Data Structure Lab (KCS351)
List of Experiments

S.No.	Program	Domain
1	Program for Array Inserion, Deletion and traversal	Array
2	Program for Insertion in Sorted Array	Array
3	***Program to Find the number which is not repeated in Array of integers, others are present for two times***	Array
4	Program For Linear Search	Array
5	Program for Binary Search	Array
6	***Program for Index Sequential Search***	Array
7	Program for Bubble, Selection and Insertion Sort	Array
8	***Program for Implementation of Shell Sort***	Array
9	Program for Quick Sort	Array
	Program for Randomized Quick Sort	Array
	Program for Quick Sort using Median element as Pivot	Array
10	Program for Merge Sort	Array
11	Program for Merging of two Sorted Arrays	Array
12	***Program for Finding set elements of A that belongs to set B***	Array
13	***Program for Finding set elements of A that does not belongs to set B***	Array
14	***Program for Set Union***	Array
15	***Program for Set Intersection***	Array
16	***Program for Set Difference***	Array
17	***Program for Counting Sort***	Array
18	***Program for Radix Sort***	Array
19	Program for Matrix Addition	Array
20	Program for Matrix Multiplication	Array
21	Program for Matrix transposition	Array
22	Program for Matrix transposition without second matrix	Array
23	Program to Print a given matrix in spiral form	Array
24	Program for Sorting the given Complex Numbers	Array
25	Program for Creation of Max Heap and Min Heap	Heap
26	***Program for Insertion in Max Heap/Min Heap***	Heap
27	***Program for Deletion from Max Heap and Min Heap***	Heap
28	***Program for Realizing Heap as Ascending/Descending Priority Queue***	Heap
Project 1: Program for Heap Sort		
Project 2: Performance Comparision of Sorting Algorithms		
29	***Program for Hash Table Implementation for Basic Hash Function (Without collisions)***	Hashing
30	***Program for Hash Table Implementation for Collision Resoulution using Linear Probing***	Hashing
31	***Program for Hash Table Implementation for Collision Resoulution using Quadratic Probing***	Hashing
32	***Program for Hash Table Implementation for Collision Resoulution using Double Hashing/Re-Hashing***	Hashing
33	***Program for Hash Table Implementation for Collision Resoulution using Chaining***	Hashing
34	***Finding Anagrams: There are two strings. Find out which characters should be deleted such that both strings contain the same characters (May be in different Order)***	Hashing
35	***There are some numbers in which some are appearing twice but one is not repeated. Find out the number which appears once.***	Hashing

36	***There are two arrays containing some elements. Find out what are the elements which are there in both the arrays what are not.***	**Hashing**
37	***find out the values of a,b,c,d (a,b,c,d<=1000) for which a3+b3=c3+d3.***	**Hashing**
	Project 3: Identification of tokens and identifiers and storage in Hash Table	
38	Program for finding length of a string	**String**
39	Program for reversing the given string	**String**
40	Program for finding if the given string is a palindrome	**String**
41	Program for finding word count in the Paragraph	**String**
42	Program for converting all lower case letters to upper case and vice versa in the given sentence	**String**
43	***Program for finding if the given word is present in the sentence and at what location***	**String**
44	***Program for sorting the given names in the dictionary order***	**String**
45	***Program for reversing all words in a sentence***	**String**
	Project 4: Program for automatic word spelling correction using Minimum Edit Distance	
46	***Program for Decimal to Binary Conversion***	**Stack**
47	***Program for Decimal to Octal Conversion***	**Stack**
48	***Program for Decimal to Hexadecimal Conversion***	**Stack**
49	***Program for Decimal to Any Base Conversion***	**Stack**
50	Program for Stack Primitive Operations	**Stack**
51	Program for Postfix Evaluation	**Stack**
52	Program for Infix to Postfix Coversion	**Stack**
53	Program for Infix to Prefix Coversion	**Stack**
54	Program for Prefix Evaluation	**Stack**
55	***Program to check the validity of Parenthesized Arithmetic Expression using Stack***	**Stack**
56	***Program to check the validity of Bracketed Arithmetic Expression using Stack***	**Stack**
57	***Program ro check if the given number is a palindrome using stacks***	**Stack**
58	***Program to Reverse the given String using Stack***	**Stack**
	Project 5: Program for evaluation of given arithmetic expression. The Expression may have variables and constants	
59	Program for finding factorial of a given number using recursion	**Recursion**
60	Program for Towers of Hanoi for n disk (user defined)	**Recursion**
61	Program for Computing A raised to power n using Recursion	**Recursion**
62	Program for Computing A raised to power n using Divide and Conquer	**Recursion**
63	Program for finding nth Fibonacci number using Recursion and improving its run time to save stack operations	**Recursion**
64	Program for finding GCD of two numbers using Recursion	**Recursion**
65	Program to reverse the given number using Recursion	**Recursion**
66	Program of Array Implementaion of Linear Queue	**Queue**
67	Program of Array Implementaion of CircularQueue	**Queue**
68	Program for Array Implementation of Double Ended Queue	**Queue**
69	Program for Array Implementation of Priority Queue	**Queue**
70	***Program for 1-D array implementation of Upper Traingular Sparse Matrix***	**Sparse Matrix**
71	***Program for 1-D array implementation of Lower Traingular Sparse Matrix***	**Sparse Matrix**
72	***Program for 1-D array implementation of Tridiagonal Sparse Matrix***	**Sparse Matrix**
73	***Program for Vector Representation of General Sparse Matrix***	**Sparse Matrix**
74	***Program For Linked List Implementation of General Sparse Matrix***	**Sparse Matrix**
75	***Program for Addition of two sparse Matrices***	**Sparse Matrix**
76	Program for Linear Linked List Primitive operations	**Linked List**

