4235

C14–CM–305/IT–**305**

4235

BOARD DIPLOMA EXAMINATION, (C–14)

OCT/NOV—2018

DCME—THIRD SEMESTER EXAMINATION

DATA STRUCTURES THROUGH C

Time : 3 hours] [*Total Marks* : 80

PART—A 3×10=30

Instructions : (1) Answer **all** questions.

(2) Each question carries **three** marks.

(3) Answers should be brief and straight to the point and shall not exceed *five* simple sentences.

1. Define data structure. List any four data structures.

2. Classify data structures. Give examples in each case.

3. Write the operations applied on singly linked list.

4. Define doubly linked list. Give example.

5. Define overflow and underflow on stacks.

6. Define queue. Give at least two day-to-day life examples.

7. Write the code in C to represent the node of the binary tree.

8. Define binary trees and draw a binary tree with at least 7 nodes.

9. What is sorting? Give its importance.

10. Distinguish between linear search and binary search.

PART—B 10×5=50

Instructions : (1) Answer *any* **five** questions.

(2) Each question carries **ten** marks.

(3) The answers should be comprehensive and the criterion for valuation is the content but not the length of the answer.

11. Write a C program to perform insertion and deletion operations on doubly linked list.

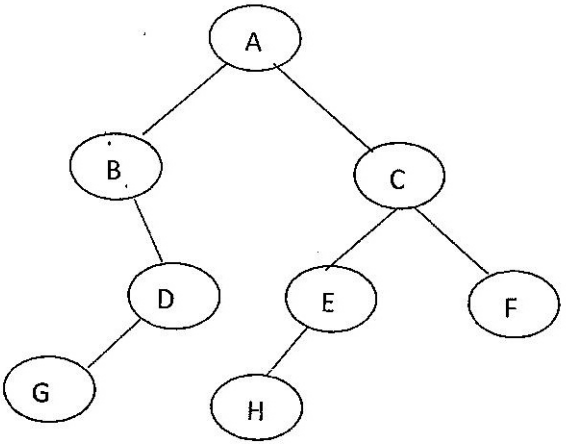
12. Write an algorithm/procedure to create singly linked list. Draw a singly linked with five nodes.

13. *(a)* What is sparse matrix? Explain the creation of sparse matrix with example.

(b) Write the algorithms for PUSH and POP operations.

14. Explain enqueue and dequeue operations on linear queues with examples.

15. Give the pre-order, in-order, post-order traversals for the following binary tree.



16. *(a)* Construct a binary tree for the following in-order and post-order :
In-order : 2*3+8/4
Post-order : 2 3*×84/+

(b) Can we construct a binary tree using pre-order and post-order expressions? Justify your answer.

17. Explain quick sort with an example.

18. *(a)* Write a C program for implementing the linear search.

(b) Write the algorithm of insertion sort.

★ ★ ★