MODULE-1: BASIC CONCEPTS

1. Write a C program to keep reading the integer elements through key board until negative number is entered, once the negative number is entered terminate the program and display the sum of positive elements without using array.
2. How does a structure differ from a union? Mention any 2 uses of structure. What is a bit field? Why are bit fields used with structures?
3. Write a C program using structure to find the average marks based on best of two test marks for 100 students.
4. Define structure? Initialization of structure and declaration of structure.
5. Write short notes on nested structure, pointer to pointer.
6. Design, develop, and execute a program in C to read a sparse matrix of integer values and to search the sparse matrix for an element specified by the user. Print the result of the search appropriately. Use the triple <row, column, value> to represent an element in the sparse matrix.
7. Write a C program to count the unique elements in an array.
8. Develop a structure to represent planet in the solar system. Each planet has field for the planets name, its distance from the sun in miles and the number of moon it has. Write a program to read the data for each planet and store. Also print the name of the planet that has less distance from the sun.
9. Write a C program using structure within a structure to read the employee database and display the same
10. List the ADT of polynomial?
11. List the ADT of Sparse matrix?
12. Illustrate Sparse transpose matrix with suitable Example.
13. Define polynomials? Explain the representation of polynomials?
14. Show how polynomial represents in data structure? Explain polynomial Addition
15. What is algorithm? Explain properties of algorithm
16. What are asymptotic notation explain wit example?
17. How performance measurements help in data structure? Explain the two methods of performance measurement?
18. Define the following

i) Space complexity

ii) Time complexity

16 Define self referential structure? Explain how it is help full for defining in linked list

MODULE-2: STACKS AND QUEUES

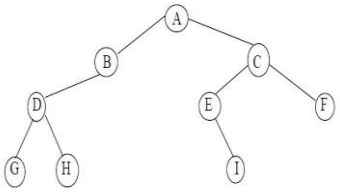
19. How do you define a data structure? How is stack a data structure? Give a C program to construct a stack of integers and perform all the necessary operations on it.
20. Write an algorithm to convert a valid infix expression to a postfix expression. Also
    i. Evaluate the following suffix expression for the values: A=1 B=2 C=3.
    ii. AB+C-BA+C$.
21. What is the advantage of circular queue over ordinary queue? Mention any 2 applications of queues. Write an algorithm CQINSERT for static implementation of circular queue.
22. Explain the working of a simple queue.
23. Define stack. Implement push and pop functions for stack using arrays.
    Implement addq and delete functions for the circular queue.
24. Write the postfix form of the following expressions
    i. (a+b)*d + e / (f + a * d) +c
    ii. ((a/b-c+d)) *(e-a)*c)
    iii. a/b-c+d*e-a*c
25. Write an algorithm to convert postfix expression to infix expression.
26. Write an algorithm to convert prefix to postfix expression.
27. Write an algorithm to evaluate prefix expression with an example.
28. Define queue? Types of queues
29. Explain the operation of ordinary queue
30. Write a C program to implement ordinary queue operation.
31. Differentiate ordinary queue and circular queue
32. Write a C program for circular queue and perform the following operation such as insertion, deletion and display.

MODULE-3: LINKED LISTS

33. List out any two applications of linked list and any two advantages of doubly linked list over singly linked list.
34. Write a C program to simulate an ordinary queue using a singly linked list.

35. Give an algorithm to insert a node at a specified position for a given singly linked list.
36. Write a function to insert a new node before the node value.
37. Write a C program to create a linked list and perform the following operations
    i. Insert the node at the beginning
    ii. Insert the node at the end
    iii. Display the list
38. Briefly explain the structures of different types of linked lists. Write a c function to count number of elements present in a singly linked list.
39. Write a c program to perform the following operations on doubly linked list :
    i. Insert a node at the beginning
    ii. Delete a node from the end.
    iii. Display the list.
40. Write short note on circular lists.
41. Write the different polynomial representation with example
42. Write the difference between singly linked list, double link list and circular doubly linked list.

MODULE-4: TREES

43. Define the following with an example
    i. Binary tree
    ii. Complete binary tree
    iii. Almost complete binary tree
    iv. Binary search tree
    v. Depth of a tree
44. Given the following graph, write the inorder, preorder and postorder traversals.
45. 

46. In brief describe any 4 applications of trees. (June / July08)
47. Construct a binary tree from the traversal order given below:

i. PREORDER = A B D E F C G H L J K

ii. INORDER = D B F E A G C L J H K

48. Construct a binary tree for: ((6 + (3 - 2) * 5) ^2 + 3).
49. Write c function for the following tree traversals:
    i. Inorder ii) Preorder iii) Postorder
50. What is a tree? Explain
    i. Root tree
    ii. Degree
    iii. Sibling
    iv. Depth of a tree and give example
51. What is a binary tree? State the properties? How it is represented using array and linked list give example.
52. Define max heap? Write a C function to insert an item into max heap.
53. For any non empty binary tree T if no is the number of leaf nodes and n2 the number of nodes of degree 2. Then prove that n0=n2+1.
54. Explain the different tree traversal methods.
55. What is binary tree? Explain the different operations performed on binary tree.
56. Explain the different schemes of representation of binary tree.
57. Define the following with a suitable example.
    i) Binary tree
    ii) Degree of a binary tree
    iii) Level of a binary tree
    iv) Sibling.
58. Explain the following with an example:
    i. i) Forest ii) graph iii) winner tree.
59. Describe the binary search tree with an example. Write a iterative function to search for a key value in a binary search tree.
60. Explain the following with an example
    i. Selection trees
    ii. Forest tree and its traversals
61. Describe the binary search tree with an example. Write a recursive function to search for a key value in a binary search tree.

62. Construct an inorder tree for the following elements. 66, 56, 12, 34, 78, 98,12,-56, 4, 18.
63. Explain selection trees, with suitable example
64. What is a forest? With a suitable example illustrate how you transform a forest into a binary tree.

MODULE-5: **EFFICIENT BINARY SEARCH TREES**

1. Describe the following with an Example
   i) Height balanced trees
   ii) ii) Optimal BST. (10 Marks)
2. Explain the Red-black tree. State its properties. (10 Marks)
3. What is an AVL tree? Write the algorithm to insert an item into AVL tree. (10 Marks)
4. Explain the red black tree with an example. State its properties. (10 Marks)
5. Write short notes on: Optimal binary search trees, AVL trees, Red – black trees and Splay trees.
   (10 Marks)
6. Write the difference between AVL tree and splay tree and compare both with binary tree.

   (10 Marks)
7. Give the node structure of an expression tree. Explain how the expression is evaluated. (10 Marks)
8. Write a function to :i) Find the maximum element in the binary search tree.
   ii) To search an element in the tree. (10 Marks)