2022 SCHEME

Basavarajeswari Group of Institutions

BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT

(Autonomous Institute under Visvesvaraya Technological University, Belagavi)

USN										

Course Code **22CS/AI/CA/CD/34**

Third Semester B.E. Degree Examinations, January 2025

DATA STRUCTURE AND APPLICATIONS

(Common to CSE, AIML, CSE- AI, CSE- DS)

Duration: 3 hrs **Max. Marks: 100**

Note: 1. Answer any* FIVE *full questions choosing ONE full Question from each Module.
2. Missing data, if any, may be suitably assumed

Q. No		*Question*	*Marks*	*(RBTL:CO:PI)*
		Module-1		
1.	**a.**	Define data structures. Explain with a neat block diagram different types of data structures with examples. Discuss the various operations on data structures.	**06**	**(2 : 1 : 1.6.1)**
	b.	What is dynamic memory allocation? Explain the dynamic memory allocation functions supported by 'C' with syntax and examples.	**08**	**(2 : 1 : 1.6.1)**
	c.	Define Sparse Matrix. Give the triplet form of a following matrix and also find its transpose. $A=\begin{bmatrix} 8 & 0 & 6 & 0 & 0 \\ 0 & 0 & 9 & 0 & 5 \\ 3 & 0 & 7 & 0 & 0 \\ 0 & 3 & 0 & 0 & 6 \end{bmatrix}$	**06**	**(3 : 1 : 1.7.1)**
		(OR)		
2.	**a.**	Explain with an example how $A(x)=8x^{13}+3x^4+4x^2+15$ and $B(x)=5x^5+20x^3+2$ are stored in 1-D array.	**06**	**(3: 1 : 1.7.1)**
	b.	Demonstrate self-referential structures with suitable example.	**06**	**(3 : 1 : 1.6.1)**
	c.	Write a program to demonstrate the basic operations of arrays.	**08**	**(2 : 1 : 1.6.1)**
		Module-2		
3.	**a.**	Outline the algorithm for converting an infix expression to a postfix one using the same algorithm, convert the following infix expression to a postfix expression. *((A + B) – C * (D / E)) + F*	**10**	**(3 : 2 : 1.7.1)**
	b.	Write a C program to perform push(), pop() and display() operations on STACK.	**10**	**(3 : 2 : 1.7.1)**
		(OR)		
4.	**a.**	Define recursion. Write the recursive function/procedure for (i) Fibonacci Numbers and (ii) GCD of 2 Numbers.	**08**	**(3 : 2 : 1.7.1)**
	b.	Write a C program to perform insertion, deletion and display operations on queue.	**07**	**(2 : 2 : 1.7.1)**
	c.	List different types of queues. State the limitation of linear queue.	**05**	**(2 : 2 : 1.7.1)**
		Module-3		
5.	**a.**	Develop C functions to implement the following in a doubly linked list (i) Insert a node at the rear end (ii) Delete a node from the rear end (iii) Insert a node after a number specified (iv) Delete a node after a number specified	**10**	(3 : 3 : 2.1.3)

Note: (RBTL - Revised Bloom's Taxonomy Level: CO - Course Outcome: PI– Performance Indicator)

	b.	With a C-function, to create a node, explain the operations add and delete on a singly linked list (SLL) with a proper message where each node contains the details of the Book in the form of BookID, BookName, BookAuthor as data fields.	**10**	(3 : 3 : 2.1.3)

(OR)

6.	**a.**	Write the differences between arrays and linked lists.	**04**	**(2 : 3 : 1.6.1)**
	b.	Write the node representation for the linked list representation of a polynomial. Write an algorithm to add two polynomials represented as linked list.	**08**	**(3 : 3 : 2.1.3)**
	c.	Develop C functions to implement the various operations of queues using linked list.	**08**	**(3 : 3 : 1.7.1)**

Module-4

7.	**a.**	With suitable examples, define the following: (i) Binary Tree (ii) Complete Binary Tree (iii) Degree of a node (iv) Level of a binary tree	**06**	**(2 : 4 : 1.6.1)**
	b.	Construct a Binary search tree for the given data **47,12,75,88,90,73,57,1,85,50,62** Also, write inorder, preorder, postorder traversal for the constructed binary search tree.	**08**	**(3 : 4 : 2.1.3)**
	c.	Write a C recursive routine to traverse the tree (i) Pre-order traversal (ii) Post-order traversal (iii) In-Order traversal	**06**	**(3 : 4 : 1.7.1)**

(OR)

8.	**a.**	Write a program to create a binary search tree and perform a searching operation on BST.	**10**	**(3 : 4 : 2.1.3)**
	b.	Consider the following tree T in Fig. Q8 (b). Write C functions for inorder, preorder, postorder traversals for the tree T. Also find the depth of tree T.	**10**	**(3 : 4 : 2.1.3)**

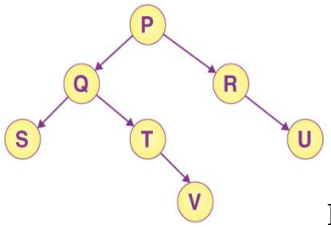

Fig. Q8 (b)

Module-5

9.	**a.**	Define the following with examples (i) Digraph (ii) Weighted graph (iii) Multi graph (iv) Adjacency matrix	**06**	**(2 : 5 : 1.6.1)**
	b.	Define Graphs. Give the Adjacency matrix and Adjacency list representation for the given graph Fig. Q9 (b).	**07**	**(3 : 5 : 2.1.3)**

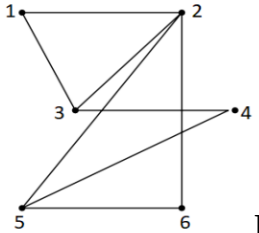

Fig. Q9 (b)

	c.	Define Hashing. Explain 3 Hash functions.	**07**	**(2 : 5 : 1.6.1)**

(OR)

10	**a.**	Write an algorithm for Breadth First Search and Depth First Search. Explain each with suitable example.	**10**	**(3 : 5 : 2.1.3)**
	b.	Explain hashing and collision. What are the methods to resolve collision? Provide example for each.	**10**	**(2 : 5 : 1.7.1)**

** ** **

Note: (RBTL - Revised Bloom's Taxonomy Level: CO - Course Outcome: PI– Performance Indicator)