

**GANPAT UNIVERSITY****B. TECH SEMESTER: IV (Computer Engineering / Information Technology)****Regular Examination April - June 2017****2CE404/2IT404: Data Structures****Time: 3 Hours]****[Total Marks: 60****Instructions:**

1. Figures to the right indicate full marks.
2. Each section should be written in a separate answer book.
3. Be precise and to the point in your answer.
4. Assume suitable data if required.

**SECTION-I**

- Q-1** [A] What is a pointer? How do you declare and initialize the pointers? [2]  
 [B] Explain implementation of priority queue. [3]  
 [C] Write a C program to insert data to the immediate right of the  $k^{\text{th}}$  node in the singly linked list. [5]

**OR**

- Q-1** [A] Write appropriate structure definition and variable declarations to store following information about 100 students: [2]  
 Name, Gender, Date of birth and marks in three subjects S1, S2 & S3. Date of birth should be a structure containing fields day, month and year.  
 [B] Assume that size of circular queue is 8 elements. Find the number of elements in circular queue for the following cases (Assume index of first element is 0) : [3]  
     **Front =3, Rear = 6**  
     **Front =5, Rear = 3**  
     **Front =7, Rear = 5**  
 [C] What is a circular queue? Write a user define function to implement insert and delete operation on circular queue. [5]

- Q-2** [A] Define stack. Write an algorithm to perform basic operation on stack. [5]  
 [B] Write a C program to implement doubly linked list with following operation: [5]  
     1) InsertBegin                      2) DeleteEnd

**OR**

- Q-2** [A] Convert each of the following infix expressions into its postfix and prefix form. [4]  
     1)  $A*(B+D)/E-F*(G+H/K)$   
     2)  $((A+B)-(C-D)*F/(G-H*F))$   
 [B] Write an algorithm to convert infix to postfix expression. Also convert the given infix expression  $A+B*C/D$  into postfix expression and evaluate using stack simulation. (A=10, B=2, C=15, D=5) [6]  
**Q-3** [A] What is recursion? Explain efficiency of recursion. Write a C recursive program to solve Tower of Hanoi problem. [5]  
 [B] Explain Circular linked list and Doubly linked list with suitable diagram. [5]

**[P.T.O]**

## SECTION-II

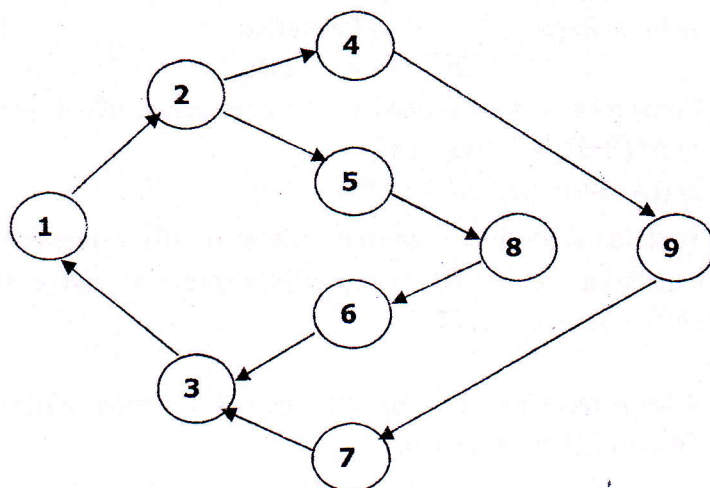
- Q-4 [A] Differentiate binary tree and simple tree. Construct binary tree using following inorder and postorder traversal sequences: [5]  
**Inorder: A, B, C, D, E, F, G, H, I**  
**Postorder: A, C, E, D, B, H, G, I, F**
- [B] Write an algorithm to perform simple merge and two way merge sort. [5]

OR

- Q-4 [A] Perform sorting of following data using heap sort algorithm: [5]  
**32, 52, 63, 84, 25, 100, 93, 74, 50, 39**
- [B] How binary search is differing from sequential search? Write a user defined function of binary search and sequential search in C. [5]
- Q-5 [A] What is binary search tree? Construct binary search tree using following data: [4]  
**70, 23, 82, 25, 40, 78, 111, 23, 28, 57, 80, 44**  
After construction delete node **25** from binary search tree and reconstruct it.
- [B] Perform sorting of following data using bubble sort algorithm: [3]  
**70, 103, 37, 114, 59, 26, 81, 92, 43, 32, 22**
- [C] Define following terms: 1) Full binary tree 2) Strict binary tree 3) Sibling [3]

OR

- Q-5 [A] Explain shell sort algorithm with example and also write a user define function of shell sort algorithm in C. [4]
- [B] Explain any three properties of binary tree using proper diagram. [3]
- [C] Define following terms: [3]  
1) Complete binary tree 2) Isolated vertex 3) Complete graph
- Q-6 [A] Construct expression tree for the given postfix expression : **123+\*45+6\*7/+** [3]
- [B] Consider binary tree with three nodes A, B, and C. How many binary trees are possible in which preorder sequence is "ABC", post order sequence is "CBA" and inorder sequence is 'BCA'. Construct all binary trees in which preorder sequence is "ABC", postorder sequence is "CBA" and inorder sequence is "BCA". [2]
- [C] Differentiate BFS and DFS. Write BFS and DFS traversal sequence for below directed graph. Starting with vertex labeled 1. [5]



END OF PAPER