

| Lab | Type | Practical |
|----------------------------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| Basic programs of C | | |
| 1 | A | 1. WAP to calculate area of a Circle ($A = \pi r^2$). |
| | A | 2. WAP to find whether a number is odd or even. |
| | A | 3. WAP to determine whether the entered character is vowel or not. |
| | A | 4. WAP to find the largest among the given three numbers by user. |
| | A | 5. WAP to check for the leap year. |
| | B | 6. WAP to convert seconds into hours, minutes & seconds and print in HH:MM:SS [e.g. 10000 seconds mean 2:46:40 (2 Hours, 46 Minutes, 40 Seconds)]. |
| | C | 7. WAP to convert number of days into year, week & days [e.g. 375 days mean 1 year, 1 week and 3 days]. |
| 2 | A | 8. WAP to find factorial of a number. (Using loop & recursion) |
| | A | 9. WAP to find power of a number using loop. |
| | A | 10. WAP to reverse a number. |
| | A | 11. WAP to find factors of a given number. |
| | A | 12. WAP to check whether a number is prime or not. |
| | B | 13. WAP to print prime numbers between given interval. |
| | B | 14. WAP to find the sum of $1 + (1+2) + (1+2+3) + (1+2+3+4) + \dots + (1+2+3+4+\dots+n)$. |
| | C | 15. WAP to print Armstrong number from 1 to 1000. |
| Unit: I – Array | | |
| 3 | A | 16. Write a program to read and display n numbers using an array. |
| | A | 17. Write a program to calculate sum of numbers from m to n. |
| | A | 18. Write a program to calculate average of first n numbers. |
| | A | 19. Write a program to find position of the smallest number & the largest number from given n numbers. |
| | B | 20. Write a program to find whether the array contains a duplicate number or not. |

| Lab | Type | Practical |
|----------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | B | 21. Read n numbers in an array then read two different numbers, replace 1 st number with 2 nd number in an array and print its index and final array. |
| | C | 22. WAP to print Pascal triangle. |
| | C | 23. WAP to sort the N names in an alphabetical order. |
| 4 | A | 24. Write a program to insert a number at a given location in an array. |
| | A | 25. Write a program to delete a number from a given location in an array. |
| | A | 26. Write a program to insert a number in an array that is already sorted in an ascending order. |
| | B | 27. Write a program to delete a number from an array that is already sorted in an ascending order. |
| | B | 28. Write a program to delete duplicate numbers from an array. |
| | B | 29. Write a program to merge two unsorted arrays. |
| | C | 30. WAP to replace lowercase characters by uppercase & vice-versa in a user specified string. |
| 5 | A | 31. Read two 2x2 matrices and perform addition of matrices into third matrix and print it |
| | A | 32. Read two matrices, first 3x2 and second 2x3, perform multiplication operation and store result in third matrix and print it. |
| | B | 33. Design anagram game using array. <ul style="list-style-type: none"> • Allow a user to enter N words and store it in an array. • Generate a random number between 0 to N-1. • Based on the random number generated display the word stored at that index of an array and allow user to enter its anagram. • Check whether the word entered by the user is an anagram of displayed number or not and display an appropriate message. • Given a word A and word B. B is said to be an anagram of A if and only if the characters present in B is same as characters present in A, irrespective of their sequence. For ex: "LISTEN" == "SILENT" |

