

# Maharaja Surajmal Institute

## Department of Computer Application

### LIST OF PRACTICALS FOR (Data Structure Practical)

Programme: BCA	Semester: II	Paper Code: BCA 174	Academic Year:2023-2024
----------------	--------------	---------------------	-------------------------

S No	Unit No & Name	Concept/ Content/Area/Functionality/ on which practical is based	Problem Statement & Teaching Method
1.	Unit I	Arrays	1. WAP to implement following operation on one dimensional array (i) Insertion in sorted and unsorted arrays (ii) Deletion from sorted and unsorted arrays (iii) Traversal (iv) Reverse
2.	Unit I	Arrays	2. WAP to perform following string related function (user defined) on an array <ul style="list-style-type: none"> <li>i. Finding string length</li> <li>ii. Concatenation of two strings</li> <li>iii. Comparing two string</li> <li>iv. Copy one string to another</li> </ul>
3.	Unit I	Arrays	3. WAP to perform sorting in array(Menu Driven) :- <ul style="list-style-type: none"> <li>i. Bubble sort</li> <li>ii. Selection sort</li> <li>iii. Insertion sort</li> <li>iv. merge sort</li> <li>v. Merging two sorted array</li> </ul>
4.	Unit I	Arrays	4. WAP to perform searching in array(Menu Driven) :- <ul style="list-style-type: none"> <li>i. Linear search</li> <li>ii. Binary Search</li> </ul>
5.	Unit I	Sparse Matrix	5. WAP to accept a matrix from user and find out whether matrix is sparse or not and convert into triplex matrix or tuple form
6.	Unit I	Sparse Matrix	6. WAP to display the sparse matrix in the following way(Menu Driven) :- <ul style="list-style-type: none"> <li>i. Upper triangular matrix</li> <li>ii. Lower triangular matrix</li> <li>iii. Diagonal matrix</li> </ul>
7.	Unit I	Sparse Matrix	7. WAP to perform the following operations on sparse matrix(Menu Driven:- <ul style="list-style-type: none"> <li>i. Transpose of sparse matrix</li> <li>ii. Add two sparse matrices</li> <li>iii. Multiply two sparse matrices</li> </ul>
8.	Unit II	Linked List	8. WAP to do the following operations on Singly linked list- <ul style="list-style-type: none"> <li>i. Create a list</li> <li>ii. Traverse a list(forward/backward)</li> <li>iii. Reversal of a list</li> </ul>
9.	Unit II	Linked List	9. WAP to do the following operations on Singly linked list <ul style="list-style-type: none"> <li>i. Insertion in a list(sorted/unordered list) :-               <ul style="list-style-type: none"> <li>a. At the beginning</li> <li>b. At the end</li> <li>c. Anywhere in the middle</li> </ul> </li> </ul>
10.	Unit II	Linked List	10. WAP to do the following operations on Singly linked list:- <ul style="list-style-type: none"> <li>i. Deletion in a list(sorted/ unordered list)               <ul style="list-style-type: none"> <li>a. At the beginning</li> <li>b. At the end</li> <li>c. Anywhere in the middle</li> </ul> </li> </ul>
11.	Unit II	Linked List	11. WAP to do the following operations on linked list:- <ul style="list-style-type: none"> <li>i. Searching from a list</li> <li>ii. Sorting from a list</li> </ul>
12.	Unit II	Linked List	12. WAP to do the following operations on Doubly linked list- <ul style="list-style-type: none"> <li>i. Create a list</li> <li>ii. Traverse a list(forward/backward)</li> </ul>