77	Program for Pair wise swap of elements in linked list	**Linked List**
78	Program to print Linked List contents in reverse order	**Linked List**
79	Program for Reversing the Linear Linked List	**Linked List**
80	Program for concatenation of Linear Linked List	**Linked List**
81	Program for Creation of Ascending Order Linear Linked List	**Linked List**
82	***Program for Merging two sorted Linked List***	**Linked List**
83	***Program for Union of two sorted Linked List (consider lists as sets)***	**Linked List**
84	***Program for Intersection of two sorted Linked List (consider lists as sets)***	**Linked List**
85	***Program for finding difference of two linked list (consider lists as sets)***	**Linked List**
86	***Program for Sorting the Linear Linked List***	**Linked List**
87	***Program for Splitting a Linked List***	**Linked List**
88	***To Detect if there is any cycle in the linked list. (use two pointers, once moves at a speed of one node, other moves at a speed of two nodes. If they collide with each other it means there is a cycle.***	**Linked List**
89	Program for Polynomial Addition using Linked List	**Linked List**
90	Program for Circular Linked List Primitive Operations	**Linked List**
91	Program for concatenation of Circular Linked List	**Linked List**
92	Program for Doubly linked list Primitive operations	**Linked List**
93	Program for Circular Doubly Linked List Primitive Operations	**Linked List**
94	Program for Linked List Implementaion of Linear Queue	**Linked List**
95	Program for Linked List Implementaion of Circular Queue	**Linked List**
96	***Program for Linked List Implementaion of Priority Queue***	**Linked List**
97	Program for Linked List Implementation of Stacks	**Linked List**
	Project 6: Program for Addition of Two very long Numbers	
	Project 7: Implementatio of Josephus Problem	
98	Program for recursive cretaion of Binary Tree and Traversals	**Binary Tree**
99	Program for cretaion of Binary Tree and finding its height	**Binary Tree**
100	Program for cretaion of Binary Tree and finding count of nodes having 2 children	**Binary Tree**
101	Program for cretaion of Binary Tree and finding count of nodes having 1 child	**Binary Tree**
102	Program for cretaion of Binary Tree and finding count of nodes having 0 child	**Binary Tree**
103	Program for finding if the given binary tree is complete	**Binary Tree**
104	Program for Level Order Traversal	**Binary Tree**
105	Program for finding Balance factor of a given node	**Binary Tree**
	Project 8: Program for Huffman Coding	
	Project 9: Creation of Binary Tree from Pre-Order and Inorder Traversal	
	Project 10: Program to Create Expression Tree and its Traversal	
106	Program for BST Insertion, traversal, Minumum, maximum and Successor operations	**Binary Search Tree**
107	Program for BST Deletion	**Binary Search Tree**
108	Program to convert the given BST to Max Heap	**Binary Search Tree**
109	Program to check if a binary tree is BST or not	**Binary Search Tree**
	Project 11: Program for Binary Search Tree Deletions	
110	Program for AVL Tree Rotations, Insertion and Traversal opearations	**AVL Tree**
	Project 12: Program for Implementation of Interval Tree	
111	Program for BFS on a Graph	**Graph**

112	Program for DFS on a Graph	**Graph**
113	Program for Warshall's Algorithm for APSP	**Graph**
114	Program for Dijikstra's Algorithm for SSSP	**Graph**
115	Program for Warhall's Algorithm for Transitive Closure	**Graph**
116	Prorgram for Prim's Algorithm for Minimal Spanning Tree	**Graph**
117	Prorgram for Kruskal's Algorithm for Minimal Spanning Tree	**Graph**
118	Program for topological sorting of given graph	**Graph**
Project 13: Program for implemenation of Travelling Salesman Problem		