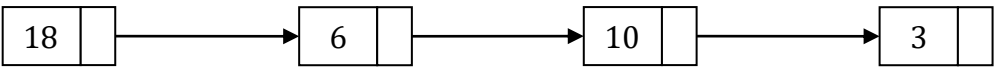
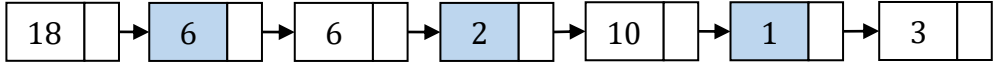
| Lab | Type | Practical |
|------------------------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Unit: I – Stack | | |
| 6 | A | 34. Write a menu driven program to implement following operations on the Stack using an Array <ul style="list-style-type: none"> PUSH, POP, DISPLAY PEEP, CHANGE |
| | A | 35. How stack can be used to recognize strings aca, bcb, abcba, abbcbbba? Write a program to solve the above problem. |
| | B | 36. 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 | 37. Valid Parenthesis Problem Chef has a string which contains only the characters '{', '}', '[', ']', '(' and ')'. Now Chef wants to know if the given string is balanced or not. If is balanced then print 1, otherwise print 0. A balanced parenthesis string is defined as follows: <ul style="list-style-type: none"> The empty string is balanced If P is balanced then (P), {P}, [P] is also balanced if P and Q are balanced PQ is also balanced "()", "{}", "[]" are balanced parenthesis strings "([{}])", "())" are not balanced. <p>Input Format: The first line of the input contains a single integer T denoting the number of test cases. The description of T test cases follows. The first and only line of each test case contains a single string</p> <p>Output Format: For each test case, print a single line containing the answer.</p> <p>Sample Example:</p> <div style="display: flex; justify-content: space-around; align-items: flex-start;"> <div style="border: 1px solid black; padding: 5px; width: 40%;"> <p><u>Input:</u></p> <p>4</p> <p>()</p> <p>(D]</p> <p>({}O){}</p> <p>[{}]</p> </div> <div style="border: 1px solid black; padding: 5px; width: 40%;"> <p><u>Output:</u></p> <p>1</p> <p>0</p> <p>1</p> <p>0</p> </div> </div> |

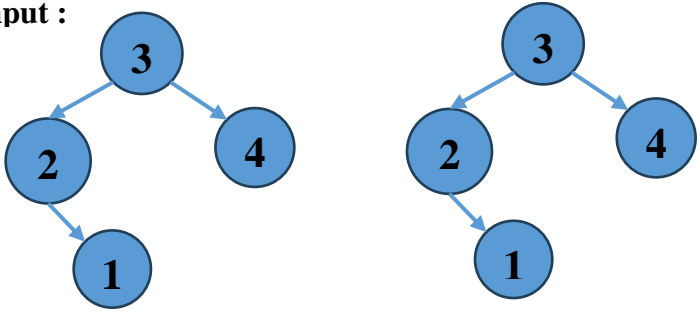
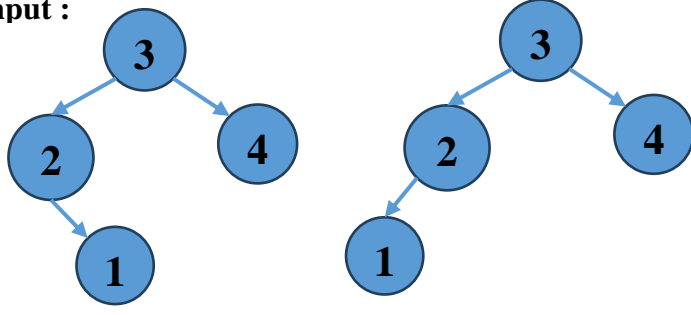
| Lab | Type | Practical |
|-----|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | C | <p>38. Removing stars from a string Problem</p> <p>You are given a string <i>s</i>, which contains stars *. In one operation, you can: Choose a star in <i>s</i>. Remove the closest non-star character to its left, as well as remove the star itself. Return the string after all stars have been removed.</p> <p>Note :</p> <ul style="list-style-type: none"> The input will be generated such that the operation is always possible. It can be shown that the resulting string will always be unique. <p>Sample Example-1: Input: <i>s</i> = "leet**cod*e". Output: "lecoe"</p> <p>Sample Example-2: Input: <i>s</i> = "erase*****" Output: ""</p> |
| | C | <p>39. Merge Intervals Problem</p> <p>Given a set of time intervals in any order, our task is to merge all overlapping intervals into one and output the result which should have only mutually exclusive intervals.</p> <p>Sample Example-1: Input: Intervals = {{1,3},{2,4},{6,8},{9,10}} Output: {{1, 4}, {6, 8}, {9, 10}} Explanation: Given intervals: [1,3],[2,4],[6,8],[9,10], we have only two overlapping intervals here,[1,3] and [2,4]. Therefore we will merge these two and return [1,4],[6,8],[9,10]</p> <p>Sample Example-2: Input: Intervals = {{6,8},{1,9},{2,4},{4,7}} Output: {{1, 9}}</p> |

