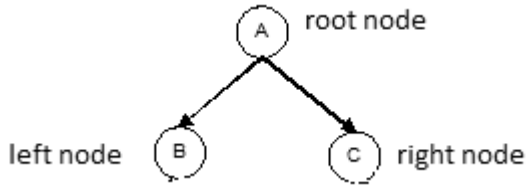
Exercise 5 Binary Tree Traversals

5. Write a C++ Program for implementation of Binary tree using following Traversals on it
a) Inorder b) preorder c) postorder

Objective: The objective of this exercise is to enable you to program traversing a binary tree using Tree traversal.

Procedure and description:
There are many operations that we often want to perform on trees. One notion that arises frequently is the idea of traversing a tree or visiting each node in the tree exactly once. A full traversal produces a linear order for the information in a tree. When traversing a binary tree we want to treat each node and its sub trees in the same fashion. If we let L, D, R stand for moving left, printing the data, and moving right at a node then there are six possible combinations of traversal: LDR, LRD, DLR, DRL, RDL, and RLD If we adopt the convention that we traverse left before right then only three traversals remain: LDR, LRD and DLR . To these we assign the names inorder, postorder and preorder.

Representation of binary tree:



Declaration of binary tree:

Class Tree_Node
Start
Int Left node;
Int rootnode;
Int right node;
End

Algorithm: The steps for this exercise are given below

Inorder traversal

Inorder traversal helps to print the binary search tree in sorted order. Here the traversal starts with left most sub trees then prints the parent node and then proceeds with the right subtree.

Following algorithm explain each step of Inorder traversal in detail

Step 1: Start from the root node.

Step 2: Now go to the leftmost node from the root node in the left subtree.

Step 3: When the leftmost node is reached. Print it down.

Step 4: Now visit/ go the leftmost node's parent node. Print it down.

Step 5: Now visit/ go to the first node of the right subtree.

Step 6: Proceed in the same way as in Step 2 - 5 till 'Printing' of all the nodes is finished

Post order traversal

In post order traversal, traversal starts with the left most sub tree then proceeds with right sub tree and print the parent of those nodes.

Following algorithm explain each step of postorder traversal in detail

Step 1: Start from the root node.

Step 2: Now go to the leftmost node from the root node in the left subtree.

Step 3: When the leftmost node is reached. Print it down.

Step 4: Now visit/ go the leftmost node's parent node.

Step 5: Now visit/ go to the first node of the right subtree.

Step 6: Proceed in the same way as in Step 2 - 5 till 'printing' of all the nodes is finished.

Preorder traversal

Here in preorder traversal, we start the traversal technique with the root node then proceeds towards the end of the left sub tree and then towards the right sub tree.

Following algorithm explain each step of preorder traversal in detail.

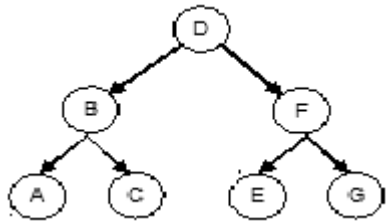
Step 1: Start from the root node. Print it down.

Step 2: Now go to the leftmost node from the root node in the left sub tree. On its way to the left most node, print all the nodes as it comes.

Step 3: When the leftmost node is reached. Print it down.
Step 4: Now visit the right sub tree of the leftmost node's parent node.
Step 5: Write down the first node of the right subtree.
Step 6: Now proceed in the same way as in Step 2 - 5 till 'printing' of all the nodes is finished.

Expected Output:

After execution of the program. Enter input binary tree nodes. For better understanding see pictorial representation of the input Binary tree and output.



Inorder: A B C D E F G
Postorder: A C B E G F D.
Preorder: D B A C F E G.