

# CBCS SCHEME - Make-Up Exam

USN

|  |  |  |  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|--|--|--|
|  |  |  |  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|--|--|--|

BCS304

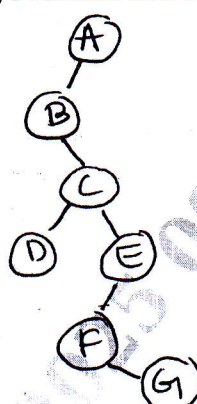
## Third Semester B.E./B.Tech. Degree Examination, June/July 2025 Data Structures and Applications

Time: 3 hrs.

Max. Marks: 100

*Note: 1. Answer any FIVE full questions, choosing ONE full question from each module.  
2. M : Marks , L: Bloom's level , C: Course outcomes.*

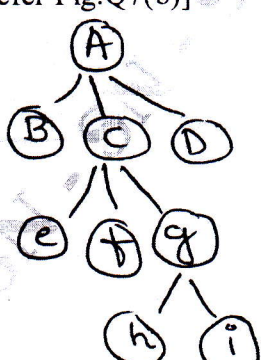
| Module – 1 |    |                                                                                                                                                                                       | M  | L  | C   |
|------------|----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|----|-----|
| Q.1        | a. | Differentiate between static and dynamic memory allocation. Write syntax for malloc( ), calloc( ), realloc( ) and free( ) functions.                                                  | 08 | L2 | CO1 |
|            | b. | Write KMP pattern matching algorithm and apply the same to search the pattern "abcdabcy" in the text "abcxabcdbxabcdbcdabcy".                                                         | 08 | L3 | CO2 |
|            | c. | Differentiate between Structure and Union.                                                                                                                                            | 04 | L1 | CO1 |
| OR         |    |                                                                                                                                                                                       |    |    |     |
| Q.2        | a. | Write a C function to evaluate postfix expression. Evaluate 231 * + 9 – using the same.                                                                                               | 08 | L2 | CO2 |
|            | b. | Implement push( ), pop( ) and display operations of stack using array.                                                                                                                | 08 | L2 | CO2 |
|            | c. | Write a C function to perform transpose of a given sparse matrix.                                                                                                                     | 04 | L3 | CO1 |
| Module – 2 |    |                                                                                                                                                                                       |    |    |     |
| Q.3        | a. | Explain the need of Circular Queue. Implement Insert( ), delete( ) and display( ) operations on the circular queue.                                                                   | 10 | L3 | CO2 |
|            | b. | Implement following operations on Single Linked List:<br>(i) Delete a node at the end of SLL.<br>(ii) Insert a node at the beginning of SLL.                                          | 10 | L3 | CO3 |
| OR         |    |                                                                                                                                                                                       |    |    |     |
| Q.4        | a. | Explain and write a C function to implement Multiple Stack.                                                                                                                           | 10 | L3 | CO2 |
|            | b. | Develop a C function to add two polynomials using Single Linked List.                                                                                                                 | 10 | L3 | CO3 |
| Module – 3 |    |                                                                                                                                                                                       |    |    |     |
| Q.5        | a. | Give the diagrammatical representation of below sparse matrix using Linked List.<br>$\begin{bmatrix} 0 & 0 & 7 & 0 \\ 0 & 0 & 0 & 10 \\ 3 & 1 & 0 & 4 \\ 0 & 0 & 0 & 0 \end{bmatrix}$ | 05 | L2 | CO3 |

|     |    |                                                                                                                                                                                                                                                               |    |    |     |
|-----|----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|----|-----|
| Q.5 | b. | <p>Traverse given tree using Inorder, Preorder and Post order traversal. Also write C functions for Inorder, Preorder and Postorder. [Refer Fig.Q5(b)]</p>  <p>Fig.Q5(b)</p> | 08 | L3 | CO4 |
|     | c. | <p>What are the advantages of threaded binary tree over a binary tree? Construct threaded binary tree for the following elements:<br/>A,B,C,D,E,F,G,H,I</p>                                                                                                   | 07 | L3 | CO4 |