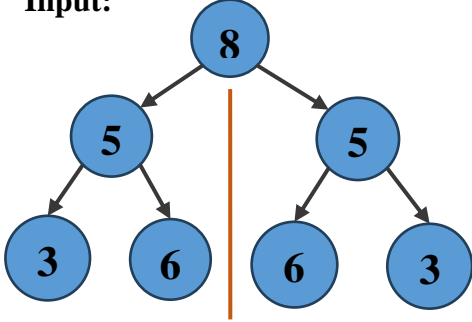
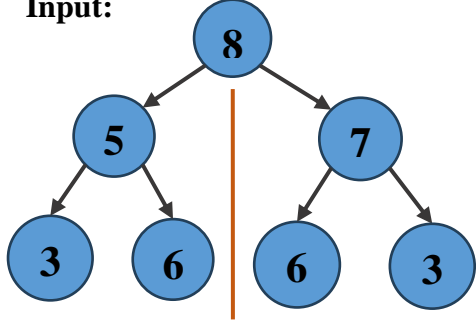
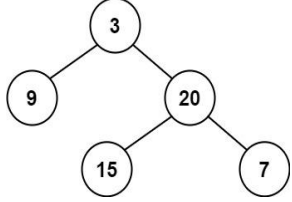
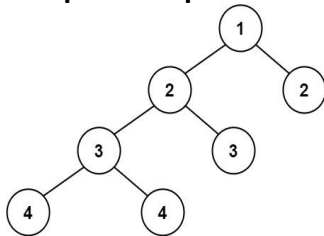
| Lab | Type | Practical |
|-----|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | C | <p>40. Vowel Anxiety Problem</p> <p>Utkarsh has recently started taking English-language classes to improve his reading and writing skills. However, he is still struggling to learn English. His teacher gave him the following problem to improve his vowel-identification skills:</p> <p>There is a string S of length N consisting of lowercase English letters only. Utkarsh has to start from the first letter of the string. Each time he encounters a vowel; he has to reverse the entire substring that came before the vowel. Utkarsh needs help verifying his answer. Can you print the final string after performing all the operations for him?</p> <p>Input Format:</p> <ul style="list-style-type: none"> • First line will contain T, number of test cases. Then T test cases follow. • The first line of each test case contains N, the length of the string. • The second line contains S, the string itself. <p>Output Format:</p> <p>For each test case, output in a single line the final string after traversing S from left to right and performing the necessary reversals.</p> <p>Sample Example:</p> <div style="display: flex; justify-content: space-around; align-items: flex-start;"> <div style="border: 1px solid black; padding: 5px; width: 30%;"> <p><u>Input:</u></p> <p>2</p> <p>10</p> <p>abcdefghij</p> <p>7</p> <p>bcadage</p> </div> <div style="border: 1px solid black; padding: 5px; width: 30%;"> <p><u>Output:</u></p> <p>hgfeabcdij</p> <p>gacbade</p> </div> </div> |
| 7 | A | 41. Write a program to convert infix notation to postfix notation using stack. |
| | B | 42. Write a program to convert infix notation to prefix notation using stack. |
| | A | 43. Write a program for evaluation of postfix Expression using Stack. |
| | B | 44. Write a program for evaluation of prefix Expression using Stack. |

| Lab | Type | Practical |
|-------------------------|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | C | <p>45. You have an array A of integers of size N, an array B (initially empty) and a stack S (initially empty). You are allowed to do the following operations:</p> <ul style="list-style-type: none"> • Take the first element of array A and push it into S and remove it from A. • Take the top element from stack S, append it to the end of array B and remove it from S. <p>You have to tell if it possible to move all the elements of array A to array B using the above operations such that finally the array B is sorted in ascending order.</p> <p>Input Format :</p> <ul style="list-style-type: none"> • First line will contain T, number of testcases. Then the testcases follow. • First line of each testcase contains a single integer N. • Second line of each testcase contains N distinct integers : A_1, A_2, \dots, A_N. <p>Output Format :</p> <p>For each testcase, if it possible to move all the elements of array A to array B using the above operations such that finally, the array B is sorted in ascending order, print "YES" (without quotes), else print "NO" (without quotes).</p> <p>Sample Example:</p> <div style="display: flex; justify-content: space-around; align-items: flex-start;"> <div style="border: 1px solid black; padding: 5px; width: 200px;"> <p><u>Input:</u></p> <p>2</p> <p>4</p> <p>1 2 4 3</p> <p>4</p> <p>1 3 4 2</p> </div> <div style="border: 1px solid black; padding: 5px; width: 100px;"> <p><u>Output:</u></p> <p>YES</p> <p>NO</p> </div> </div> |
| Unit: II – Queue | | |
| 8 | A | <p>46. Write a menu driven program to implement following operations on the Queue using an Array</p> <ul style="list-style-type: none"> • ENQUEUE • DEQUEUE • DISPLAY |
| | B | <p>47. Write a menu driven program to implement following operations on a circular queue using an Array</p> <ul style="list-style-type: none"> • Insert • Delete • Display all elements of the queue |

