

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

Third Semester B.E. Degree Examination, Aug./Sept. 2020
Data Structure and Applications

Time: 3 hrs.

Max. Marks: 100

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

Module-1

- 1 a. Define Data Structures? List and explain the classification of data structures. (05 Marks)
- b. With examples, explain pointer declaration, pointer initialization and void pointer. (05 Marks)
- c. Write the algorithms for inserting an element into a linear Array and deleting from a Linear Array. (10 Marks)

OR

- 2 a. Show the array representation of two polynomials :
 $A(x) = 2x^{1000} + 1$ and $B(x) = x^4 + 10x^3 + 3x^2 + 1$.
 Write a C function to add two polynomial $A(x)$ and $B(x)$ term by term to produce $D(x)$, where $D(x) = A(x) + B(x)$. (10 Marks)
- b. Construct the failure function array for the pattern 'a b c a b y'. Write a C functions for computing the failure functions and KMP pattern matching operation. (10 Marks)

Module-2

- 3 a. Write an algorithm to evaluate postfix expression. Trace the algorithm for the expression $6\ 5\ 1 - 4\ * 2\ 3\ \wedge / +$ and showing stack contents. (08 Marks)
- b. Write a C recursive function for Tower of Hanoi. Trace the function for 3 disks with call tree diagram. (08 Marks)
- c. Write a C function to implement push and pop functions using arrays. (04 Marks)

OR

- 4 a. Write a C functions for inserting and deleting an element from a circular Queue. (04 Marks)
- b. Write a C function to double the circular Queue capacity dynamically. (06 Marks)
- c. Convert the following infix expression to a postfix expression using stack table :
 i) $a * (b + c) / d - e * f \wedge g$
 ii) $A + B * (C - D / E * F) * G$ (10 Marks)

Module-3

- 5 a. Explain the different types of linked list with diagram. (06 Marks)
- b. List advantages, disadvantages and applications of linked list. (04 Marks)
- c. Write the node structure for linked representation of polynomial. Explain the algorithm to add two polynomials represented using linked list. (10 Marks)

OR

- 6 a. Define Linked list. Write a C program to implement insert and delete operation on stack using linked list. (10 Marks)
- b. Represent the following sparse matrix using linked list.

$$\begin{bmatrix} 2 & 0 & 0 & 0 \\ 4 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 \\ 8 & 0 & 0 & 1 \\ 0 & 0 & 6 & 0 \end{bmatrix}$$

(04 Marks)

- c. Write a C functions for following using singly linked list :
- Reverse that list
 - Concatenate two list.

(06 Marks)

Module-4

- 7 a. Define the following with respect to trees with an example.
- Degree of tree
 - Siblings
 - Leaf nodes
 - Level of a tree
 - Height of a tree.
- (05 Marks)
- b. Write a recursive function to search a key value in a Binary Search Tree. Construct a BST for the given set of values.
14, 15, 4, 9, 7, 18, 3, 5, 16, 20, 17, 9 and perform traverse on it. (10 Marks)
- c. Discuss advantages of threaded binary tree over binary tree and explain threaded binary tree construction with suitable example. (05 Marks)

OR

- 8 a. Define Binary tree. How it is represented using array and linked list? (06 Marks)
- b. Create an expression tree for a given expression $a + b * c - d / e * f$ and write C routines to traverse the tree using in-order, preorder and postorder. (08 Marks)
- c. Write a C function to
- Count number of leaf node in binary tree
 - Find a largest element in BST.
- (06 Marks)

Module-5

- 9 a. Define the following with respect to graphs with an example :
- Connected graph
 - Directed graph
 - Multigraph
 - Complete graph
 - Subgraph.
- (10 Marks)
- b. Write a C routines to implement bfs () and dfs () functions. (10 Marks)

OR

- 10 a. Write a insertion sort algorithm. Explain with an example. (08 Marks)
- b. Discuss division, mid-square and folding hash functions. (07 Marks)
- c. Initially following keys 10,16, 11, 1, 3, 4, 23 and 15 are inserted into an empty hash table of length of 10. Using open addressing with hash function $h(k) = k \text{ mod } 10$ and linear probing. What is the resultant hash table? Explain linear probing. (05 Marks)
