complexity of this algorithm in the :
(i) average case
(ii) Best case

20/1087

8. Write short notes on: 5+5+5
- (a) Sparse Arrays
(b) Priority Queue
(c) Indexing using B-Trees
9. Write a C program for (any **two**): 7½×2
- (a) Creating a B-Tree
(b) Traversing a linked list
(c) Binary Search