| Lab | Type | Practical |
|-------------------------------|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | C | 48. Write a menu driven program to implement following operations on the Doubled Ended Queue using an Array <ul style="list-style-type: none"> • Insert at front end, Insert at rear end • Delete from front end, Delete from rear end • Display all elements of the queue |
| Unit: II – Linked List | | |
| 9 | A A A A B B C | 49. WAP to allocate and de-allocate memory for int, char and float variable at runtime. 50. WAP to get and print the array elements using Pointer. 51. WAP to calculate the sum of n numbers using Pointer. 52. WAP to find the largest element in the array using Pointer. 53. WAP to sort the array elements using Pointer. 54. WAP to check whether the string is Palindrome or not using Pointer. 55. WAP to define a C structure named Student (roll_no, name, branch and batch_no) and also to access the structure members using Pointer. |
| 10 | A | 56. Write a program to implement a node structure for singly linked list. Read the data in a node, print the node. |
| | A | 57. Write a menu driven program to implement following operations on the singly linked list. <ul style="list-style-type: none"> • 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. |
| | B | 58. Write a program to count the number of nodes in a singly circular linked list. |
| | C | 59. WAP to check whether 2 singly linked lists are same or not. |
| 11 | A | 60. Write a program to implement stack using singly linked list. |
| | B | 61. Write a program to implement queue using singly linked list. |
| | C | 62. WAP to remove duplicate elements from a singly linked list. |
| 12 | A | 63. Write a program to copy a linked list. |
| | A | 64. Write a program to reverse a linked list. |

| Lab | Type | Practical |
|-----------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | B | 65. Write a program to sort elements of a linked list. |
| | C | 66. WAP to swap K^{th} node from beginning with K^{th} node from end in a singly linked list. |
| | C | <p>67. WAP to perform given operation in the linked list. There exist a Linked List. Add a node that contains the GCD of that two nodes between every pair adjacent node of Linked List.</p> <p>Input </p> <p>Output </p> |
| | C | <p>68. Write a program to swap two consecutive nodes in the linked list. Don't change the values of nodes, implement by changing the link of the nodes.</p> <ul style="list-style-type: none"> Input: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8$ Output: $2 \rightarrow 1 \rightarrow 4 \rightarrow 3 \rightarrow 6 \rightarrow 5 \rightarrow 8 \rightarrow 7$ |
| | C | <p>69. Write a program to remove the duplicates nodes from given sorted Linked List.</p> <ul style="list-style-type: none"> Input: $1 \rightarrow 1 \rightarrow 6 \rightarrow 13 \rightarrow 13 \rightarrow 13 \rightarrow 27 \rightarrow 27$ Output: $1 \rightarrow 6 \rightarrow 13 \rightarrow 27$ |
| 13 | A | <p>70. Write a menu driven program to implement following operations on the circular linked list.</p> <ul style="list-style-type: none"> 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. |
| | B | 71. WAP to split a circular linked list into two halves. |
| | C | 72. Write a program to perform addition of two polynomial equations using appropriate data structure. |
| 14 | A | <p>73. Write a menu driven program to implement following operations on the doubly linked list.</p> <ul style="list-style-type: none"> 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) |
| | B | 74. WAP to delete alternate nodes of a doubly linked list. |

| Lab | Type | Practical |
|--------------------------------------------------------------|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | C | 75. Write a program to simulate music player application using suitable data structure. There is no estimation about number of music files to be managed by the music player. Your program should support all the basic music player operations to play and manage the playlist. |
| Unit: III – Nonlinear Data Structures: Tree and Graph | | |
| 15 | A | 76. Write a menu driven program to implement Binary Search Tree (BST) & perform following operations: <ul style="list-style-type: none"> • Insert a node • Delete a node • Search a node |
| | B | 77. Write a menu driven program to implement Binary Search Tree (BST) & perform following operations: <ul style="list-style-type: none"> • Preorder Traversal • Postorder Traversal • Inorder Traversal |
| | C | 78. Write a program to check whether the given two trees are same or not. <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <p>Input :</p>  <p>Output: Given trees are same</p> </div> <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <p>Input :</p>  <p>Output: Given trees are not same</p> </div> |

