Lab	Type	Practical
Unit: I – Introduction		
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.
Unit: II – Linear Data Structures: Array, Stack, Queue and Linked List		
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 1st number with 2nd 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	18. Write a program to delete a number from an array that is already sorted in an ascending order.
	B	19. Write a program to merge two unsorted arrays.
	A	20. Read two 2x2 matrices and perform addition of matrices into third matrix and print it
	B	21. Read two matrices, first 3x2 and second 2x3, perform multiplication operation and store result in third matrix and print it.
4	**A**	22. Write a program to swap two numbers using user-defines method.
	A	23. 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	24. Create array of object of class Student_Detail with attributes Enrollment_No, Name, Semester, CPI for 5 students, scan their information and print it.
4	**A**	25. Write algorithms to perform following operations on a stack: • Push • Pop • Peep • Change
		26. Take a stack of size 3 and performing following operations. Show the position of stack at each step: • Push 1, Push 2, Push 3, Push 4 • Pop, Pop • Push 5 • Change 3rd element to 8 • Push 6 & 7 • Traverse the stack
	B	27. Write a menu driven program to implement following operations on the Stack created using an Array • PUSH • POP • DISPLAY • PEEP • CHANGE
5	**A**	28. How stack can be used to recognize strings aca, bcb, abcba, abbcbba?
	B	29. Implement a program described in 5(A)

	B	30. Write a program to determine if an input character string is of the form a^ib^i where i >= 1 i.e., Number of 'a' should be equal to number of 'b'.
6	**A**	31. Convert the following infix expressions into postfix expressions: • (A + B * C / D - E + F / G / (H + I)) • (A + B) * C + D / (B + A * C) + D 32. Convert the following infix expressions into prefix expressions: • A-B/(C*D^E) • (a + b ^ c ^ d) * (e + f / d)
	B	33. Implement a program to convert in-fix notation to post-fix notation using stack.
7	**A**	34. Evaluate the following expressions showing every status of stack in tabular form: • 5, 4, 6, +, *, 4, 9, 3, /, +, * • 7, 5, 2, +, *, 4, 1, 1, +, /, - 35. Evaluate the following expressions showing every status of stack in tabular form: • *, +, 6, 9, -, 3, 1 • +, -, *, 2, 2, 1, 16, 8, 5
	B	36. Write a program for evaluation of post-fix Expression using Stack.
	B	37. Write a program for evaluation of pre-fix Expression using Stack.
8	**A**	38. Write algorithms to perform following operations on a simple queue: • Insert • Delete
		39. Perform following operations on queue with size 4 & draw queue after each operation: • Insert 'A' , Insert 'B', Insert 'C' • Delete, Delete • Insert 'D', Insert 'E'
	A	40. Write algorithms to perform following operations on a circular queue: • Insert • Delete

		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: • F is added to the queue • Two letters are deleted • R is added to the queue • S is added to the queue • One letter is deleted
	B	42. Write a menu driven program to implement following operations on the Queue created using an Array • ENQUEUE • DEQUEUE • DISPLAY
	B	43. Write a menu driven program to implement following operations on the Queue created using an Array • ENQUEUE • DEQUEUE • DISPLAY
9	**A**	44. Draw a Node Structure of Single linked list for following example (include insertion and deletion) • Insert 10 • Insert 20 at the end of the list • Insert 5 at the beginning of the list • Delete the last node • Delete the first node
	A	45. Write a sample java code to implement a node structure
	A	46. Implement a program to create a node for singly linked list. Read the data in a node, print the node.
	B	47. Write a menu driven program to implement following operations on the singly linked list. • Insert a node at the front of the linked list. • Display all nodes. • Delete a first node of the linked list. • Insert a node at the end of the linked list. • Delete a last node of the linked list. • Delete a node from specified position.
10	**A**	48. Draw a structure of Stack implemented using LinkedList (Example)
	A	49. Draw a structure of Queue implemented using Linked List (Example)

	B	50. Write a program to implement stack using singly linked list.
	B	51. Write a program to implement queue using singly linked list.
11	**A**	52. Explain the concept of circular linked list with example • Insert 10 • Insert 20 at the end of the list • Insert 5 at the beginning of the list • Delete the last node • Delete the first node
	B	53. Write a menu driven program to implement following operations on the circular linked list. • Insert a node at the front of the linked list. • Delete a node from specified position. • Insert a node at the end of the linked list. • Display all nodes.
12	**A**	54. Explain the concept of doubly linked list with example • Insert 10 • Insert 20 at the end of the list • Insert 5 at the beginning of the list • Delete the last node • Delete the first node
	B	55. Write a menu driven program to implement following operations on the doubly linked list. • Insert a node at the front of the linked list. • Delete a node from specified position. • Insert a node at the end of the linked list. (Home Work) • Display all nodes. (Home Work)
Unit: III – Nonlinear Data Structures: Tree and Graph		
13	**A**	56. Construct a Binary Tree for Following Elements 10 20 30 40 50 45 60 15 70
	A	57. Provide a Pre-order, Post-order and in-order traversal for following binary search tree. • 10 15 20 5 4 3 18 19 • 20 30 10 15 40 9 8 7 • 50, 70, 60, 20, 90, 10, 40, 100

	A	58. Construct a Binary tree from pre-order and post order traversal • 1, 2, 4, 8, 9, 5, 3, 6, 7 (Pre) • 8, 9, 4, 5, 2, 6, 7, 3, 1 (Post)
	B	59. Implement a menu based Binary search tree for following operation • Insert a node • Delete a node • Preorder Traversal • Post order Traversal • In order Traversal
Unit: IV – Hashing and File Structures		
14	**A**	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
	A	61. Explain BFS and DFS search techniques in Graph
	B	62. Implement a Dictionary (key, value) pair using Hash-table.
Unit: V – Sorting and Searching		
15	**A**	63. Discuss a Hash table advantages and disadvantage.
	A	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	65. Implement a Hash table using Array
16	**A**	66. Implement a linear search algorithm
	A	67. Implement a binary search algorithm
17	**A**	68. Implement a given sorting technique for Array A= {10,20,5,8,9,11,30}; sort in ascending order • Bubble Sort. • Insertion Sort
	B	69. Implement a given sorting technique for Array A= {10,20,5,8,9,11,30}; sort in ascending order • Bubble Sort. • Insertion Sort

18	**A**	70. Implement a given sorting technique for Array A= {10,20,5,8,9,11,30}; sort in ascending order • Selection Sort • Bucket Sort
	B	71. Implement a given sorting technique for Array A= {10,20,5,8,9,11,30}; sort in ascending order • Selection Sort • Bucket Sort
19	**A**	72. Implement a given sorting technique for Array A= {10,20,5,8,9,11,30}; sort in ascending order • Redix Sort • Shell Sort
	B	73. Implement a given sorting technique for Array A= {10,20,5,8,9,11,30}; sort in ascending order • Redix Sort • Shell Sort
20	**A**	74. Implement a given sorting technique for Array A= {10,20,5,8,9,11,30}; sort in ascending order • Counting Sort • Tree Sort
	B	75. Implement a given sorting technique for Array A= {10,20,5,8,9,11,30}; sort in ascending order • Counting Sort • Tree Sort
21	**A**	76. Implement a given sorting technique for Array A= {10,20,5,8,9,11,30}; sort in ascending order • Merge Sort • Quick Sort • Heap Sort
	B	77. Implement a given sorting technique for Array A= {10,20,5,8,9,11,30}; sort in ascending order • Merge Sort • Quick Sort • Heap Sort