13.	Unit II	Linked List	<p>13. WAP to do the following operations on Doubly linked list</p> <ol style="list-style-type: none"> <li>Insertion in a list(sorted/unsorted list) <ol style="list-style-type: none"> <li>At the beginning</li> <li>At the end</li> <li>Anywhere in the middle</li> </ol> </li> </ol> <p>14. WAP to do the following operations on Doubly linked list:-</p> <ol style="list-style-type: none"> <li>Deletion in a list(sorted/unsorted list) <ol style="list-style-type: none"> <li>At the beginning</li> <li>At the end</li> <li>Anywhere in the middle</li> </ol> </li> </ol>
14.	Unit II	Linked List	<p>15. WAP to do the following operations on Circular linked list-</p> <ol style="list-style-type: none"> <li>Create a list</li> <li>Traverse a list(forward/backward)</li> </ol>
15.	Unit II	Linked List	<p>16. WAP to do the following operations on Circular linked list</p> <ol style="list-style-type: none"> <li>Insertion in a list(sorted/unsorted list) <ol style="list-style-type: none"> <li>At the beginning</li> <li>At the end</li> <li>Anywhere in the middle</li> </ol> </li> </ol> <p>17. WAP to do the following operations on Circular linked list:-</p> <ol style="list-style-type: none"> <li>Deletion in a list(sorted/unsorted list) <ol style="list-style-type: none"> <li>At the beginning</li> <li>At the end</li> <li>Anywhere in the middle</li> </ol> </li> </ol>
16.	Unit II	Linked List	<p>18. WAP to merge two singly sorted linked lists.</p> <p>19. WAP to implement Polynomial addition operation using Singly linked list.</p>
17.	Unit II	Linked List	<p>20. Write a C program to create two linked lists from a given list in following way :-  INPUT List: - 1 2 3 4 5 6 7 8 9 10  OUTPUT:- First List:- 1 3 5 7 9 Second List:- 2 4 6 8 10</p>
18.	Unit II	Linked List	<p>21. WAP to implement Student Database using Linked List with the following structure :-</p> <ol style="list-style-type: none"> <li>Name</li> <li>Rollno</li> <li>Marks of 5 subjects</li> <li>Average</li> <li>Result, If the average &lt; 50, then print 'Fail', otherwise 'Pass'</li> </ol>
19.	Unit II and Unit III	Stacks	<p>22. WAP to implement push and pop operations using arrays ( Static stack)</p> <p>23. WAP to implement push and pop operations using linked list.(Dynamic Stack)</p> <p>24. Write a program to evaluate (i) Prefix Expression (ii) Postfix Expression using stack.</p>
20.	Unit III	Stacks and Queues	<p>25. Write a program to convert Infix to equivalent (i) Prefix expression (ii) Postfix expression</p> <p>26. WAP to implement queue using array</p> <ol style="list-style-type: none"> <li>Insertion</li> <li>Deletion</li> </ol>
21.	Unit III	Queues	<p>27. WAP to implement queue using linked list</p> <ol style="list-style-type: none"> <li>Insertion</li> <li>Deletion</li> </ol> <p>28. WAP to do insertion and deletion in double ended queue using array( Static De-Queue) and linked list(Dynamic De-Queue)</p>
22.	Unit III	Queues	<p>29. WAP to do insertion and deletion in a Priority Queue using array and linked list</p> <p>30. WAP to implement a (i) Static Circular Queue (ii) Dynamic Circular Queue</p>
23.	Unit III	Stacks	<p>31. Let us assume a Patient's coupon generator for the Doctors' clinic. The patients are given the coupons on first-come-first-serve basis. After the visit of a patient, patient-ID is kept stack-wise. At the end of the day, the count is generated from the stack. Construct a menu-based program for patients' coupons generator using an appropriate data structure.</p>
24.	Unit III	Stacks	<p>32. Sometimes a program requires two stacks containing the same type of items. Suppose two stacks are stored in separate arrays, then one stack might overflow while there is considerable unused space in the other. A neat way to avoid this problem is to put all spaces in one stack and let this stack grow from one end of the array, and the other stack starts from the other end and grows in the opposite direction, i.e., toward the first stack. In this way, if one stack turns out to be large and the other small, then they will still both fit, and there will be no overflow until all space is used. Declare a new structure that includes these two stacks and perform various stack operations.</p>

25.	Unit IV	Trees	33. Implement recursive algorithms for the following operations on Binary Search Tree :- i. Insertion ii. Searching
26.	Unit IV	Trees	34. Implement recursive algorithms for BST traversal- Inorder, Preorder, Postorder. 35. WAP to implement an expression tree. (For example: $(a + b / (c * d) - e)$ )