

Lab	Type	Practical
<b>Unit: I – Introduction</b>		
1	A	1. Write a program to calculate area of a Circle.
	A	2. Write a program to find whether a number is odd or even
	A	3. Write a program to determine whether the entered character is vowel or not.
	A	4. Write a program to find factorial of a number. (Using Loop)
	A	5. Write a program to find factorial of a number. (Using Recursion)
	B	6. Write a program to find power of a number using loop.
	B	7. Write a program to find factors of a given number.
	B	8. Write a program to check whether a number is prime or not.
<b>Unit: II – Linear Data Structures: Array, Stack, Queue and Linked List</b>		
2	A	9. Write a program to read and display n numbers using an array.
	A	10. Write a program to calculate sum of numbers from m to n.
	A	11. Write a program to calculate average of first n numbers.
	A	12. Write a program to find position of the smallest number from given n numbers.
	B	13. Write a program to find whether the array contains a duplicate number or not.
	B	14. Read n numbers in an array then read two different numbers, replace 1 <sup>st</sup> number with 2 <sup>nd</sup> number in an array and print its index and final array.
3	A	15. Write a program to insert a number at a given location in an array.
	A	16. Write a program to delete a number from a given location in an array.
	B	17. Write a program to insert a number in an array that is already sorted in an ascending order.

	<b>B</b>	<b>18.</b> Write a program to delete a number from an array that is already sorted in an ascending order.
	<b>B</b>	<b>19.</b> Write a program to merge two unsorted arrays.
	<b>A</b>	<b>20.</b> Read two 2x2 matrices and perform addition of matrices into third matrix and print it
	<b>B</b>	<b>21.</b> Read two matrices, first 3x2 and second 2x3, perform multiplication operation and store result in third matrix and print it.
<b>4</b>	<b>A</b>	<b>22.</b> Write a program to swap two numbers using user-defines method.
	<b>A</b>	<b>23.</b> Create class Employee_Detail with attributes Employee_ID, Name, Designation, and Salary. Write a program to read the detail from user and print it.
	<b>B</b>	<b>24.</b> Create array of object of class Student_Detail with attributes Enrollment_No, Name, Semester, CPI for 5 students, scan their information and print it.
<b>4</b>	<b>A</b>	<b>25.</b> Write algorithms to perform following operations on a stack: <ul style="list-style-type: none"> <li>• Push</li> <li>• Pop</li> <li>• Peep</li> <li>• Change</li> </ul>
		<b>26.</b> Take a stack of size 3 and performing following operations. Show the position of stack at each step: <ul style="list-style-type: none"> <li>• Push 1, Push 2, Push 3, Push 4</li> <li>• Pop, Pop</li> <li>• Push 5</li> <li>• Change 3rd element to 8</li> <li>• Push 6 &amp; 7</li> <li>• Traverse the stack</li> </ul>
	<b>B</b>	<b>27.</b> Write a menu driven program to implement following operations on the Stack created using an Array <ul style="list-style-type: none"> <li>• PUSH</li> <li>• POP</li> <li>• DISPLAY</li> <li>• PEEP</li> <li>• CHANGE</li> </ul>
<b>5</b>	<b>A</b>	<b>28.</b> How stack can be used to recognize strings aca, bcb, abcba, abbcbbba?
	<b>B</b>	<b>29.</b> Implement a program described in 5(A)

	<b>B</b>	30. Write a program to determine if an input character string is of the form $a^i b^j$ where $i \geq 1$ i.e., Number of 'a' should be equal to number of 'b'.
<b>6</b>	<b>A</b>	<p><b>31.</b> Convert the following infix expressions into postfix expressions:</p> <ul style="list-style-type: none"> <li><math>(A + B * C / D - E + F / G / (H + I))</math></li> <li><math>(A + B) * C + D / (B + A * C) + D</math></li> </ul> <p><b>32.</b> Convert the following infix expressions into prefix expressions:</p> <ul style="list-style-type: none"> <li><math>A - B / (C * D ^ E)</math></li> <li><math>(a + b ^ c ^ d) * (e + f / d)</math></li> </ul>
	<b>B</b>	33. Implement a program to convert in-fix notation to post-fix notation using stack.
<b>7</b>	<b>A</b>	<p>34. Evaluate the following expressions showing every status of stack in tabular form:</p> <ul style="list-style-type: none"> <li>5, 4, 6, +, *, 4, 9, 3, /, +, *</li> <li>7, 5, 2, +, *, 4, 1, 1, +, /, -</li> </ul> <p>35. Evaluate the following expressions showing every status of stack in tabular form:</p> <ul style="list-style-type: none"> <li>*, +, 6, 9, -, 3, 1</li> <li>+, -, *, 2, 2, 1, 16, 8, 5</li> </ul>
	<b>B</b>	36. Write a program for evaluation of post-fix Expression using Stack.
	<b>B</b>	37. Write a program for evaluation of pre-fix Expression using Stack.
<b>8</b>	<b>A</b>	<p>38. Write algorithms to perform following operations on a simple queue:</p> <ul style="list-style-type: none"> <li>Insert</li> <li>Delete</li> </ul>
		<p>39. Perform following operations on queue with size 4 &amp; draw queue after each operation:</p> <ul style="list-style-type: none"> <li>Insert 'A', Insert 'B', Insert 'C'</li> <li>Delete, Delete</li> <li>Insert 'D', Insert 'E'</li> </ul>
	<b>A</b>	<p>40. Write algorithms to perform following operations on a circular queue:</p> <ul style="list-style-type: none"> <li>Insert</li> <li>Delete</li> </ul>

