U.S.N.										

B.M.S. College of Engineering, Bengaluru-560019

Autonomous Institute Affiliated to VTU

July 2023 Semester End Main Examinations

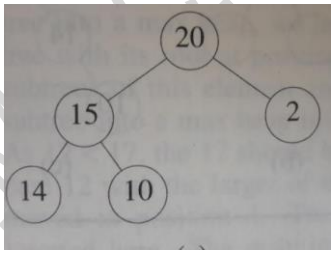
Programme: B.E. **Semester: VI**
Branch: Electronics and Communication **Duration: 3 hrs.**
Course Code: 19EC6PE3DS **Max Marks: 100**
Course: Data Structures and Applications **Date: 17.07.2023**

Instructions: 1. Answer any FIVE full questions, choosing one full question from each unit.
2. Missing data, if any, may be suitably assumed.
3. ***All codes MUST be supported by meaningful comments and sample output.***

Important Note: Completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages. Revealing of identification, appeal to evaluator will be treated as malpractice.

			UNIT - I	CO	PO	Marks
1	a)	Create a database to store approximately 1000k data. Develop a function STORE(data_value, position), where data_value can be of any data type and position specifies where in the data structure the data is to stored. Similarly, a function, DElete(data_value, position) that will retrieve the data at the specified position in the variable data_value and free the memory location is to be written. Outline the working of the developed functions with a main program.		*CO1*	*PO1*	**10**
	b)	Develop a class to store data using Linked Representation. Include member functions to insert data 'x' at given position 'k'.		*CO 1*	*PO 1*	**10**
		UNIT - II				
2	a)	Analyse how memory can be optimally used to store data of a diagonal matrix. List the uses of a diagonal matrix.(no codes)		*CO 2*	*PO 2*	**05**
	b)	Illustrate the scheme of deriving a mapping function for a lower triangular matrix. (no codes)		*CO 1*	*PO 1*	**05**
	c)	Develop a function to add two sparse matrices. Illustrate the working of your code with a sample.		*CO 1*	*PO 1*	**10**
		OR				
3	a)	Illustrate with a mapping function the storage mechanism of a tridiagonal matrix.(no codes)		*CO 1*	*PO 1*	**8**
	b)	Analyse how data can be stored in a 1D array to facilitate performing of arithmetic / logical operations. Create a class 'array1D' and include appropriate data and functions for the following code to compile:		*CO 2*	*PO 2*	**12**

			array1D<int> obj1(10), obj2(20), obj3(5); // create 3 objects with sizes as indicated
			obj1 = obj2+obj3; // add the corresponding data of obj2 with that of obj3 and store the result in corresponding positions of obj1
			obj2--; // decrement the value of data contained in obj2 by 1
			}			

			UNIT - III			
	4	a)	Realise a data structure using linked representation that lets addition and deletion from one end only.	*CO 2*	*PO 2*	**10**
		b)	Explain any one application of queue. Outline the logic with necessary **pseudo code.**	*CO 2*	*PO 2*	**10**
			OR			
	5	a)	Develop a **function** parenthesis_match(*expr) using an appropriate data structure to check if the left and right parenthesis in the given '*expr' match. Provide sample output.	*CO 2*	*PO 2*	**10**
		b)	Analyse how a circular queue is better than a linear queue. Show **pictorially** the movement of front and rear pointers with each add and delete operation. Write the **equations** to check queue full and empty conditions. (NO codes)	*CO 2*	*PO 2*	**10**
			UNIT - IV			
	6	a)	Write **functions** to traverse a tree in any two different ways. Outline with an example.	*CO 1*	*PO 1*	**10**
		b)	Construct an expression tree for the expression ((a+b)>(c-e)) || (a<f && (x<y || y>z)). Mention the number of levels of the tree and the leaf nodes. Write prefix form of the expression. (No codes)	*CO 1*	*PO 1*	**10**
			UNIT V			
	7	a)	 Discuss max heap. For the heap represented in the adjacent figure, outline the process of inserting 5 and 21 to it. (No codes)	*CO 1*	*PO 1*	**10**
		b)	Discuss AVL trees with an example. Explain the different rotations that can be performed to balance the tree with an example. (No codes)	*CO 1*	*PO 1*	**10**

****