| Lab | Type | Practical |
|-----|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | C | <p>79. Write a program to check whether the given tree is symmetric or not.</p> <div> <p>Input:</p>  <p>Output: Given tree is symmetric</p> </div> <div> <p>Input:</p>  <p>Output: Given tree is not symmetric</p> </div> |
| | C | <p>80. Given a binary tree, determine if it is height-balanced (A height-balanced binary tree is a binary tree in which the depth of the two subtrees of every node never differs by more than one).</p> <div> <p>Sample Example-1:</p>  <p>Input: root = [3,9,20,null,null,15,7] Output: TRUE</p> </div> <div> <p>Sample Example-2:</p>  <p>Input: root = [1,2,2,3,3,null,null,4,4] Output: FALSE</p> </div> |
| 16 | A | 81. Write a program to construct a binary tree from given Postorder and Preorder traversal sequence. |
| | B | 82. WAP to find the smallest and largest elements in the Binary Search Tree. |

| Lab | Type | Practical |
|-----|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | C | <p>83. Write a program to implement phone book dictionary using Binary Search Tree which provides following operations:</p> <ul style="list-style-type: none"> • Add new entry in phone book, • Remove entry from phone book, • Search phone number • List all entries in ascending order of name and • List all entries in descending order of name |
| 17 | A | <p>84. Write a program to create a graph & implement the adjacency list representation of the graph</p> <ul style="list-style-type: none"> • Apply DFS and BFS on the given graph. |
| | A | <p>85. Write a program to apply DFS & BFS for a graph.</p> |
| | C | <p>86. You are given an undirected graph with N nodes (numbered 1 through N). For each valid i, the i-th node has a weight W_i. Also, for each pair of nodes i and j, there is an edge connecting nodes if $j - i \neq W_j - W_i$.</p> <p>Find the number of connected components in this graph.</p> <p>Input Format:</p> <ul style="list-style-type: none"> • The first line of the input contains a single integer T denoting the number of test cases. The description of T test cases follows. • The first line of each test case contains a single integer N. • The line contains N space-separated integers W_1, W_2, \dots, W_N. <p>Output Format :</p> <p>For each test case, print a single line containing one integer --- the number of components in the graph.</p> <p>Sample Example:</p> <div style="display: flex; justify-content: space-around; align-items: flex-start;"> <div style="border: 1px solid black; padding: 5px; margin: 5px;"> <p><u>Input:</u></p> <p>2</p> <p>2</p> <p>1 2</p> <p>2</p> <p>2 1</p> </div> <div style="border: 1px solid black; padding: 5px; margin: 5px;"> <p><u>Output:</u></p> <p>2</p> <p>1</p> </div> </div> |

| Lab | Type | Practical |
|----------------------------------------|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Unit: IV – Hashing | | |
| 18 | A | 87. In an array of 20 elements, arrange 15 different values, which are generated randomly between 1,00,000 to 9,99,999. Use hash function to generate key and linear probing to avoid collision. $H(x) = (x \bmod 18) + 2$. Write a program to input and display the final values of array. |
| | B | 88. Implement a Dictionary (key, value) pair using Hash-table. |
| Unit: V – Sorting and Searching | | |
| 19 | A | 89. Write a program to implement a Linear Search using Array. |
| | A | 90. Write a program to implement a Binary Search using Array. (Iterative & recursive) |
| 20 | A | 91. Write a program to implement Bubble using Array. |
| | B | 92. Write a program to implement Selection Sort using Array. |
| 21 | A | 93. Write a program to implement Insertion Sort using Array. |
| | B | 94. Write a program to implement Shell Sort using Array. |
| 22 | A | 95. Write a program to implement Merge Sort using Array. |
| | B | 96. Write a program to implement Quick Sort using Array. |
| 23 | A | 97. Write a program to implement Counting Sort using Array. |
| | B | 98. Write a program to implement Radix Sort using Array. |
| 24 | A | 99. Write a program to implement Heap Sort using Array. |
| | B | <p>100. Given an array nums with n objects colored red, white, or blue, sort them in-place so that objects of the same color are adjacent, with the colors in the order red, white, and blue. We will use the integers 0, 1, and 2 to represent the color red, white, and blue, respectively.</p> <p>Sample Example-1: Input: nums = [2,0,2,1,1,0] Output: [0,0,1,1,2,2]</p> <p>Sample Example-2: Input: nums = [2,0,1] Output: [0,1,2]</p> |