1)	Beginning with an empty binary tree, construct binary tree by inserting the values in the order given. After constructing a binary tree perform following operations on it- • Perform preorder recursive traversal • **Postorder, Inorder non-recursive traversal** • Count number of leaves, number of internal nodes. **OR** Construct an expression tree from the given prefix expression (e.g. +--a*bc/def) and perform following operations on it 1) traverse it using non recursive post order traversal 2) Count and Display operands in given expression 3) Count and display operators in given expression
2)	Beginning with an empty binary tree, construct binary tree by inserting the values in the order given. After constructing a binary tree perform following operations on it- • Perform post order recursive traversal • **Inorder, Preorder non-recursive traversal** • Find the height of tree **OR** Construct an expression tree from the given prefix expression (e.g. +--a*bc/def) and perform following operations on it 1) traverse it using non recursive post order traversal 2) Count and Display operands in given expression 3) Count and display operators in given expression
3)	Beginning with an empty binary tree, construct binary tree by inserting the values in the order given. After constructing a binary tree perform following operations on it- • Perform inorder, recursive traversal • **Preorder, post-order non-recursive traversal** • Find mirror image of a tree **OR** Construct an expression tree from the given prefix expression (e.g. +--a*bc/def) and perform following operations on it 1) traverse it using non recursive post order traversal 2) Count and Display operands in given expression 3) Count and display operators in given expression

4)	Beginning with an empty binary tree, construct binary tree by inserting the values in the order given. After constructing a binary tree perform following operations on it- • Perform non-recursive inorder traversal, recursive postorder • **Level wise printing** • Display and count leaf nodes • Display and count internal nodes **OR** Construct an expression tree from the given prefix expression (e.g. +--a*bc/def) and perform following operations on it 4) traverse it using non recursive post order traversal 5) Count and Display operands in given expression 6) Count and display operators in given expression
5)	Construct binary search tree. Provide facility for **adding new entries**, **deleting any element**. Provide facility to display whole data sorted in **ascending order using non-recursive traversal**. **Search** an element present in a tree and **display number of comparisons** required to search.
6)	Construct binary search tree. Provide facility for adding **new entries**. Provide facility to display whole data sorted in **descending order using non-recursive traversal**. Find **Minimum and maximum** data value in tree. Also find how many **maximum comparisons** may require for finding any keyword.
7)	A dictionary stores keywords and its meanings. Provide facility for adding **new keywords**, **updating values** of any entry, **delete** any entry. Provide facility to display whole data sorted in **ascending/ Descending** order. Also find how many **maximum comparisons** may require for finding any keyword. Use Binary Search Tree for implementation. **OR** Beginning with an empty binary search tree. Construct the binary search tree by inserting the values in given order. After constructing binary search tree perform following operations 1) Find numbers of node in longest path 2) Minimum and maximum data value found in tree 3) Change a tree so that the roles of the left and right pointers are swapped at every node. 4) Display in level order.

8)	Construct In ordered Threaded binary Search Tree. Perform its **inorder** traversal. Display **Smallest and largest** value in it. Display lbit and rbit of each node.
9)	A dictionary stores keywords and its meanings. Provide facility for **adding new keywords, deleting keywords**. Provide facility to display whole data sorted in **Descending order**. Also find **how many maximum comparisons** may require for finding any keyword. Use Binary Search Tree for implementation. **OR** Beginning with an empty binary search tree. Construct the binary search tree by inserting the values in given order. After constructing binary search tree perform following operations 1) Find numbers of node in longest path 2) Minimum and maximum data value found in tree 3) Change a tree so that the roles of the left and right pointers are swapped at every node 4) Display in level order.
10)	Create an inordered threaded binary search tree for integer. Perform **inorder traversal**, and **deletion of a node**. Note: Display lbit, rbit for every node
11)	Create an inordered threaded binary search tree for integer value. Perform **preorder** traversals and **deletion of a node**. Note: Display lbit, rbit for every node
12)	Create an inordered threaded binary search tree for integer value. Perform **inorder** and **preorder** traversals. Note: Display lbit, rbit for every node
13)	Implement all the functions of a dictionary (ADT) using open hashing technique: separate chaining using linked list Data: Set of (key, value) pairs, Keys are mapped to values, Keys must be comparable, and Keys must be unique. Standard Operations: Insert (key, value), Find(key), Delete(key). Use hash function as H(x) = (3x+5) %10

