[Total No. of Questions - 9] [Total No. of Printed Pages - 2]
(2125)

15607

MCA 2nd Semester Examination

Data Structures (C++ and Java) (NS)

MCA-201

Time : 3 Hours **Max. Marks : 60**

The candidates shall limit their answers precisely within the answer-book (40 pages) issued to them and no supplementary/continuation sheet will be issued.

Note : Candidates are required to attempt five questions in all selecting one question from each of the sections A, B, C and D of the question paper and all the subparts of the question in section E.

SECTION - A

1. What are the different control structures available in C++? Explain with examples. (12)

2. Define sparse matrix. What are the different methods of representing a sparse matrix? Explain with examples. (12)

SECTION - B

3. Write an algorithm for implementing quick sort using stacks. (12)

4. Write an algorithm for implementing circular queue. (12)

SECTION - C

5. (a) Define threaded binary tree. (4)

   (b) Write and explain the different tree traversal algorithms. (8)

[P.T.O.]

6. (a) Write the methods of representing graphs. (4)

   (b) Write the different algorithms used for graph traversal. (8)

SECTION - D

7. What are the main file organization techniques? Explain the features of each type of technique. (12)

8. Write the algorithm for implementing radix sort on strings. (12)

SECTION - E

9. All questions are compulsory.

   (a) Define encapsulation.

   (b) What is the data type and size of a pointer?

   (c) Distinguish between arrays and linked list.

   (d) What is time space trade off?

   (e) How does stack facilitate simulation of recursion?

   (f) Convert the following expressions to reverse polish notations:

   (i) a + b * c + d

   (ii) a * b + c * d.

   (g) Define binary search tree.

   (h) Write whether following statements are true or false i. Every tree is a graph, ii. Every graph is not a tree.

   (i) Name the auxiliary data structures used for breadth first search and depth first search respectively.

   (j) Distinguish between a full and a complete binary tree.

   (k) Write the average and worst case complexities of quick and merge sort algorithms.

   (l) Define index sequential access method. (1×12=12)