		<p>41. Consider the following circular queue having 6 memory cells. Front=2, Rear=4 Queue: _, A, C, D, _, _. Describe queue as following operation take place:</p> <ul style="list-style-type: none"> <li>• F is added to the queue</li> <li>• Two letters are deleted</li> <li>• R is added to the queue</li> <li>• S is added to the queue</li> <li>• One letter is deleted</li> </ul>
	<b>B</b>	<p>42. Write a menu driven program to implement following operations on the Queue created using an Array</p> <ul style="list-style-type: none"> <li>• ENQUEUE</li> <li>• DEQUEUE</li> <li>• DISPLAY</li> </ul>
	<b>B</b>	<p>43. Write a menu driven program to implement following operations on the Queue created using an Array</p> <ul style="list-style-type: none"> <li>• ENQUEUE</li> <li>• DEQUEUE</li> <li>• DISPLAY</li> </ul>
<b>9</b>	<b>A</b>	<p>44. Draw a Node Structure of Single linked list for following example (include insertion and deletion)</p> <ul style="list-style-type: none"> <li>• Insert 10</li> <li>• Insert 20 at the end of the list</li> <li>• Insert 5 at the beginning of the list</li> <li>• Delete the last node</li> <li>• Delete the first node</li> </ul>
	<b>A</b>	45. Write a sample java code to implement a node structure
	<b>A</b>	46. Implement a program to create a node for singly linked list. Read the data in a node, print the node.
	<b>B</b>	<p>47. Write a menu driven program to implement following operations on the singly linked list.</p> <ul style="list-style-type: none"> <li>• Insert a node at the front of the linked list.</li> <li>• Display all nodes.</li> <li>• Delete a first node of the linked list.</li> <li>• Insert a node at the end of the linked list.</li> <li>• Delete a last node of the linked list.</li> <li>• Delete a node from specified position.</li> </ul>
<b>10</b>	<b>A</b>	48. Draw a structure of Stack implemented using LinkedList (Example)
	<b>A</b>	49. Draw a structure of Queue implemented using Linked List (Example)

	<b>B</b>	50. Write a program to implement stack using singly linked list.
	<b>B</b>	51. Write a program to implement queue using singly linked list.
<b>11</b>	<b>A</b>	52. Explain the concept of circular linked list with example <ul style="list-style-type: none"> <li>• Insert 10</li> <li>• Insert 20 at the end of the list</li> <li>• Insert 5 at the beginning of the list</li> <li>• Delete the last node</li> <li>• Delete the first node</li> </ul>
	<b>B</b>	53. Write a menu driven program to implement following operations on the circular linked list. <ul style="list-style-type: none"> <li>• Insert a node at the front of the linked list.</li> <li>• Delete a node from specified position.</li> <li>• Insert a node at the end of the linked list.</li> <li>• Display all nodes.</li> </ul>
<b>12</b>	<b>A</b>	54. Explain the concept of doubly linked list with example <ul style="list-style-type: none"> <li>• Insert 10</li> <li>• Insert 20 at the end of the list</li> <li>• Insert 5 at the beginning of the list</li> <li>• Delete the last node</li> <li>• Delete the first node</li> </ul>
	<b>B</b>	55. Write a menu driven program to implement following operations on the doubly linked list. <ul style="list-style-type: none"> <li>• Insert a node at the front of the linked list.</li> <li>• Delete a node from specified position.</li> <li>• Insert a node at the end of the linked list. (Home Work)</li> <li>• Display all nodes. (Home Work)</li> </ul>
<b>Unit: III – Nonlinear Data Structures: Tree and Graph</b>		
<b>13</b>	<b>A</b>	56. Construct a Binary Tree for Following Elements 10 20 30 40 50 45 60 15 70
	<b>A</b>	57. Provide a Pre-order, Post-order and in-order traversal for following binary search tree. <ul style="list-style-type: none"> <li>• 10 15 20 5 4 3 18 19</li> <li>• 20 30 10 15 40 9 8 7</li> <li>• 50, 70, 60, 20, 90, 10, 40, 100</li> </ul>

