

CBCS Scheme

USN

--	--	--	--	--	--	--	--	--	--

15CS33

Third Semester B.E. Degree Examination, June/July 2017

Data Structure and Applications

Time: 3 hrs.

Max. Marks: 80

Note: Answer FIVE full questions, choosing one full question from each module.

Module-1

- 1 a. Write a C program with an appropriate structure definition and variable declaration to read and display information about 5 employees using nested structures. Consider the following fields like Ename, Empid, DOJ (Date, Month, Year) and Salary (Basic, DA, HRA). (08 Marks)
- b. Give ADT of sparse matrix and show with a suitable example sparse matrix representation storing as triples. Give a sample transpose function to transpose sparse matrix. (08 Marks)

OR

- 2 a. What is a polynomial? What is the degree of the polynomial? Write a function to add two polynomials. (08 Marks)
- b. List and explain the functions supported by C for dynamic memory allocation. (04 Marks)
- c. Write a C program to concatenate Fname and Lname of a person without using any library function. (04 Marks)

Module-2

- 3 a. Define stack and write the ADT of stack. Implement push and pop functions for stack using arrays with StackFull and StackEmpty conditions. (08 Marks)
- b. What is an input restricted double ended queue? Implement the same with the supporting functions. (08 Marks)

OR

- 4 a. Write the postfix form of the following expression using stack:
i) $(a + b) * d + e / (f + a * d) + c$ ii) $((a/(b - c + d)) * (e - a) * c)$ (04 Marks)
- b. Write a function to evaluate a postfix expression and trace the same for the expression $a/b - d e * + a c *$ where $a = 6, b = 3, c = 1, d = 2, e = 4$. (06 Marks)
- c. Explain with a suitable example, how would you implement circular queue using dynamically allocated arrays. (06 Marks)

Module-3

- 5 a. Give the node structure to create a linked list of integers and write C functions to perform the following:
i) Create a three node list with data 10, 20 and 30.
ii) Insert a node with the data value 15 in between the nodes having the data values 10 and 20.
iii) Delete the node whose data is 20.
iv) Display the resulting singly linked list. (10 Marks)
- b. Write a node structure for linked representation of polynomial. Explain the algorithm to add two polynomials represented using linked list. (06 Marks)

OR

- 6 a. Write C functions to perform the following:
- Reversing a singly linked list.
 - Concatenating singly linked list.
 - Finding the length of the list. (06 Marks)
- b. List out the difference between the doubly linked list and singly linked list. Illustrate with example the following operations on a doubly linked list:
- Inserting a node at the beginning.
 - Inserting at the intermediate position.
 - Deletion of a node with a given value.
 - Search a key element. (10 Marks)

Module-4

- 7 a. Define binary trees. Explain the following with example:
- Complete binary tree
 - Skewed binary tree
 - Almost complete binary tree
 - Degree of a binary tree. (09 Marks)
- b. For the given data, draw a binary search tree and show the array and linked representation of the same 100, 85, 45, 55, 110, 20, 70, 65. (07 Marks)

OR

- 8 a. Draw a binary tree for the following expression $3 + 4 * (7 - 6) / 4 + 3$. Traverse the above generated tree using inorder, preorder and postorder. Also write a function in C for each one. (09 Marks)
- b. What is the advantage of threaded binary tree over binary tree? Explain the construction of threaded binary tree for 10, 20, 30, 40, 50. (07 Marks)

Module-5

- 9 a. Define graph. Write the difference between graph and trees. For the given graph, show the adjacency matrix and adjacency list representation of the graph. [Refer Fig.Q9(a)]

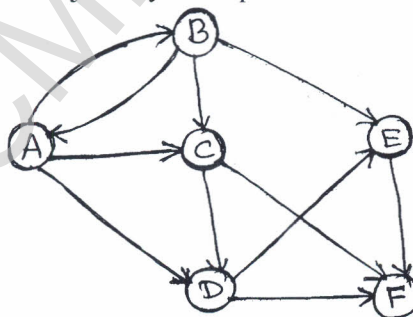


Fig.Q9(a)

- b. What are the methods used for traversing a graph? Explain any one with example. (08 Marks)

OR

- 10 a. Write a C function for insertion sort. Sort the following list using insertion sort: 50, 30, 10, 70, 40, 20, 60. (08 Marks)
- b. What is collision? What are the methods to resolve collision? Explain linear probing with an example. (08 Marks)

* * * * *