14)	Consider telephone book database of N clients. Make use of a hash table implementation to quickly look up client‘s telephone number. Make use of two collision handling techniques and compare them using number of comparisons required to find a set of telephone numbers (Note: Use l**inear probing with replacement and without replacement**). Perform following operations 1) Insert 2) Display 3) Search (number of comparisons required to search) Use Hash function as H(x) = (3*x + 5)%10
15)	Implement all the functions of a dictionary (ADT) using **open hashing technique**: **separate chaining using linked list Data**: Set of (key, value) pairs, Keys are mapped to values, Keys must be comparable, and Keys must be unique. Standard Operations: Insert (key, value), Find(key), Delete(key), Display
16)	Consider telephone book database of N clients. Make use of a hash table implementation to quickly look up client‘s telephone number. Make use of two collision handling techniques and compare them using number of comparisons required to find a set of telephone numbers (Note: Use linear probing with replacement and without replacement). Perform following operations 1. Insert 2. Delete 3. Search 4. Display
17)	Write menu driven C++ program to 1) Represent a given graph using adjacency list/Adjacency Matrix. Use the map of the area around the college as the graph 2) Display Graph 3) Perform DFS traversal using recursive/non-recursive algorithm. 4) Perform BFS traversal. **OR** There are fight paths between the cities. If there is a fight between the city A and B then there is an edge between the city A to B. The cost of an edge represents the time required or fuel required to travel from city A to B. Represent this as a graph using adjacency list where every node of graph represented by city name. Perform following operations 1) calculate in degree, out degree of vertices 2) check whether graph is connected or not

18)	Write menu driven C++ program to 1. Represent a given graph using adjacency list/Adjacency Matrix. Use the map of the area around the college as the graph 2. Display Graph 3. Perform BFS traversal/DFS 4. Calculate degree of each node. Note: Write your own Queue class **OR** There are fight paths between the cities. If there is a fight between the city A and city B then there is an edge between the city A to B. The cost of an edge represents the time required or fuel required to travel from city A to B. Represent this as a graph using adjacency list where every node of graph represented by city name. Perform following operations 1) calculate in degree, out degree of vertices 2) check whether graph is connected or not
19)	Create Graph using Adjacency Matrix /Adjacency List. Perform DFS & BFS traversal from given vertex.
20)	Department maintains a student information. The file contains roll number, name, division and address. Allow user to add, delete information of student. Display information of particular employee. If record of student does not exist an appropriate message is displayed. If it is, then the system displays the student details. Use **sequential file** to maintain the data.
21)	Implement the Heap sort algorithm for demonstrating heap data structure with modularity of programming language (consider integer data)
22)	Read the marks obtained by students of second year in an online examination of particular subject. Find out minimum marks obtained in that subject. Use heap data structure. Analyze the algorithm.

23)	A Dictionary stores keywords and its meanings. Provide facility for adding new keywords. Provide facility to display whole data sorted in ascending/ Descending order. Also find how many maximum comparisons may require for finding any keyword. Use Height balanced (AVL) tree.
24)	Given sequence k = k1 <k2 < … <kn of n sorted keys, with a successful and unsuccessful search probability pi and qi for each key ki. Build the Binary search tree that has the least search cost given the access probability for each key.
25)	You have a business with several offices; you want to lease phone lines to connect them up with each other; and the phone company charges different amounts of money to connect different pairs of cities. You want a set of lines that connects all your offices with a minimum total cost. Solve the problem by using weighted graph and applying minimum spanning tree algorithm. **OR** Tour operator organizes guided bus trips across the Maharashtra. Tourists may have different preferences. Tour operator offers a choice from many different routes. Every day the bus moves from starting city S to another city F as chosen by client. On this way, the tourists can see the sights alongside the route travelled from S to F. Client may have preference to choose route. There is a restriction on the routes that the tourists may choose from, the bus has to take a short route from S to F or a route having one distance unit longer than the minimal distance. Two routes from S to F are considered different if there is at least one road from a city A to a city B which is part of one route, but not of the other route
26)	Implementation of a direct access file -Insertion and deletion of a record from a direct access file.