	<b>A</b>	58. Construct a Binary tree from pre-order and post order traversal <ul style="list-style-type: none"> <li>1, 2, 4, 8, 9, 5, 3, 6, 7 (Pre)</li> <li>8, 9, 4, 5, 2, 6, 7, 3, 1 (Post)</li> </ul>
	<b>B</b>	59. Implement a menu based Binary search tree for following operation <ul style="list-style-type: none"> <li>Insert a node</li> <li>Delete a node</li> <li>Preorder Traversal</li> <li>Post order Traversal</li> <li>In order Traversal</li> </ul>
<b>Unit: IV – Hashing and File Structures</b>		
<b>14</b>	<b>A</b>	60. Implement an AVL tree for following node value. 13 10 15 12 18 2 1 4 After that delete a node 15 and redraw a AVL tree
	<b>A</b>	61. Explain BFS and DFS search techniques in Graph
	<b>B</b>	62. Implement a Dictionary (key, value) pair using Hash-table.
<b>Unit: V – Sorting and Searching</b>		
<b>15</b>	<b>A</b>	63. Discuss a Hash table advantages and disadvantage.
	<b>A</b>	64. Insert a Following value into Hash table where table size is 10, use a linear probing method 13 10 15 12 18 2 1 4
	<b>B</b>	65. Implement a Hash table using Array
<b>16</b>	<b>A</b>	66. Implement a linear search algorithm
	<b>A</b>	67. Implement a binary search algorithm
<b>17</b>	<b>A</b>	68. Implement a given sorting technique for Array A= {10,20,5,8,9,11,30}; sort in ascending order <ul style="list-style-type: none"> <li>Bubble Sort.</li> <li>Insertion Sort</li> </ul>
	<b>B</b>	69. Implement a given sorting technique for Array A= {10,20,5,8,9,11,30}; sort in ascending order <ul style="list-style-type: none"> <li>Bubble Sort.</li> <li>Insertion Sort</li> </ul>

<b>18</b>	<b>A</b>	70. Implement a given sorting technique for Array A= {10,20,5,8,9,11,30}; sort in ascending order <ul style="list-style-type: none"> <li>• Selection Sort</li> <li>• Bucket Sort</li> </ul>
	<b>B</b>	71. Implement a given sorting technique for Array A= {10,20,5,8,9,11,30}; sort in ascending order <ul style="list-style-type: none"> <li>• Selection Sort</li> <li>• Bucket Sort</li> </ul>
<b>19</b>	<b>A</b>	72. Implement a given sorting technique for Array A= {10,20,5,8,9,11,30}; sort in ascending order <ul style="list-style-type: none"> <li>• Redix Sort</li> <li>• Shell Sort</li> </ul>
	<b>B</b>	73. Implement a given sorting technique for Array A= {10,20,5,8,9,11,30}; sort in ascending order <ul style="list-style-type: none"> <li>• Redix Sort</li> <li>• Shell Sort</li> </ul>
<b>20</b>	<b>A</b>	74. Implement a given sorting technique for Array A= {10,20,5,8,9,11,30}; sort in ascending order <ul style="list-style-type: none"> <li>• Counting Sort</li> <li>• Tree Sort</li> </ul>
	<b>B</b>	75. Implement a given sorting technique for Array A= {10,20,5,8,9,11,30}; sort in ascending order <ul style="list-style-type: none"> <li>• Counting Sort</li> <li>• Tree Sort</li> </ul>
<b>21</b>	<b>A</b>	76. Implement a given sorting technique for Array A= {10,20,5,8,9,11,30}; sort in ascending order <ul style="list-style-type: none"> <li>• Merge Sort</li> <li>• Quick Sort</li> <li>• Heap Sort</li> </ul>
	<b>B</b>	77. Implement a given sorting technique for Array A= {10,20,5,8,9,11,30}; sort in ascending order <ul style="list-style-type: none"> <li>• Merge Sort</li> <li>• Quick Sort</li> <li>• Heap Sort</li> </ul>