## OR

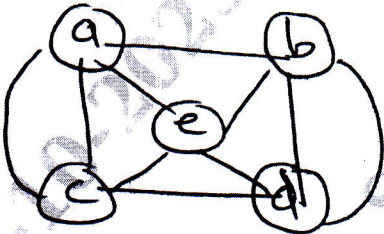
|     |    |                                                                                                                                                              |    |    |     |
|-----|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------|----|----|-----|
| Q.6 | a. | <p>Define Double Linked List. Create a C function for the following DLL operations:</p> <p>(i) Inserting a node at front<br/>(ii) Deleting a node at end</p> | 08 | L3 | CO3 |
|     | b. | <p>Define Tree. Explain the representation of Tree with an example.</p>                                                                                      | 06 | L2 | CO4 |
|     | c. | <p>Define the following with example :</p> <p>i) Complete Binary Tree      (ii) Height of a Tree      (iii) Full Binary Tree</p>                             | 06 | L2 | CO4 |

## Module - 4

|     |    |                                                                                                                                                                                                                                     |    |    |     |
|-----|----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|----|-----|
| Q.7 | a. | <p>Write a iterative C function to perform following operations on Binary Search Tree :</p> <p>i) Search a key element      ii) Insert a node</p>                                                                                   | 10 | L3 | CO4 |
|     | b. | <p>Convert the following forest into Binary Tree and perform inorder, preorder and post order traversal. [Refer Fig.Q7(b)]</p>  <p>Fig.Q7(b)</p> | 10 | L2 | CO4 |



OR

|     |    |                                                                                                                                                                                                                                                                                                                                                                                          |    |    |     |    |    |    |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |
|-----|----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|----|-----|----|----|----|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|
| Q.8 | a. | Define Selection Tree. Construct first 4 winners from given runs below:<br><table><tr><td>5</td><td>9</td><td>18</td><td>8</td><td>90</td><td>09</td><td>7</td><td>8</td></tr><tr><td>14</td><td>13</td><td>19</td><td>11</td><td>95</td><td>17</td><td>13</td><td>10</td></tr><tr><td>24</td><td>16</td><td>20</td><td>15</td><td>98</td><td>19</td><td>14</td><td>15</td></tr></table> | 5  | 9  | 18  | 8  | 90 | 09 | 7 | 8 | 14 | 13 | 19 | 11 | 95 | 17 | 13 | 10 | 24 | 16 | 20 | 15 | 98 | 19 | 14 | 15 | 10 | L2 | CO4 |
| 5   | 9  | 18                                                                                                                                                                                                                                                                                                                                                                                       | 8  | 90 | 09  | 7  | 8  |    |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |
| 14  | 13 | 19                                                                                                                                                                                                                                                                                                                                                                                       | 11 | 95 | 17  | 13 | 10 |    |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |
| 24  | 16 | 20                                                                                                                                                                                                                                                                                                                                                                                       | 15 | 98 | 19  | 14 | 15 |    |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |
|     | b. | Design the algorithm to traverse below graph using BFS and DFS technique. Traverse the below graph using BFS from source vertex. [Refer Fig.Q8(b)]<br><br>Fig.Q8(b)                                                                                                                                    | 10 | L3 | CO4 |    |    |    |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |

## Module – 5

|     |    |                                                                                                                                                                                                                                |    |    |     |
|-----|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|----|-----|
| Q.9 | a. | Write a note on Static and Dynamic Hashing.                                                                                                                                                                                    | 06 | L2 | CO5 |
|     | b. | Construct the OBST for following items. Let $n = 4$<br>$(a_1, a_2, a_3, a_4) = (\text{do}, \text{if}, \text{int}, \text{while})$<br>Let $(p_1, p_2, p_3, p_4) = (3, 3, 1, 1)$<br>$(q_0, q_1, q_2, q_3, q_4) = (2, 3, 1, 1, 1)$ | 10 | L3 | CO5 |
|     | c. | Explain the following with an example :<br>(i) HBLT                      (ii) WBLT                                                                                                                                             | 04 | L3 | CO5 |

OR

|      |    |                                                                                                                                   |    |    |     |
|------|----|-----------------------------------------------------------------------------------------------------------------------------------|----|----|-----|
| Q.10 | a. | Define Collision. Explain the methods to resolve collision. Explain linear probing for the data<br>12, 13, 15, 16, 74, 75, 83, 42 | 08 | L2 | CO5 |
|      | b. | Explain Leftist tree with an example.                                                                                             | 08 | L2 | CO5 |
|      | c. | Explain different Hash functions.                                                                                                 | 04 | L2 | CO5 |

\*\*\*\*\*